UNISYS

A Series GETSTATUS/SETSTATUS

Programming Reference Manual

Release 3.9.0

Priced Item

September 1991

U S America
8600 0346–000

UNISYS

A Series
GETSTATUS/SETSTATUS
Programming Reference Manual

Release 3.9.0

September 1991

U S America
8600 0346-000

Priced Item

Printed on recycled paper

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 THE DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or 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, indirect, 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.

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, Product Information, 25725 Jeronimo Road, Mission Viejo, CA 92691.

Page Status

Page	Issue
iii through ix	–000
x	Blank
xi through xix	–000
xx	Blank
xxi	–000
xxii	Blank
1–1 through 1–3	–000
1–4	Blank
2–1 through 2–23	–000
2–24	Blank
3–1 through 3–46	–000
4–1 through 4–48	–000
5–1 through 5–3	–000
5–4	Blank
6–1 through 6–18	–000
7–1 through 7–3	–000
7–4	Blank
8–1 through 8–36	–000
9–1 through 9–117	–000
9–118	Blank
10–1 through 10–4	–000
11–1 through 11–6	–000
12–1 through 12–55	–000
12–56	Blank
A–1 through A–2	–000
B–1 through B–9	–000
B–10	Blank
C–1 through C–2	–000
D–1 through D–5	–000
D–6	Blank

continued

continued

Page	Issue
Glossary–1 through 23	–000
Glossary–24	Blank
Bibliography–1 through 2	–000
Index–1 through 13	–000
Index–14	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

This manual provides information on how to perform GETSTATUS and SETSTATUS calls. These calls are a subset of Data Communications ALGOL (DCALGOL). GETSTATUS allows you to retrieve specific and general information about jobs, tasks, the status of peripheral and disk units, the status of the operating system and mainframe configuration, and the files in the disk directory. SETSTATUS provides you with interface form control of mix, unit, and operational functions from an object program.

Scope

This manual applies to all Unisys A Series computer systems.

Audience

This manual is written for system programmers.

Prerequisites

This manual assumes that you are proficient at using ALGOL, especially POINTER and hexadecimal constructs. You should also be familiar with the system commands that are available to computer operators. The manual further assumes that you are familiar with the data communication subsystem and logical I/O operations, which include message and file manipulation.

How to Use This Manual

This manual is intended to be used a reference source, not as a task-based set of instructions. It is to be accessed randomly.

Information returned by the GETSTATUS intrinsic is sometimes taken directly from Master Control Program (MCP) tables. The format and meaning of these tables are subject to changes between MCP releases.

The standard arithmetic operators are used in this manual as follows:

Operator	Meaning
+	Addition
–	Subtraction
*	Multiplication
/	Division

Organization

This manual is divided into two parts, one for GETSTATUS calls and the other for SETSTATUS calls. In turn, the GETSTATUS and SETSTATUS calls are organized by functional type such as mix entries, miscellaneous requests, unit requests, and so on. A glossary, a bibliography, and an index appear at the end of this manual.

Part 1. GETSTATUS Calls

This part of the manual describes GETSTATUS calls.

Section 1. General Information on GETSTATUS

This section provides general information about GETSTATUS calls.

Section 2. GETSTATUS Request Type 0 (Mix Entries)

This section describes the various GETSTATUS calls for mix entries.

Section 3. GETSTATUS Request Type 2 (Miscellaneous Requests)

This section describes the various GETSTATUS calls for miscellaneous requests. Not all miscellaneous requests are documented.

Section 4. GETSTATUS Request Type 3 (Directory Calls)

This section describes the various directory calls that you can make with GETSTATUS.

Section 5. GETSTATUS Request Type 4 (Disk and Tape Volumes)

This section describes the various GETSTATUS calls for disk and tape volumes.

Section 6. GETSTATUS Request Type 5 (Unit Requests)

This section describes the various GETSTATUS calls for unit types.

Part 2. SETSTATUS Calls

This part of the manual describes SETSTATUS calls.

Section 7. General Information on SETSTATUS

This section provides general information about SETSTATUS calls.

Section 8. SETSTATUS Request Type 0 (Mix Entries)

This section describes the various SETSTATUS calls for mix entries.

Section 9. SETSTATUS Request Type 2 (Miscellaneous Requests)

This section describes the various SETSTATUS calls for miscellaneous requests. Not all miscellaneous requests are documented.

Section 10. SETSTATUS Request Type 3 (Directory Requests)

This section describes the various SETSTATUS calls for archive directory requests.

Section 11. SETSTATUS Request Type 4 (Disk and Tape Volumes)

This section describes the various SETSTATUS calls for disk and tape volumes.

Section 12. SETSTATUS Request Type 5 (Unit Requests)

This section describes the various SETSTATUS calls for unit types.

Appendix A. Hard Errors

This appendix contains a list of hard error numbers and their meanings.

Appendix B. Soft Errors

This appendix contains a list of soft error numbers and their meanings.

Appendix C. Hardware Resource Codes and Unit Type Codes

This appendix contains a list of hardware resource codes and unit type codes that are used by GETSTATUS and SETSTATUS calls.

Appendix D. Example of a GETSTATUS Directory

This appendix contains an example of a GETSTATUS directory.

Related Product Information

***A Series ALGOL Programming Reference Manual, Volume 1: Basic Implementation* (form 8600 0098)**

This manual describes the basic features of the Extended ALGOL programming language. This manual is written for programmers who are familiar with programming concepts.

***A Series DCALGOL Programming Reference Manual* (form 8600 0841)**

This manual describes the Data Communications ALGOL (DCALGOL) language. This language is designed to support the implementation of message control systems (MCSs) and other resource monitoring and controlling programs that require access to special operating system interfaces. This manual is written for systems programmers.

A Series File Attributes Programming Reference Manual* (form 8600 0064). Formerly the *A Series I/O Subsystem Programming Reference Manual

This manual contains information about each file attribute and each direct I/O buffer attribute. The manual is written for programmers and operations personnel who need to understand the functionality of a given attribute. The *A Series I/O Subsystem Programming Guide* is a companion manual.

A Series I/O Subsystem Programming Guide* (form 8600 0056). Formerly the *A Series I/O Subsystem Programming Reference Manual

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 Security Administration Guide* (form 8600 0973)**

This guide describes systems-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* (form 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 SYSTEMSTATUS Programming Reference Manual* (form 8600 0452)**

This manual documents the SYSTEMSTATUS intrinsic of the Master Control Program (MCP). The SYSTEMSTATUS intrinsic provides information that can be used to efficiently monitor the performance of a running system. This manual is written for systems programmers.

A Series Task Attributes Programming Reference Manual* (form 8600 0502). Formerly the *A Series Work Flow Administration and Programming Guide

This manual describes all the task attributes available on A Series systems. It also gives examples of statements for reading and assigning task attributes in various programming languages. The *A Series Task Management Programming Guide* is a companion manual.

***A Series Work Flow Language (WFL) Programming Reference Manual* (form 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

Section 3. GETSTATUS Request Type 2 (Miscellaneous Requests)

Section 4. GETSTATUS Request Type 3 (Directory Calls)

Section 9. SETSTATUS Request Type 2 (Miscellaneous Requests)

Section 11. SETSTATUS Request Type 4 (Disk and Tape Volumes)

Section 12. SETSTATUS Request Type 5 (Unit Requests)

Appendix A. Hard Errors

Appendix B. Soft Errors

Appendix C. Hardware Resource Codes and Unit Type Codes

Figures

Tables

Part 1
GETSTATUS Calls

Section 1
General Information on GETSTATUS

The GETSTATUS intrinsic allows the user to retrieve specific and general information about jobs, tasks, the status of peripheral and disk units, the status of the operating system and mainframe configuration, and the files in the disk directory. In general, the GETSTATUS intrinsic can be employed only by a privileged process. However, some directory information (Request Type 3), volume library information (Request Type 4), and selected miscellaneous information (Request Type 2) can be requested by nonprivileged processes.

The GETSTATUS intrinsic is referenced as a type Boolean procedure with four parameters as follows:

```
BOOL := GETSTATUS (TYPE, SUBCLASS, MASK, ARY);
```

The TYPE, SUBCLASS, and MASK arguments are single-precision, REAL variables. ARY is a single-precision, REAL array row.

In the following descriptions of various GETSTATUS calls, two fields belonging to the TYPE parameter are referred to by the following names:

- Request Type refers to the field [7:8] of the TYPE parameter.
- SUBTYPE refers to the field [15:8] of the TYPE parameter.

GETSTATUS Request Types

GETSTATUS functions are grouped into major categories by Request Type. The Request Type categories currently implemented are as follows:

Request Type	Description
0	Mix requests. These requests retrieve information about processes in the system.
1	Reserved.
2	Miscellaneous requests.
3	Directory requests. These requests retrieve information about disk files, disk directories, and cataloged disk or tape information.
4	Volume library requests. These requests retrieve information about disk and tape volumes from the volume library and the volume directory.
5	Unit requests. These requests retrieve information about peripheral I/O units.

Each Request Type category is further broken down into specific SUBTYPEs for obtaining specific information. These SUBTYPEs are discussed in detail as part of the explanation for each Request Type.

Unless otherwise indicated, the process that is called must have been initiated from an ODT or must be privileged to use Request Types 0, 2, or 5. Unless otherwise indicated, a nonprivileged process can use Request Types 3 and 4.

Array Limits

The array parameter ARY can be a segmented or a long array. For all GETSTATUS calls, the value ARY[0].[19:20] must be greater than 1. If ARY[0].[19:20] is less than 2, GETSTATUS returns hard error 35: Boolean (1 & 35 [11:08]).

For all GETSTATUS calls, the value ARY[0].[19:20] must be less than SIZE(ARY) in words. If ARY[0].[19:20] is greater than or equal to SIZE(ARY), GETSTATUS returns hard error 36: Boolean (1 & 36 [11:08]). If bit 47 of the TYPE parameter is ON, ARY[0].[19:20] must be less than SIZE(ARY)– 3. Otherwise, hard error 36 is returned.

For all GETSTATUS calls, ARY[0].[39:20] must be less than ARY[0].[19:20]. Otherwise, hard error 42: Boolean (1 & 42 [11:08]) is returned.

Additional Words (ADDLWORDs)

Some requests for the GETSTATUS intrinsic require or might optionally use more than one word per logical entry to describe the request. The caller indicates the presence of additional words by storing a count of them in the field VALUEF ([38:06]) of certain words in the array ARY. In this subsection, the additional words are called ADDLWORD(1), ADDLWORD(2), and so forth.

Additional information cannot be supplied for requests that require GETSTATUS to generate the entire response after starting at some given point. For example, on directory requests (TYPE = 3), ADDLWORDs can be supplied only for SUBTYPE = 0.

Substandard Form Names

The parameters and results for some GETSTATUS calls use substandard form names. A substandard form name is a one-byte binary character count followed by the EBCDIC characters of the name. The parameters and results for other GETSTATUS calls use standard form names. A standard form name is coded in binary with lengths and codes. For additional information about standard forms, refer to the DISPLAYTOSTANDARD function in the *A Series DCALGOL Programming Reference Manual.*

Optional Dynamic MASK and SUBCLASS Values

MASK and SUBCLASS values other than the ones entered as the MASK and SUBCLASS parameters can be designated by placing one or more substitute pairs of values in the array ARY. The substitutes are placed in the array ARY starting at word ARY[1]. The number of words used for substitute MASK and SUBCLASS values should be indicated by the value of ARY[0].[39:20]. This field is normally 0; GETSTATUS assumes that the first information can be found (or is to be returned) relative to word 1.

However, if ARY[0].[39:20] is not 0, the assumed first word location is shifted to the value specified in ARY[0].[39:20]. When you use this technique, you should increase ARY[0].[19:20] by a corresponding amount.

The first word of each pair becomes the dynamic MASK, and the second word becomes the SUBCLASS value. Each parameter entry above ARY[0].[39:20] selects the particular MASK and SUBCLASS to be used with an index value in the ADDLINFOF ([46:08]) field. When the ADDLINFOF field of a parameter word in the array ARY is not equal to 0, the MASK and SUBTYPE parameters are changed. For example, suppose that ADDLINX is the value of the ADDLINFOF field in a parameter word. Then MASK is replaced by ARY [ADDLINX], and SUBCLASS is replaced by ARY [ADDLINX + 1]. The ADDLINFOF fields in the parameter words are not preserved in the returned results.

Error Detection

Bit 0 of the Boolean value returned by GETSTATUS indicates whether or not an error occurred while GETSTATUS was processing a request. When bit 0 is turned on, GETSTATUS detected either a soft error or a hard error. If the field [11:08] of the Boolean result equals 0, then a soft error occurred. When a soft error occurs, GETSTATUS stores one or more soft error words in the array ARY. Depending on the specific Request Type and SUBTYPE of the call, GETSTATUS stores coded result words in the array. These result words are described for each individual Request Type later in this manual. Instead of a coded result word, GETSTATUS might store a soft error word when necessary. A soft error word can be recognized because the ERRORF bit (bit 47) is turned on. Further examination of the ERRORVALUEF field [46:08], which is within the soft error word, indicates the type of soft error. Appendix B contains a list of all the possible soft errors that can be returned.

If the field [11:08] of the Boolean result is not 0, GETSTATUS encountered a hard error. GETSTATUS was unable to complete the original request, and you should not rely on any information returned by GETSTATUS in the array ARY. Appendix A contains a list of all the possible hard errors that can be returned.

Section 2
GETSTATUS Request Type 0 (Mix Entries)

GETSTATUS mix requests are grouped under Request Type 0. These GETSTATUS calls retrieve information about the status of processes (jobs, tasks, libraries, and so on) in the system. For example, the controller uses these calls to generate displays for the system commands MX (Mix Entries), W (Waiting Mix Entries), and Y (Status Interrogate). For information on these and any other system commands, refer to the *A Series System Commands Operations Reference Manual*.

To completely understand this discussion of GETSTATUS mix type calls, you should be familiar with terms such as *stack number* and *independent runner* and with task attributes. These concepts are explained in the *A Series Task Attributes Programming Reference Manual*.

To use mix calls, the calling process must have privileged status or must have been started from an ODT with a WFL *JOB* or *RUN* statement, but not with a primitive *??RUN* system command. Otherwise, a security error occurs and GETSTATUS returns hard error 43 (refer to Appendix A).

The general form of the GETSTATUS call is as follows:

```
B := GETSTATUS (TYPE, SUBCLASS, MASK, ARY);
```

Parameters

The following parameters are needed to make GETSTATUS calls about mix entries.

TYPE

This parameter generally selects the specific case and subcase within the GETSTATUS intrinsic. For mix requests, the fields within the word are as follows:

Word	Field	Value	Description
TYPE.TYPEF	[7:08]	0	Selects Request Type 0.
TYPE.SUBTYPEF	[15:08]	0	Retrieves information for a designated list of mix numbers.
		1	Retrieves information corresponding to a sequence of stack numbers starting at a designated stack number.

continued

continued

Word	Field	Value	Description
		2	Retrieves information about a process and the siblings and descendents of that process.
		3	Returns a structured summary list of processes scheduled for execution, processes waiting for operator intervention, processes that are active, processes that are libraries, and database stacks.
		4	Retrieves information for a designated list of stack numbers.
		5	Returns a structured summary list of processes scheduled for execution, processes waiting for operator intervention, processes that are active, processes that are libraries, and database stacks.
		6	Reports the kind of job for a list of mix numbers.
		7	Reserved.
		8	Retrieves information about a process and the descendents of that process.
		9	Retrieves the USERCODE and processor time for a list of stack numbers.
TYPE.GSORGUNITF	[31:16]	Varies	For SUBTYPEs 2, 3, 5, and 8, GETSTATUS checks the ORGUNIT attribute of processes before it reports the processes. If the caller process is privileged, the GSORGUNITF field can be used by the caller to limit the report. If the GSORGUNITF field is not 0, GETSTATUS reports only on processes with an ORGUNIT value that matches the values in TYPE.GSORGUNITF. If the caller is privileged and the GSORGUNITF field is 0, no limits are applied. If the caller is not privileged (and the process was started from an ODT), the report is limited to processes with the same ORGUNIT as the caller.

continued

continued

Word	Field	Value	Description
TYPE.RETURNRAWF	[36:01]	0 or 1	If you turn this bit on, GETSTATUS returns the RSVP and DISPLAY messages of a program in an untranslated format. These messages are those returned if you turn on bit 12 (for RSVP) or 13 (for DISPLAY) in the MASK parameter for SUBCLASS 1 or 2.
TYPE.GSMLSLANGF	[37:01]	0 or 1	If you turn this bit on, you must place in the array ARY the name of the native language into which the multilingual system (MLS) display and RSVP messages are to be translated. The language should appear in the array starting at word SIZE (ARY)–6 of the array. The language should consist of a 1-byte binary length indicator followed by 1 to 17 EBCDIC characters. If you turn this bit off, if the length indicator is 0, or if the message is not available in the requested language, the translation is to be done with the LANGUAGE parameter of the calling process, the SYSTEMLANGUAGE system command, or the language in which the message is written.

continued

continued

Word	Field	Value	Description
TYPE.GSORGUSERF	[47:01]	1	The report is limited to processes running under a designated usercode. The usercode limitation does not apply to SUBTYPEs 6 or 9. If a usercode is designated, GETSTATUS does not report message control system (MCS) processes. The caller should place the usercode in the last three words of the array ARY starting at word SIZE (ARY)–3. The usercode should appear in the following form: *<character count>* *<usercode-ID>*. The variable <character count> is a 1-byte binary number that gives the number of characters in the usercode ID. The variable <usercode-ID> is the usercode in EBCDIC characters.
TYPE.GSORGUSERF	[47:01]	0	If GSORGUSERF is 0 or if <character count> is 0, the mix report is not limited by the usercode.

SUBCLASS

For SUBTYPEs 0, 1, 2, 4, and 8, the caller selects one of the two MASK tables to be used with SUBCLASS. For SUBTYPEs 0, 1, 2, 3, 4, 5, and 8, the sign bit (bit [46:01]) of SUBCLASS determines if GETSTATUS is to report invisible independent runners. If the sign bit is turned on, SUBCLASS is negative, and GETSTATUS can report the invisible independent runners; if the sign bit is turned off, GETSTATUS does not report them.

MASK

For SUBTYPEs 0, 1, 2, 4, and 8, the MASK parameter is used to select the process attributes and status information to be retrieved by GETSTATUS. Each bit that is turned on in the MASK parameter instructs GETSTATUS to report the corresponding task attribute (refer to Table 2–1 and Table 2–2 for a listing of valid MASK bits and their uses). For SUBTYPE 5, bits in the MASK parameter are used to select the sets of processes that are to be reported (refer to Table 2–3).

Array ARY Parameter

The caller should place information such as mix numbers or stack numbers, usercode, and so forth, in the array ARY to indicate to GETSTATUS the tasks that are to be reported. In turn, GETSTATUS stores the retrieved status and attribute information into the array ARY. The exact format of the array parameter and result information is described later in this section under the description of each particular SUBTYPE.

Request Type 0 Calls by SUBTYPE

The following Request Type 0 (mix entry) calls are grouped by SUBTYPE because they are functionally related and share common characteristics.

SUBTYPE 0, 1, 2, 4, and 8 Calls (Designated Processes)

SUBTYPEs 0, 1, 2, 4, and 8 return information about one or more designated processes. The information returned depends on the SUBCLASS in the MASK parameter, GETSTATUS returns available information that corresponds to the bit for each task reported. The SUBCLASS parameter selects the requested task attribute from the two available choices. If SUBCLASS equals plus or minus one, the information found in Table 2–1 applies. If SUBCLASS equals plus or minus two, the information found in Table 2–2 applies.

General Input Information

Some of the values reported in Tables 2–1 and 2–2 correspond exactly to task attributes. Other values are related to task attributes or status. These latter values are mainly for internal use by the Master Control Program (MCP) and might change in future releases.

In the two status information tables that follow, the link format for mix requests is as follows:

Word	Field	Description
GSINFOF	[15:16]	Length in bytes of the item linked.
XSLINKF	[32:17]	Word offset to the beginning information in the array ARY. When this offset is added to the *base*, it gives the index of the information. Refer to Figure 2–1.

Many of these links point to a name that is in *standard form*. For a description of standard form, refer to the DISPLAYTOSTANDARD function in the *DCALGOL Reference Manual*.

Table 2–1. Status Information Reported for SUBCLASS 1

MASK Bit	Description
0	MASK bit 0 is not used.
1	Bits [47:12] contain the stack number of the process. Bits [31:16] contain the JOBNUMBER attribute or 0. Bits [15:16] contain the MIXNUMBER attribute or 0.
2	For a compiler process, a link to the NAME task attribute of the process in standard form.
3	For compiler processes, a link to the NAME task attribute of the code file being compiled in standard form. For a process that is not a compiler process, a link to the NAME task attribute of the process in standard form.
4	PRIORITY task attribute.
5	Stack state, similar to the STATUS task attribute, but the numerical codes are different: • 2 = Terminating • 3 = Scheduled • 4 = Being initiated • 5 = Segment dictionary • 6 = To be continued • 7 = Frozen library • 8 = Waiting for event or task • 9 = Waiting for software interrupt (HOLDING) • 12 = Special wait (ASLEEP) • 13 = Ready for swap-in • 14 = Ready-queued • 15 = Processor is running program now For more information, refer to the system command Y (Status Interrogate) in the *System Commands Reference Manual*.
6	RSVP reply mask. If bit [47:01] is turned on, the process has an active RSVP.
7	Bit [47:01] is turned on for MCS. If the process is an MCS, bits [46:07] contain the MCS number.
8	1 if process has produced a display.

continued

Table 2–1. Status Information Reported for SUBCLASS 1 (cont.)

MASK Bit	Description
9	Bit [0:01] is turned on if the task is visible—that is, if it is not an *invisible* system process. Bit [1:01] is turned on if the process is suppressed from the mix picture. Refer to the system command SUPPRESS (Suppress Display) in the *System Commands Reference Manual*.
10	For a compiler process, if the compiler code file has a usercode, a link to that usercode in standard form.
11	For a compiler process, if the NAME task attribute of the code file being compiled has a usercode, a link to that usercode in standard form. For a process that is not a compiler process, if the NAME task attribute of the process has a usercode, a link to that usercode in standard form.
12	Link to the RSVP message. If the TYPE.GSRETURNRAWF bit is off, the link word refers to the plain message text. If the TYPE.GSRETURNRAWF bit is turned on, the link word refers to an area as follows: • Word 0 = Task mix number • Word 1 = Internal MCP message number • Word 2 = Start of untranslatable message
13	Link to the most recent DISPLAY message. The format is the same as for the RSVP message (MASK bit 12).
14	MASK bit 14 is reserved.
15	Bits [15:16] ORGUNIT or 0. Bits [39:16] DESTSTATION or 0.
16	Word of information concerning tape RESOURCE task attribute.
17	Word of information concerning tape RESOURCE task attribute.
18	Number of processes sharing a library or process stack.
19	Link to MCS name (in standard form) of the originating MCS. This is the MCSNAME attribute of the task.
20	Link to the station name in (standard form) that originated the task. This is the SOURCENAME attribute of the task.
21	Link to the IDENTITY code file attribute of the process in standard form.
22	MASK bit 22 is reserved.
23	Link to the USERCODE task attribute of the process in standard form.

continued

Table 2–1. Status Information Reported for SUBCLASS 1 (cont.)

MASK Bit	Description
24	Bit [0:01] contains the CHECKPOINTABLE task attribute. Bits [4:04] contain the BRCLASS task attribute. Bits [8:04] contain a code for the check point status displayed by the system command BR (Breakout): • 0 = Checkpointable • 1 = Checkpoint requested • 2 = Checkpoint running • 3 = Checkpoint running • 4 = Restarting (program) • 5 = Restarting (once only) • 6 = Restarting (multiple)
25	Link to the CHARGE task attribute code of the process in standard form.
26	Link to the ACCESSCODE task attribute of the process in standard form.
27	ORGUNIT task attribute.
28	DESTSTATION task attribute.
29	Count of actual segment descriptors (ASDs) in use by the process. For more information, refer to the system command ASDU (ASDU Usage) in the *System Commands Reference Manual*.
30	Count of maximum number of ASDs in use at one time by the process.
31	Word of information concerning the LIBRARYSTATE task attribute.
32	For a scheduled task, the TIME (14) value when the task was scheduled.
33	If bit 47 of the word is turned on, this word contains the mix number of the fast-tasking worker process. Otherwise, this word contains the mix number of the fast process.
34	If the process is a library stack, this word will return a list of the process stack numbers that are using the library. A link word is returned that points to a word in the array that contains the number of processes that are linked followed by the stack number of each process.

continued

Table 2–1. Status Information Reported for SUBCLASS 1 (cont.)

MASK Bit	Description
35	This bit returns one word of information as follows: • [0:01] = If turned on, bits [47:12] contain the job queue from which the task originated. • [01:01] = Is turned on if the stack is a database stack. • [02:01] = Is turned on if the parent job of the task is a session. • [03:01] = Is turned on if the task is a privileged user.

Table 2–2. Status Information Reported for SUBCLASS 2

MASK Bit	Description
0	MASK bit 0 is not used.
1	Bits [47:12] contain the stack number of the process. Bits [31:16] contain the JOBNUMBER task attribute or 0. Bits [15:16] contain the MIXNUMBER task attribute or 0.
2	MASK bit 2 is reserved.
3	Compiler information.
4	MAXPROCTIME task attribute.
5	ACCUMPROCTIME task attribute (in 2.4-microsecond units).
6	MAXIOTIME task attribute (in 2.4-microsecond units).
7	ACCUMIOTIME task attribute (in 2.4-microsecond units).
8	ACTIVETIME task attribute (in 2.4-microsecond units).
9	Time of day when process started (in terms of 2.4-microsecond intervals of time).
10	TARGET task attribute.
11	CORE task attribute.
12	Link to the RSVP message.
13	Link to the display message.
14	Words of memory assigned to the process.
15	Words of save memory assigned to the process.

continued

Table 2–2. Status Information Reported for SUBCLASS 2 (cont.)

MASK Bit	Description
16	For a scheduled process, the time (in 2.4-microsecond intervals of time) that the task has been in the schedule; otherwise, the ELAPSEDTIME task attribute (in 2.4-microsecond intervals of time).
17	Information concerning the parent, siblings, and descendents. Bits [35:12] contain the stack number of the parent or 0. Bits [23:12] contain the stack number of an older sibling or 0. Bits [11:12] contain the stack number of an offspring or 0.
18	TASKVALUE task attribute.
19	HISTORY task attribute.
20	Similar to STATUS task attribute: • 1 = If scheduled • 3 = If suspended • 2 = Otherwise
21	MASK bit 21 is reserved.
22	LOCKED task attribute.
23	STOPPOINT attribute.
24	MASK bit 24 is reserved.
25	Bits [15:16] ORGUNIT or 0. Bits [39:16] DESTSTATION or 0.
26	Bits [15:16] contain the STATION task attribute. Bits [33:02] contain the TANKING task attribute. Bit [46:01] contains the AUTOSWITCHTOMARC task attribute. Bit [47:01] contains the DISPLAYONLYTOMCS task attribute.
27	OPTION task attribute.
28	ERROR task attribute.
29	RESTART task attribute.
30	Link to the USERCODE task attribute in standard form.
31	Link to two words in the array. The first word contains the tag value of the stack cell. The second word contains the 48 data bits of the stack cell. This MASK bit corresponds to the system command OT (Inspect Stack Cell) <number>. The variable <number> is the number of the stack cell that is to be returned and is specified in ADDLWORD(2).

continued

Table 2–2. Status Information Reported for SUBCLASS 2 (cont.)

MASK Bit	Description
32	MASK bit 32 is reserved.
33	Size of the segment dictionary stack for the process (in words).
34	Amount of memory in use by the segment dictionary for the process for code and value arrays.
35	Amount of save memory in use by the segment dictionary for the process.
36	MASK bit 36 is reserved.
37	MASK bit 37 is reserved.
38	MASK bit 38 is reserved.
39	SAVECORELIMIT attribute.
40	Maximum amount of save memory used by the process.
41	MASK bit 41 is reserved.
42	Bit [0:01] is turned on if the process is a compiler.
43	Total time (in 2.4-microsecond intervals of time) that the process has spent in the ready queue. For more information, refer to the system command TI (Times) in the *System Commands Reference Manual*.
44	Link to four words of presence-bit-processing statistics. The first word is the processor time (in 2.4-microsecond intervals of time) used for initial presence-bit operations. The second word is a count of the number of initial presence-bit interruptions. The third word is the amount of processor time (in 2.4-microsecond intervals of time) used for presence-bit operations other than initial presence-bit operations. The fourth word is the count of other presence bits. For more information, refer to the system command TI (Times) in the *System Commands Reference Manual*.
45	Bit [0:01] contains the CHECKPOINTABLE task attribute. Bits [4:04] contain the BRCLASS task attribute.

continued

Table 2–2. Status Information Reported for SUBCLASS 2 (cont.)

MASK Bit	Description
	Bits [8:04] contain a code for the check point status displayed by the system command BR (Breakout): • 0 = Checkpointable • 1 = Checkpoint requested • 2 = Checkpoint running • 3 = Checkpoint running • 4 = Restarting (program) • 5 = Restarting (once only) • 6 = Restarting (multiple)
46	ORGUNIT task attribute.
47	DESTSTATION task attribute or 0.

General Results Returned for SUBTYPE 0, 1, 2, 4, and 8 Calls

GETSTATUS returns one of two kinds of result words in the array ARY for each process that is reported, soft error words and stack info pointer words.

Soft Error Word

If the GSERRORF bit ([47:01]) of the word is turned on, an error prevented GETSTATUS from reporting on the process. This bit distinguishes a soft error word from a stack info pointer word. The GSERRORVALUEF field ([46:08]) contains the soft error code (refer to Appendix B).

Stack Info Pointer Word

A stack info pointer word points to the information for the process and contains the stack number that identifies the process. Stack info pointer words are structured as follows:

Word	Field	Description
GSERRORF	[47:01]	The bit is turned off.
GSADDLINFOF	[46:08]	For SUBTYPEs 1 and 4, GETSTATUS makes the GSADDLINFOF field of the stack info pointer word equal to the GSADDLINFOF field of the corresponding stack or mix number request word that the caller originally supplied.

continued

continued

Word	Field	Description
GSLEVELF	[38:06]	For SUBTYPEs 2 and 8, GETSTATUS stores the relative level of descendent processes reported in the GSLEVELF field. The designated process and siblings of the process designated by the caller are reported as level 0, direct descendents of the designated process and direct descendents of its siblings are reported as level 1, second generation descendents are reported as level 2, and so forth.
XSLINKF	[32:17]	If this field is not 0, the value is the index or *base* of the validity mask, and the attribute list returned for the process follows the validity mask in the array ARY (refer to Figure 2–1). This field is not 0 if the MASK parameter has bits other than bit [0:01] turned on or if the call is for SUBTYPE 0 or SUBTYPE 4.
GSINFOF	[15:16]	This field contains the stack number of the process.

Figure 2–1 is a diagram of the information pointed to by the stack info pointer word.

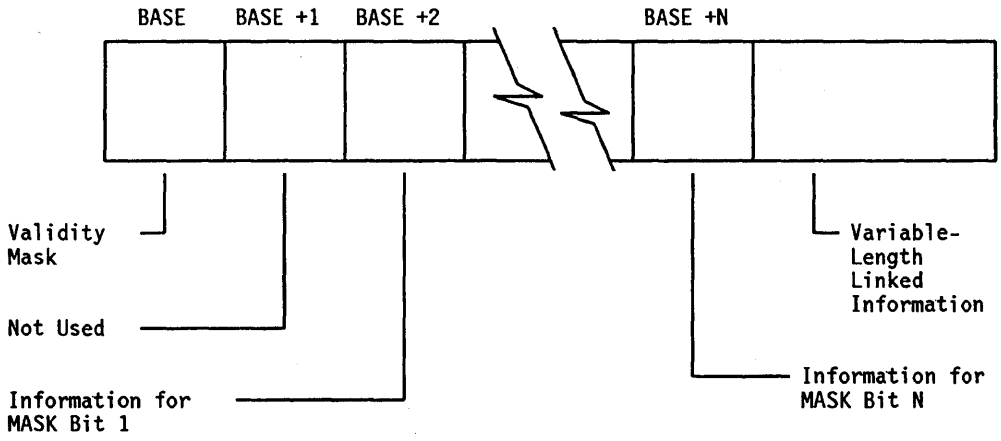

Figure 2–1. Information Pointed to by the Stack Info Pointer Word

In Figure 2–1, BASE is the value contained in the XSLINKF field of the stack info pointer word.

The validity mask has a bit turned on that corresponds to bits in the MASK parameter that are turned on for information that was successfully retrieved.

The variable-length linked information follows the last word referenced by the validity mask. GETSTATUS places information in this area for items described in the MASK tables (Table 2–1 and Table 2–2) with the phrase *link to*. Given the value of the link word, the caller can locate the linked information with an index calculated with one of the following formulas:

```
INDEX := ARY [BASE + (MASK bit number + 1)].XSLINKF + BASE;

INDEX := (link word).XSLINKF + BASE;
```

XSLINKF equals [32:17], and INDEX is the word index that locates the beginning of the information. For example, the link word for display messages (MASK bit 13) is contained in ARY [BASE + 14]).

SUBTYPE 0 and 4 Calls (Report for a List of Mix or Stack Numbers)

SUBTYPE 0 retrieves information for a designated list of mix numbers. SUBTYPE 4 retrieves information for a designated list of stack numbers.

Input for SUBTYPE 0 and 4 Calls

Figure 2–2 shows the format of input for SUBTYPE 0 and 4 calls:

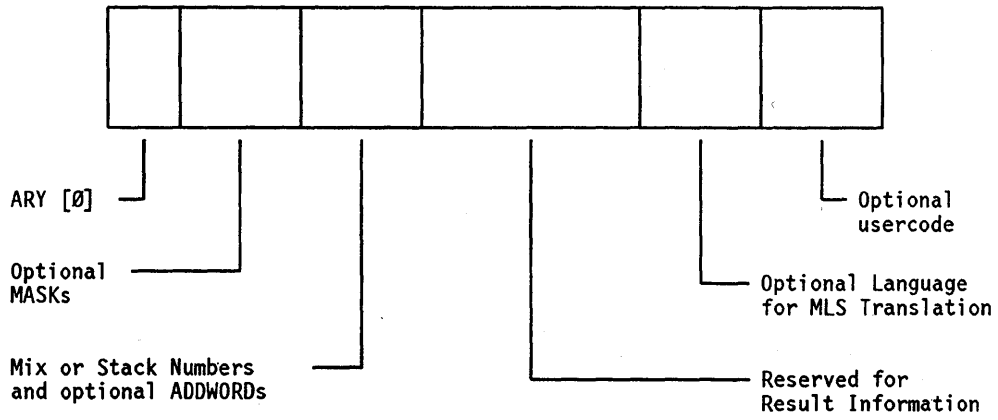

Figure 2–2. Input Format for SUBTYPE 0 and 4 Calls

For SUBTYPE 0 and 4 calls, the information in the array ARY should be structured as follows:

ARY	Description
0	The caller should store the index of the first word reserved for results in ARY[0].[19:20]. The results immediately follow the last mix or stack number or the optional ADDLWORD parameter. If the caller wants to use the optional dynamic MASK and SUBCLASS mechanism, he or she should store the index of the first word that contains a stack number or mix number in ARY[0].[39:20].

Optional Dynamic MASK and SUBCLASS Values

The caller can optionally supply dynamic MASK and SUBCLASS parameter values in the array. For example, dynamic MASK and SUBCLASS values could be used to retrieve different information for different processes. Or dynamic MASK and SUBCLASS values could be used to retrieve diverse information about the same process. That is, because the available information is divided by SUBCLASS into two sets of values (refer to Table 2–1 and Table 2–2), complete information can only be obtained by executing two separate GETSTATUS calls (one with each SUBCLASS value) or by placing the mix or stack number in the array twice and using the dynamic SUBCLASS mechanism.

Note that bit [46:01] (the sign bit) of the dynamic SUBCLASS values must be 0 (positive); the sign of the original SUBCLASS parameter is used to determine if invisible independent runners should be reported.

Mix Numbers or Stack Numbers and Optional ADDLWORDs

The caller should place the list of mix numbers or stack numbers beginning either at ARY[1] or at the word indexed by ARY[0].[39:20], if that value is greater than 0, for the information that is to be retrieved. These words should be structured as follows:

Word	Field	Description
GSINFOF	[15:16]	For SUBTYPE 0, this value is the mix number in binary. For SUBTYPE 4, this value is the stack number in binary.
GSVALUEF	[38:06]	The count of ADDLWORDs that apply for this mix or stack number. ADDLWORD(2) is used by GETSTATUS for SUBCLASS 2 calls if MASK bit 31 is ON (stack cell request).
GSADDLINFOF	[46:08]	The value is 0 or the index of the dynamic MASK and SUBCLASS pair to use. When the value in GSADDLINFOF is greater than 0 and the value in ARY[0].[39:20] is greater than 1, GETSTATUS uses the following method to replace the MASK and SUBCLASS parameters. In this example, suppose that the value of the GSADDLINFOF field is 3: MASK := ARY [3];

continued

continued

Word	Field	Description
		SUBCLASS := ARY [3 + 1]; After retrieving information for a stack or mix number, GETSTATUS locates the next stack or mix number in the array ARY by increasing the index with the following formula: AI := * + 1 + ARY [AI].GSVALUEF AI is used to index the next stack or mix number word in the array ARY. When AI exceeds the value in ARY[0].[29:20], GETSTATUS returns to the calling program.

Area Reserved for Result Information

The caller must reserve sufficient room in the array ARY after the last mix number or stack number so that GETSTATUS can return the information requested by the MASK bits. For example, if the highest MASK bit that is turned on is bit 15, then 17 words are needed for each mix or stack number and extra words are necessary to report variable-length *linked* information such as the RSVP message or process NAME. After copying the MLS language and usercode, or both that the caller placed at the end of the array, GETSTATUS might overwrite the last six words of the array with result information.

If enough space has not been reserved, GETSTATUS returns an indicator. If there were not enough words to completely report the status of the first (or only) process, GETSTATUS returns hard error 41 (refer Appendix A). If the complete status for some process has been reported and there were not enough words in the array to report the status of all the requested processes, GETSTATUS stores an indicator into the first word of the array, as follows:

- ARY[0].[47:01]

  This bit is turned on.

- ARY[0].[19:20]

  This field is the index of the word that contains the mix or stack number on which processing stopped. The information for the indexed process does not appear in the array, but complete information for preceding entries does appear.

Optional Language for MLS Translation

If the bit for TYPE.GSMLSLANGF is turned on, the array ARY contains the native language into which the MLS display and RSVP messages are to be translated. The language should be in substandard form. The language should start at word SIZE (ARY)–6 of the array.

Optional Usercode

If the bit for TYPE.GSORGUSERF is turned on, the report is limited to processes running under a designated usercode. The caller should place the usercode is substandard form in the last three words of the array ARY, starting at word SIZE (ARY)–3.

Results Returned for SUBTYPE 0 and 4 Calls

GETSTATUS stores in ARY[0].[19:20] the index plus 1 of the last stack number or mix number processed. If there were not enough reserved words to store the result information for all the processes requested, bit ARY[0].[47:01] is turned on.

GETSTATUS replaces each stack or mix number word by a soft error word or a stack info pointer word. The XSLINKF field of each stack info pointer word establishes the base at which the validity mask and process status information for that process is stored.

SUBTYPE 1 Calls (Range of Processes)

For SUBTYPE 1 requests, GETSTATUS returns information corresponding to MASK bits (as defined by the SUBCLASS parameter). GETSTATUS retrieves information corresponding to a sequence of stack numbers that start at a specific number.

Input

You should structure the input for SUBTYPE 1 requests as follows:

- ARY[0] should contain the count plus 1 of the maximum number of processes to be reported.
- ARY[1] should specify the starting stack number and the ending stack number of the stacks to be reported.

  ARY[1].[15:16] should contain the stack number from which to start or continue. If the value is 0, the report starts from the first stack in the system. Otherwise, GETSTATUS increases the number by 1 and starts the report from that point. GETSTATUS continues by increasing the stack number by 1, reporting that process, and so forth until the ending stack number is reached.

  When an increase in the stack number causes that number to exceed the highest valid stack number, the number wraps around, and the report continues from the first stack number. When the cycling of the stack number causes that number to equal the value in ARY[1].[32:17], GETSTATUS stops and returns to the caller.

  Note: *Information is not reported for either the beginning stack number or the ending stack number. Rather, information is reported only for the stacks between the beginning and ending numbers. All stacks can be searched by specifying 0 for the beginning and ending numbers on the first call and then by issuing continuation calls as necessary, with the beginning number equal to the last number reported by the previous call and the ending number equal to 0.*

Results Returned

For a SUBTYPE 1 request, GETSTATUS stores stack information pointer words in the array ARY beginning at ARY[1] up to ARY [ARY [0]].

Then GETSTATUS stores in ARY[0].[19:20] the index plus 1 of the last stack information pointer word. If ARY[0].[19:20] = 1, no stacks were reported.

If the limit of stacks specified by the caller in ARY[0] is reached before the stack number specified by the caller in ARY[1].[32:17] is reached or if there was not enough reserved space in the array to store all the information for the selected process, GETSTATUS turns on bit ARY[0].[47:01].

SUBTYPE 2 and 8 Calls (Process Families)

SUBTYPEs 2 and 8 return information corresponding to MASK parameter bits for selected processes, as determined by the SUBCLASS parameter. For SUBTYPE 8, GETSTATUS returns information for a designated process and for the descendents of that process. For SUBTYPE 2, GETSTATUS returns information about a designated process, the older siblings of the process, and all descendents of those siblings and all the descendents of the designated process. But if one of the siblings or descendents is running on a processor or is in the ready queue, GETSTATUS does not report any descendents of that process. The selection of the process is restricted by usercode if the parameter bit GSORGUSERF is turned on, and the selection is restricted further by the originating unit if the parameter field TYPE.GSORGUNITF is not 0 or if the caller is not privileged.

Input

The information in the array ARY should be structured as follows:

ARY	Description
0	This word designates the index plus 1 of the last word of the array that GETSTATUS can use to return stack info pointer words. GETSTATUS returns validity masks and status information for sibling and descendent tasks.
1	This word designates the mix number of the process. GETSTATUS reports that process and its descendents, and, for SUBTYPE 8, GETSTATUS reports the siblings of that process and their descendents.

Results Returned

For a SUBTYPE 1 request, GETSTATUS stores stack info pointer words in the array ARY beginning at ARY[1] up to ARY [ARY [0]]. Then GETSTATUS stores in ARY[0].[19:20] the index plus 1 of the last stack info pointer word. If ARY[0].[19:20] = 1, no stacks were reported.

If the limit of stacks specified by the caller in ARY[0] is reached before all the processes in the family were reported or if there was not enough reserved space in the array to store all the information for the selected process, GETSTATUS turns on bit ARY[0].[47:01].

SUBTYPE 3 and 5 Calls (Processes Scheduled for Execution)

The functions for SUBTYPE 3 have been replaced by those of SUBTYPE 5, so SUBTYPE 3 is not be described here.

SUBTYPE 5 returns a structured summary list of the following processes:

- Processes that are scheduled for execution
- Processes that are waiting for operator intervention
- Processes that are active
- Processes that are databases
- Processes that are libraries

These categories correspond to the system commands S, W, A, DBS, and LIBS respectively.

For each process reported, the returned summary information includes the mix number of the process, the stack number of the process, whether or not the process has a displayed message, and, if the process is a compiler, whether or not the compiler has detected a syntax error.

Input

For SUBTYPE 5, use MASK bits to indicate the lists of processes that GETSTATUS is to construct. These lists are shown in Table 2–3.

Table 2–3. Process Lists for SUBTYPE 5 Calls

MASK Bit	Process State
0	MASK bit 0 is not used.
1	Active entries.
2	Waiting entries (suspended processes).
3	Scheduled entries.
4	DBS entries.
5	Library entries.

If you turn on more than one of these MASK bits, GETSTATUS constructs lists for more than one set of processes. You should store in ARY[0] the index of the last word in the array ARY into which GETSTATUS can store results.

You can control the selection of processes reported as follows:

- If the sign bit (bit [46:01]) of the SUBCLASS parameter is turned off, GETSTATUS does not report independent runners.
- If TYPE.GSORGUSERF (bit [47:01] of the TYPE parameter) is turned on, GETSTATUS reports processes only if their USERCODEs are equal to the string supplied in the last three words of the array parameter.
- If TYPE.GSORGUNITF (bit [31:16] of the TYPE parameter) is not 0, GETSTATUS reports only processes with that ORGUNIT.
- If MASK bit [47.01] is turned on, GETSTATUS reports only processes that were started by an MCS whose MCSNUMBER is equal to the value supplied in ARY [SIZE (ARY) –5].
- If MASK bit [46:01] is turned on, GETSTATUS reports only processes that were started from the job queue whose queue number equals the value supplied in ARY [SIZE (ARY) –4].

You must provide enough words for all the processes to be reported, for extra words corresponding to MASK bits that are on, and for intervening MASK bits that are off or are not used.

Note: *You can safely calculate the number of necessary words by using the following formula:*

```
ARY [Ø] := FIRSTONE (MASK)
+ (maximum number of processes expected);
```

The maximum number of processes expected should be estimated with respect to any selection requested (such as usercode), but without regard to the MASK bits that are turned on.

Results Returned

GETSTATUS stores one word in the array ARY for each process reported. The words for processes with states corresponding to MASK bits that are turned on are linked together. The index of the first word in the list of active processes begins in ARY[1], the index of the first word in the list of suspended processes begins in ARY[2], and so forth. That is, the list for each MASK bit turned on begins in the word of the array that corresponds to the MASK bit number plus 1. If the index value in a head-of-list word is 1, the list is empty.

For each process reported, GETSTATUS returns a single word of information formatted as follows:

Field	Value	Description
[47:12]	Varies	Stack number of the process.
[35:01]	1	The process has displayed a message and the process is not a database stack (DBS).
[34:01]	1	The process is a DBS.
[33:01]	1	The process is a library.

continued

continued

Field	Value	Description
[32:01]	1	The process is a job or a DBS. The following structure applies: • Field [15:01] has a 1 if the process is a WFL job with offspring processes. • Field [14:15] contains the MIXNUMBER of the process or 0. Invisible independent runners do not always have a MIXNUMBER.
	0	The following values apply: • Field [15:04] contains the level of this process as a a descendent from its root parent (its parent's parent's parent). • If field [11:12] has a 0, the field contains the index of a word in the array ARY that contains a descriptor for a process of which this process is a descendent. The indexed word might be for the parent of the process or for the parent of the parent of the process (and so forth).
[31:02]		This field contains an indicator of the state that the process is in. The indicator is abbreviated because there are only two bits. Thus, the field contains one of the following values: • 0 = DBS entries • 1 = Active or library entries • 2 = Suspended entries • 3 = Scheduled entries
[29:01]	1	The process is a compiler that has detected a syntax error.
[28:01]	1	The process, which is not a DBS, is suspended.
[27:12]		If not 0, this field is the index of the word that describes the net process in the same list (suspended, DBS, and so forth).

SUBTYPE 6 Calls (Kind of Job)

For SUBTYPE 6, GETSTATUS reports the kind of job for a list of mix numbers.

Input

The caller should fill in ARY as follows:

ARY	Description
0	This word should contain the count of mix numbers provided.
1 to n	Each word should contain in binary the mix number of the process to report.

Results Returned

GETSTATUS replaces each mix number word with either a soft error word if the mix number is not in the range of 1 to 9999 or one of the code values in Table 2–4.

Table 2–4. Code Values for SUBTYPE 6 Calls

Code Value	Meaning
3	The process is a session.
2	The process has been initiated, and it is not a session.
1	The process is a WFL job that is in the job queue but has not been initiated.
0	There is no process with this mix number.

SUBTYPE 9 Calls (ACCUMPROCTIME and USERCODE for Stack Numbers)

For SUBTYPE 9 requests, GETSTATUS returns ACCUMPROCTIME and USERCODE for a list of stack numbers.

Input

The caller should set up the array ARY as follows:

ARY	Description
0	This word should contain the index plus 1 of the last word in the array that contains a stack number.
1 through n	Each word should contain a stack number.

Results Returned

GETSTATUS returns six words per stack number, starting at the word indexed by the value that the caller placed in ARY[0]. Before returning, GETSTATUS replaces ARY[0] by the index plus 1 of the last stack number word processed. If the array is not large enough, that is, if it does not contain at least (7 * ARY [0] + 1) words, GETSTATUS turns on bit [47:01] of ARY[0].

For each stack number word, GETSTATUS either replaces the word with a soft error word (with bit [47:01] turned on) or stores six words in the array that describes the status of the process running on the given stack. GETSTATUS places the six words for a requested stack number at the word given by the following method.

Suppose that N is the original value that the caller placed in ARY[0]. Then the six words of information for the stack number in ARY[1] begin at word N, the six words for the stack number in ARY[2] start at word N + 6, the six words for the stack number in ARY[3] start at word N + 2 * 6, and so forth.

GETSTATUS uses the above method even if some of the requested stack numbers yield soft error words, in which case GETSTATUS leaves the corresponding six words unused.

The six words of information for each stack reported are as follows:

Word	Description
0	Contains the validity mask. Bit [0:01] is turned on if the process is not scheduled and its ACCUMPROCTIME has been retrieved. Bit [1:01] is turned on if the process has a USERCODE.
1	Represents the ACCUMPROCTIME for processes in 2.4-microsecond intervals of time.
2 through 5	Represent the USERCODE of the process in standard form.

Section 3
GETSTATUS Request Type 2 (Miscellaneous Requests)

GETSTATUS miscellaneous requests are grouped under Request Type 2. These GETSTATUS calls retrieve information about the status of the system. For example, the controller uses these calls to generate displays for the system commands WM (What MCP), SL (Support Library), and OP (Options). For more information about these commands, refer to the *System Commands Reference Manual*.

To use miscellaneous calls, the calling process must have privileged status or must have been started from an ODT with a WFL *JOB* or *RUN* statement, but not with a primitive *??RUN* system command. Otherwise, a security error occurs and GETSTATUS returns hard error 43 (refer to Appendix A).

The general form of the GETSTATUS call is as follows:

```
B := GETSTATUS (TYPE, SUBCLASS, MASK, ARY);
```

Parameters

The following are Request Type 2 parameters.

TYPE

This parameter generally selects the specific case and subcase within the GETSTATUS intrinsic. For miscellaneous requests, the fields within the word are as follows:

- TYPE.TYPEF [7:08] = 2
- TYPE.SUBTYPEF [15:08]

Field [15:08] contains the SUBTYPE for Request Type 2.

The following information lists the various SUBTYPES. Not all numerical miscellaneous SUBTYPEs are assigned. Numbers not listed are invalid. In the following information, HYPERchannel® refers to the HYPERchannel adapter.

SUBTYPE	Description
0	Reserved.
1	Retrieves information about operations.

continued

HYPERchannel is a registered trademark of Network Systems Corporation.

continued

SUBTYPE	Description
2	Retrieves information about the hardware environment.
3	Retrieves information about system usage of statistics; refer to the system command U (Utilization) in the *System Commands Reference Manual*.
4	Retrieves information about system balancing parameters.
5	Reserved.
6	Reserved.
7	Reserved for internal use.
8	Retrieves reliability statistics for data link processors (DLPs), controllers, I/O processors (IOPs), or ports.
11	Retrieves information about library functions; refer to the SL (Support Library) system command description in the *System Commands Reference Manual*.
12	Retrieves information about operations as selected by MASK bits.
13	Retrieves information about the ASD factor; refer to the system command SF (Set Factor) in the *System Commands Reference Manual*.
14	Retrieves information about security administration.
18	Retrieves PS status information.
19	Retrieves memory dump type (MDT) information.
50	Reserved.
51	Retrieves HLUNIT information.
53	Retrieves change MCP information; refer to the system command CM (Change MCP) in the *System Commands Reference Manual*.
54	Retrieves BOOTCODE file information; refer to the system command MB (Make Boot) in the *System Commands Reference Manual*.
56	Retrieves information about autopower.
63	Retrieves information about the resident programs; refer to the system command RP (Resident Program) in the *System Commands Reference Manual*.
69	Retrieves information about data communications.
70	Retrieves information about HYPERchannel statistics.
71	Retrieves information about support libraries; refer to the system command SL (Support Library) in the *System Commands Reference Manual*.
72	Retrieves information about disk cache.
76	Retrieves information about selective logging; refer to the system command LOGGING (Logging Options) in the *System Commands Reference Manual*.

continued

continued

SUBTYPE	Description
78	Retrieves information about suppress warnings; refer to the system command SUPPRESSWARNING in the *System Commands Reference Manual*.
79	Retrieves information about MCS programs.
80	Retrieves information about a time zone.
81	Retrieves information about PrintS (print subsystem).
85	Retrieves information about attributes for dependent task accounting and file accounting; refer to the system command SYSTEMACCOUNTING (Resource Accounting) in the *System Commands Reference Manual*.

SUBCLASS

When SUBCLASS is required, its usage depends on the particular SUBTYPE. Therefore, SUBCLASS is explained later in the description of the SUBTYPE requests that use SUBCLASS.

MASK

For SUBTYPEs 1, 2, and 12, the MASK parameter is used to select the process attributes and status information to be retrieved by GETSTATUS. Each bit that is turned on in the MASK parameter instructs GETSTATUS to report the corresponding status item. Refer to Table 3–1, Table 3–2, and Table 3–3 for listings of valid MASK bits and their uses.

The use of some of the MASK parameter bits by SUBTYPEs 1, 2, and 12 is special and differs from the normal use of these bits for other Request Types (mix, unit, and directory). For miscellaneous request types, combinations of MASK bits should not be arbitrarily used. There are two reasons:

- Some MASK bits, such as MASK bit 38 for a SUBTYPE 1 call (MCS name or number), require special parameter information in the array ARY that might be overwritten by GETSTATUS if other MASK bits are turned on.
- At different installations, the use of some MASK bits might cause GETSTATUS to return a soft error word. When a soft error word is returned, it takes the place of the miscellaneous info pointer word so that none of the information requested by other MASK bits is available.

As a practical matter, if information for several MASK bits is wanted, the easiest thing to do is to issue individual GETSTATUS calls for each individual MASK bit.

Array ARY Parameter

For some SUBTYPEs, you must place parameter data such as names in the array ARY to indicate to GETSTATUS the information that is to be retrieved. In turn, GETSTATUS stores the retrieved information into the array ARY. The exact format of the array parameter and result information is described later under the description of each particular SUBTYPE.

Request Type 2 Calls by SUBTYPE

The following Request Type 2 (miscellaneous) calls are grouped by SUBTYPE because they are functionally related and share common characteristics.

SUBTYPE 1, 2, and 12 Calls (Operations and Hardware Information)

For SUBTYPE 1 and 12 calls, GETSTATUS returns information about operations. For SUBTYPE 2 calls, GETSTATUS returns information about the hardware environment.

Some of the information returned for subtype 1 calls corresponds to information displayed by the CF, CM, CS, DC, DL, HN, ID, OP, SF, SI, and WM system commands.

Some of the information returned for subtype 2 calls corresponds to information displayed by the CU, SC, PC, and GC system commands.

Some of the information returned for subtype 12 calls corresponds to information displayed by the WM, SYSTEMLANGUAGE, SECOPT, DRC, COMPILERTARGET, SEGARRAYSTART, and NETEX system commands.

Input for SUBTYPE 1, 2, and 12 Calls

Input information should be structured as follows:

Parameter	Value or Description
TYPE	Value = 2.
SUBTYPE	Value = 1, 2, or 12.
SUBCLASS	Value = 0.
MASK	Bits in the MASK parameter select the information that is to be returned. If you turn on more than one bit, you can retrieve information corresponding to all those bits with one call. For specific information about the information that can be returned by MASK bits, refer to Tables 3–1, 3–2, and 3–3.
ARY	ARY [0] = 2. ARY [1] = 0. The value for BASE is returned in ARY [1].[32:17]. BASE is defined in "Results Returned for SUBTYPE 1, 2, and 12 Calls."

Results Returned for SUBTYPE 1, 2, and 12 Calls

Figure 3–1 shows a diagram of the information returned in the array ARY for SUBTYPE 1, 2, and 12 calls. (In Figure 3–1, BASE is 2.)

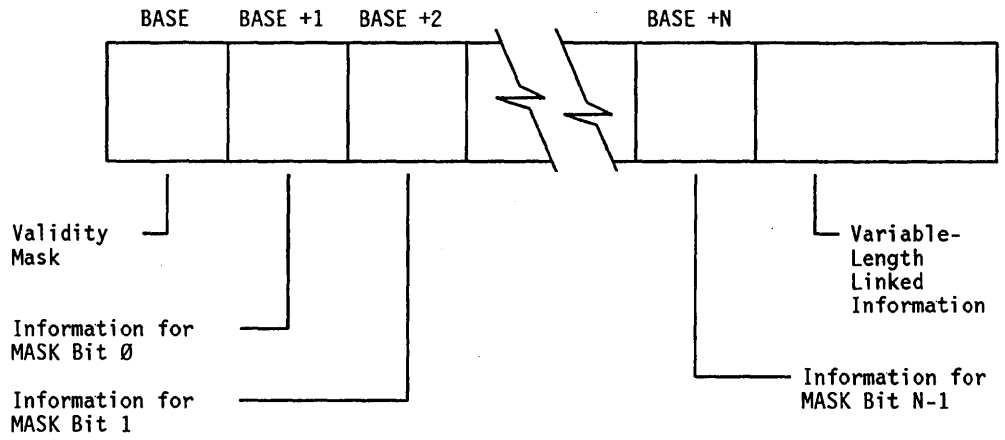

Figure 3–1. Format of Information Returned for SUBTYPE 1, 2, and 12 Calls

The validity mask has a bit turned on that corresponds to bits in the MASK parameter. These MASK bits are turned on for information that was successfully retrieved.

Each word after the validity mask (for which the corresponding validity mask bit is turned on) is either the status information retrieved or is a link to the status information retrieved.

The variable-length linked information follows the last word referenced by the validity mask. GETSTATUS places information in this area for items described with the phrase *link to* in the MASK tables (Tables 3–1, 3–2, and 3–3). Given the value of the link word, the caller can locate the linked information with an index calculated with one of the following formulas in which BASE equals 2:

```
INDEX := ARY [BASE + (MASK bit number + 1)].XSLINKF + BASE;

INDEX := (link word).XSLINKF + BASE;
```

XSLINKF = [32:17], and INDEX is the word index that locates the beginning of the information. For example, the link word for display messages (MASK bit 13) is contained in ARY [BASE + 14].

In Tables 3–2, 3–3, and 3–4, the link format for miscellaneous requests is as follows:

- GSINFOF [15:16]

  This word is the length in bytes of the item linked.

- XSLINKF [32:17]

  This word is the offset to the beginning information in the array ARY. When this offset is added to the BASE value of 2, the sum is the index of the information in the array.

Some of these links point to a name that is in standard form.

Besides regular links, some miscellaneous MASK bits return word links. A word link has the same field layout as a link, but the GSINFO field contains the length in words instead of bytes.

Table 3–1. Status Information Reported for SUBTYPE 1 Calls

MASK Bit	Description
0	OP options bit mask.
1	Contains the value 1 if the system command HS (Hold Schedule) is in force. Otherwise, the field contains 0.
2	Bits [19:20] contain the first valid stack number on the system. Bits [39:20] contain the last valid stack number on the system.
3	Contains the count of the number of processes in the system; frozen libraries are excluded.
4	The time base in 2.4-microsecond intervals of time. When this value is added to the time of day, the value gives the time since the system halt/load.
5	The current time of day in 2.4-microsecond intervals of time.
6	MASK bit 6 is reserved.
7	Working set factors: • Bits [15:16] contain FACTOR. • Bits [31:16] contain OLAYGOAL. • Bits [47:16] contain AVAILMIN. For more information, refer to the system command SF (Set Factor) in the *System Commands Reference Manual*.
8	Link to supervisor name in standard form. Refer to the system command CS (Change Supervisor) in the *System Commands Reference Manual*.
9	If a *CM* system command is pending, a link to the name of the next MCP in standard form.

continued

Table 3–1. Status Information Reported for SUBTYPE 1 Calls (cont.)

MASK Bit	Description
10	Link to the intrinsics name in standard form.
11	Link to the disk location information. The disk location information is controlled by the SUBCLASS parameter as follows: • 0 = OVERLAY • 1 = LOG • 2 = BACKUP • 3 = USERDATA • 4 = JOBS • 5 = CATALOG • 6 = DPFILES • 7 = SORT • 8 = IPFILES The first word is the number of family name entries. Each entry consists of four words formatted as follows: Bit [47:01] of the first word is turned on if no family name has been designated. The next three words contain designated family names in substandard form. For SUBCLASS 0 (overlay) more than one 4-word entry can be returned; for all the other SUBCLASS values, only one entry is returned.
12	Halt/load backup unit numbers (for duplicated or triplicated MCPs).
13	Date.
14	Time of day in 2.4-microsecond intervals of time.
15	Link to the data communications processor (DCP) prefix in standard form.
16	Link to the MCP names in standard form. Bits [46:08] of the link word contain the box mask for which the MCP names are being reported. When more than one bit is on in this field, more than one name is returned.
17	Link to the MCP compiler control ($) options that were set when the MCP was compiled.
18	Automatic backup information.
19	MCP level.
20	Link to the next DCP prefix in standard form (to be used after the next halt/load).
21	Link to the halt/load reason; two words in Burroughs Common Language (BCL) characters.

continued

Table 3–1. Status Information Reported for SUBTYPE 1 Calls (cont.)

MASK Bit	Description
22	Halt/load unit number.
23	MASK bit 23 is reserved.
24	Link to substitute backup information.
25	Catalog level.
26	System serial number in binary.
27	Link to the hub map.
28	Mask of types of controlware saved or –1.
29	Link to failure analysis summary.
30	Memory priority; refer to the system command SF (Set Factor) in the *System Commands Reference Manual*.
31	Plus (+) or minus (–) the BUFFERGOAL. If the value is negative, then the system is using the default BUFFERGOAL.
32	State bits. Bits [47:02] are a code value of the system AUTORESTORE archiving option: • 0 = DONTCARE • 1 = NEVER • 2 = YES
33	Link to dump name and HLDUMPDISK information.
34	MASK bit 34 is reserved.
35	MASK bit 35 is reserved.
36	MCP level: • [47:08] contains the Mark digit in binary. • [39:08] contains the level in binary. • [31:16] contains the cycle number in binary. • [15:16] contains the patch number in binary.
37	Returns the MCS name or MCS number. MASK bit 37 is special and cannot be used in conjunction with other MASK bits. Refer to "SUBTYPE 1 MASK Bit 37 (MCSNAME and MCSNUMBER Conversion)" immediately following this table for details on how to use this MASK bit.
38	Link to the configuration file name in standard form.
39	Link to the HOSTNAME in standard form.

continued

Table 3–1. Status Information Reported for SUBTYPE 1 Calls (cont.)

MASK Bit	Description
40	Running status of BNA. The value is 1 if BNA is running and 2 if BNA is not running.
41	Link to the GROUP ID name in standard form.
42	Link to the host usercode in standard form.
43	Reserved.
44	Link to two MCP code file status words. The first word contains the MCP compile time and date. The second word contains the MCP compiler information.
45	Link to the next group HOSTNAME in standard form.
46	MASK bit 46 is reserved.
47	MASK bit 47 is reserved.

SUBTYPE 1 MASK Bit 37 (MCSNAME and MCSNUMBER Conversion)

A special method is used to convert MCSNAME to MCSNUMBER values and MCSNUMBER to MCSNAME values.

Input for MCSNAME-to-MCSNUMBER Conversion

Use the following input for array ARY to convert MCSNAME to MCSNUMBER values:

ARY	Value or Description
ARY [0]	2
ARY [1]	0
ARY [3]	<mcsname> in standard form

Call

```
B := GETSTATUS (2 & 1 [15:18] & 1 [46:Ø1], Ø, Ø & 1 [37:Ø1], ARY);
```

Results Returned for MCSNAME-to-MCSNUMBER Conversion

GETSTATUS returns the MCSNUMBER value in array ARY [2 + 37 + 1].
GETSTATUS returns errors in the normal manner if no MCS exists with that name or if data comm is not active.

Input for MCSNUMBER-to-MCSNAME Conversion

Use the following input for array ARY to convert MCSNUMBER to MCSNAME values:

ARY	Value or Description
ARY [0]	2
ARY [1]	0
ARY [3]	<mcsnumber>

Call

```
B := GETSTATUS (2 & 1 [15:18] & Ø [46:Ø1], Ø, Ø & 1 [37:Ø1], ARY);
```

Results Returned for MCSNUMBER-to-MCSNAME Conversion

The MCSNAME is returned in standard form in the following array:

```
ARY [2 + ARY [2 + 37 + 1].[32:17]]
```

Errors are returned in the normal method if no MCSNAME exists with that number or if data comm is not active.

Table 3–2. Status Information Reported for SUBTYPE 2 Calls

MASK Bit	Description
0	Microcode version or 0.
1	Bit mask of processors.
2	Bit mask of I/O processors.
3	Maximum valid MCP unit table index.
4	MASK bit 4 is reserved.
5	Word link to memory and buffer status information.
6	Reserved.
7	Overlay size.
8	Word link to memory status information.
9	MASK bit 9 is reserved.
10	Nonstandard link to cache status; refer to the system command GC (Group Configuration) in the *System Commands Reference Manual*.
11	Memory address of the D0 register.
12	MASK of memory modules or 0.

continued

Table 3–2. Status Information Reported for SUBTYPE 2 Calls (cont.)

MASK Bit	Description
13	Link to I/O channel status information.
14	MASK bit 14 is reserved.
15	Link to CPM microcode versions.
16	MASK bit 16 is reserved.
17	Link to module status.
18	MASK bit 18 is reserved.
19	MASK bit 19 is reserved.
20	Link to group unit information.
21	MASK bit 21 is reserved.
22	MASK bit 22 is reserved.
23	Word link to the configuration table.
24	MASK bit 24 is reserved.
25	MASK bit 25 is reserved.
26	Link to the configuration table.
27	Link to 900 words of formatted configuration information.
28	Link to MSM information or to page information.
29	MASK bit 29 is reserved.
30	MASK bit 30 is reserved.
31	MASK bit 31 is reserved.
32	Word link to memory module information. Refer to "SUBTYPE 2 MASK Bit 32 (Memory Module Information)" immediately following this table.
33	Link to list of temporary units.
34	Link to information about the HYPERchannel map being used in the current configuration. You can use this information to determine the names that are associated with the HYPERchannel units: • The first word contains the number of adapter specifications. • The information for the adapters starts in the second word. Each adapter specification is 64 words long. Refer to the HYPERchannel map for details.
35	For task control processor (TCP) systems, the bit mask of TCPs.
36	Link to two words of TCP microcode-level information.
37	Link to three words of memory page assignment status.

SUBTYPE 2 MASK Bit 32 (Memory Module Information)

Figure 3–2 shows the structure of the information returned by this call.

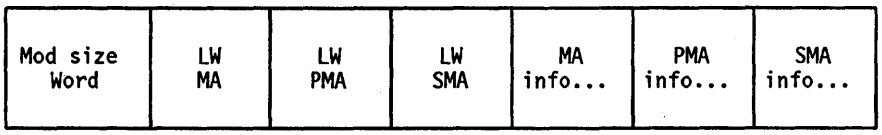

Figure 3–2. Structure of Information Returned for SUBTYPE 2 MASK Bit 32

LWMA is the link word to the memory module bit vector MA, LWPMA is the link word to the memory module bit vector PMA, and LWSMA is the link word to the memory module bit vector SMA.

Figure 3–3 shows the structure of the memory module bit vectors MA, PMA, and SMA returned to the caller.

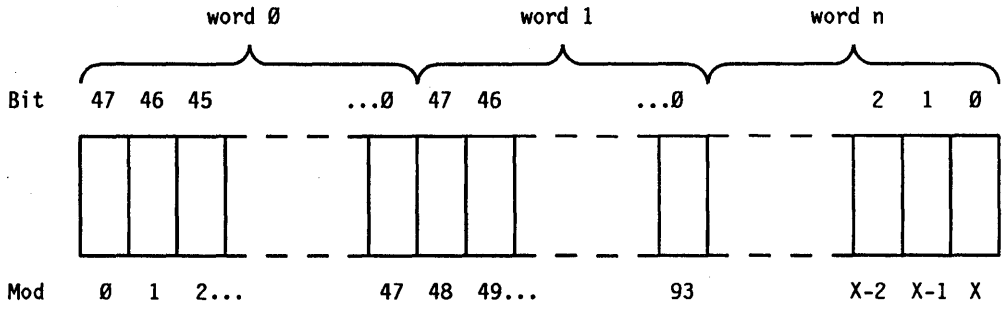

Figure 3–3. Structure of Memory Module Bit Vectors MA, PMA, and SMA

If a bit in an MA bit vector is turned on, the corresponding memory module is in-use by the MCP. If a bit in a PMA bit vector is turned on, the corresponding memory module is present (that is, it was found by the last halt/load). If a bit in an SMA bit vector is turned on, the corresponding memory module is has been saved by the SV (Save) system command.

The format used to return information in array A is as follows:

Word	Field	Name and Meaning
0		MOD SIZE WORD
	19:20	Contains the memory module size in units of K words.
1		LWMA
	19:20	Contains the index to the bit vector MA (shown below as word number 4).
	39:20	Contains the length of the bit vector MA in words.

continued

continued

Word	Field	Name and Meaning
2		LWPMA
	19:20	Contains the index to the bit vector PMA (shown below as X).
	39:20	Contains the length of the bit vector PMA in words.
3		LWSMA
	19:20	Contains the index to the bit vector SMA (shown below as Y).
	39:20	Contains the length of the bit vector SMA in words.
4		This word is the starting word of the bit vector MA.
X		This word is the starting word of the bit vector PMA.
Y		This word is the starting word of the bit vector SMA.

SUBTYPE 2 MASK Bit 34 (HYPERchannel Map)

Each adapter specification consists of 64 words. It is divided into 4-word logical pieces of two types, each of which is defined below. Indexes into the ARY for each of these types for any of the adapter specifications are determined by using the following algorithms:

- ASPEC = BASE + LINK + 1 + (n*64)

  ASPEC is the starting index of relative information of the *nth* adapter specification, where BASE is 2 and where LINK is the value extracted from the XSLINKF [32:17] field of the result word returned in ARY [BASE + 34 + 1].

- USPEC = ASPEC + 4 + (4*m)

  USPEC is the starting index of *mth* logical unit specification within the *nth* adapter specification, where m is a number in the range 0 to the last logical unit designated for the adapter.

The first four words of an adapter specification contain adapter relative information. They are formatted as follows:

- ARY [ASPEC] through ARY [ASPEC + 2] of the adapter name in substandard form.
- ARY [ASPEC + 3].[7:8]

  The trunk address of the adapter.

- ARY [ASPEC].[47:1]

  If turned on, a valid entry.

- ARY [ASPEC].[15:8]

  The number of logical unit specifications for the adapter.

The remaining 60 words of an adapter specification are divided into 4-word entries, each called a logical unit specification. For the A222 adapter, there can be only one unit specification, which corresponds to the one unit that is available on the system. Each logical unit specification is formatted as follows:

- ARY [USPEC] through ARY [USPEC + 2] of the unit name in substandard form.
- ARY [USPEC+3].[47:1]

  If turned on, a unit specification exists for the adapter.
- ARY [USPEC].[46:1]

  If turned on, trunk information exits.
- ARY [USPEC].[45:1]

  If turned on, write partner information exists.
- ARY [USPEC].[44:1]

  If turned on, read partner information exists.
- ARY [USPEC].[43:8]

  Logical unit number. For an A222 adapter, this number can be in the range 0 to 63, inclusive.
- ARY [USPEC].[35:4]

  Trunk mask.
- ARY [USPEC].[31:8]

  Trunk address of the read partner.
- ARY [USPEC].[23:8]

  Logical address of the read partner.
- ARY [USPEC].[15:8]

  Trunk address of the write partner.
- ARY [USPEC].[7:8]

  Logical address of the write partner.

In Table 3–3, NETEX® refers to software that enables different computers or workstations with different operating systems to communicate over a HYPERchannel network.

NETEX is a registered trademark of Network Systems Corporation.

Table 3–3. Status Information Reported for SUBTYPE 12 Calls

MASK Bit	Description
0	Link to the halt/load and backup halt/load unit number list for the processor on which the caller is running. The first word of information contains a count in [47:8] and a bit mask. The remaining words contain unit numbers of the halt/load and backup halt/load units (disk units with duplicated or triplicated copies of the MCP).
1	Unit number of the halt/load unit.
2	Link to the backup halt/load unit number list—in the same form as for MASK bit 0, except the active halt/load unit number is not returned.
3	Default compiler target.
4	MASK bit 4 is reserved.
5	Link to SEGARRAYSTART information.
6	Reserved.
7	Returns the SYSTEMLANGUAGE value. MASK bit 7 is special and cannot be used in conjunction with other MASK bits. Refer to "SUBTYPE 12 MASK Bit 7 Calls (System Language)" immediately following this table for details on how to retrieve the SYSTEMLANGUAGE value. This call is being deimplemented on a future release. The preferred method is to use Mask bit 17.
8	Link to the next HOSTGROUP name that is to take effect after the next halt/load and the current hostname. The names are in substandard form. The format is as follows: The first word is reserved. The next three words contain the next HOSTGROUP name or 0. The last three words contain the current HOSTGROUP name.
9	A link to the current and next BNA version; refer to the system command BNAVERSION in the *System Commands Reference Manual*. In the information returned, word 1 contains the next BNA version and word 2 contains the current BNA version. Numerical values are 1 for BNA Version 1 and 2 for BNA Version 2.
10	0 if BNA V2 is in isolated mode. 1 if BNA V2 is in networking mode.
11	A nonstandard link to the list of languages that are available in the MCP for system messages translated by the MLS. XSLINKF [32:17] contains the offset to the start of the list; GSINFOF [15:16] contains the length of the list in words. The first word of the list contains a count of the number of languages available in the MCP. Starting at the second word of the list, there are three words for each language bound into the MCP. Each language name appears in substandard form.
12	Information about the initialization file to be used the next time NETEX software is initialized:

continued

Table 3–3. Status Information Reported for SUBTYPE 12 Calls (cont.)

MASK Bit	Description
	If the XSLINKF field [32:17] is not equal to 0, it is the offset to the title of the initialization file in standard form. The GSINFOF field [15:16] contains a title indicator code: • 1 = NULL • 2 = DEFAULT • 3 = SPECIFIED TITLE
13	Security options word: • Bit [00:01] = 1 (Security administrator is authorized.) • Bit [01:01] = 1 (Program dump filtering is turned on.) • Bit [02:01] = 1 (Mandatory disk scrubbing is turned on.) • Bit [03:01] = Nonuser file security: – 0 = PUBLIC – 1 = PRIVATE • Bit [04:01] = 1 (All hosts are restricted.) • Bit [05:01] = 1 (Backup files have usercodes of owners.) • Bit [07:01] = Reserved for future use. • Bits [11:04] = Security class: – 0 = Class U (unspecified) – 1 = Class S0 – 2 = Class S1 – 3 = Class S2 • Bits [13:02] = Current tape security: – 0 = NONE – 1 = AUTOMATIC • Bits [15:02] = Tape security will be the following: – 0 = NONE – 1 = AUTOMATIC • Bits [18:03] = Password management: – 0 = MINIMAL – 1 = GENERATED

continued

Table 3–3. Status Information Reported for SUBTYPE 12 Calls (cont.)

MASK Bit	Description
	• Bits [27:09] = Reserved for future use. • Bits [30:03] = An integer in the range 0 to 6, inclusive; the CANDE *LAISSEZFILE* option is turned on. • Bits [31:01] = Reserved for future use. • Bits [32:01] = 1 (CANDE *DIALLOGIN* option is turned on.) • Bits [33:01] = 1 (CANDE *ALLLOGIN* option is turned on.) • Bits [34:01] = 1 (CANDE *SECDIALIN* option is turned on.) • Bits [35:01] = 1 (CANDE *SECPSEUDO* option is turned on.) • Bits [36:01] = 1 (CANDE *SECALL* option is turned on.) • Bits [37:01] = 1 (CANDE *USECOMSPRIV* option is turned on.) • Bits [46:09] = Reserved for future use. • Bits [47:01] = 1 (InfoGuard is authorized). If security administrator status is authorized on the system, the call can be invoked by a security administrator usercode or any privileged user. If security administrator status is not authorized on the system, the call can be invoked by any privileged user. For more information about security, refer to the SECOPT (Security Options) system command in the *System Commands Reference Manual*.
14	Disk resource control (DRC) subsystem information. Bit [7:08] contains the DRC subsystem state: • 0 = Inactive • 1 = DRC system will be started after the next halt/load • 2 = Initializing • 3 = Active • 4 = Terminating
15	Reserved.
16	Reserved
17	Returns a link word that points to the system language. The format of the data is 1 byte for length followed by the language.
18	Returns the system CCSVERSION file attribute.
19	Returns a link word that points to the system convention. The format of the data is 1 byte for length followed by the convention name, such as ASERIESNATIVE.

SUBTYPE 12 MASK Bit 7 Calls (System Language)

Note: This call will be deimplemented in a future release. Programs that use this call in the Mark 3.9 software release will receive a warning message. It is better to use MASK bit 17 to return the system language.

Input

A special method is used to return the SYSTEMLANGUAGE. Use the following input for array ARY:

ARY	Value or Description
ARY [0]	5
ARY [1]	0

Call

```
B := GETSTATUS (2 & 12 [15:Ø8], Ø, Ø & 1 [7:Ø1], ARY);
```

Results Returned

The SYSTEMLANGUAGE is returned in substandard form, starting at the pointer given by POINTER (ARY [1]) + 5.

See the examples of subtype 1, 2, and 12 GETSTATUS calls at the end of this section.

SUBTYPE 11 Calls (System Library)

For subtype 11 calls, GETSTATUS returns the title of a support library. GETSTATUS is passed the name of the function library in substandard form and returns title in standard form if the current support library is a code file that is not the active MCP. See also SUBTYPE 71.

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	2
ARY [2] through ARY [5]	Library function name in substandard form

Call

```
B := GETSTATUS (2 & 11 [15:Ø8], 2, Ø, ARY);
```

Results Returned

If there are no errors, word 1 of the array contains a word index to the standard form title as defined by the SL (Support Library) system command.

SUBTYPE 13 Calls (ASD Tables)

For SUBTYPE 13 calls, GETSTATUS returns information about the ASD table.

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	2
ARY [1]	20

Call

```
B := GETSTATUS (2 & 13 [15:08], Ø, Ø, ARY);
```

Results Returned

Information is returned in the following format:

ARY Word	Contents
0	2
1	0 or soft error code.
2	The factor used for calculating the length of the ASD table at the last halt/load.
3	The factor used for calculating the length of the ASD table at the next halt/load.
4	Field [19:20] contains the total number of ASDs.
5	The percentage of ASDs currently in use.
6	The percentage of ASDs used at one time since the last halt/load.
7	Current memory size.
8	Next memory size.

SUBTYPE 14 Calls (Security Administrator Status)

For SUBTYPE 14 calls, GETSTATUS returns information about security administrator status such as the designated session, job, or task. This status is determined by the usercode associated with the job, task, or session. A usercode has security administrator status if the SECADMIN option is designated for the usercode in the USERDATA file. There is no corresponding system command for this GETSTATUS call.

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	2
ARY [1]	1
ARY [2]	Session number or mix number of the job or task

Call

```
B := GETSTATUS (2 & 14 [15:Ø8], Ø, Ø, ARY);
```

Results Returned

The status of the specified session, job, or task is returned in word 1 of the array ARY:

- If word 1 equals 0, the session, job, or task does not have security administrator status.
- If word 1 equals 1, the session, job, or task does have security administrator status.

SUBTYPE 15 Calls (Network Status)

For SUBTYPE 15 calls, GETSTATUS returns information about the status of the network as a state, initialization information, and the network name. The corresponding system commands for this GETSTATUS call are the *NET* (BNAv1) command, the *BNA* (BNAv2) command, and the *X25* (X25MCS) command.

Input

Use the following input information for ARY:

ARY	Value or Description
ARY [0]	2
ARY [1]	0
ARY [3]	Indicates which network is to be queried: If ARY [3].[47:01] = 0, then ARY [3] = the network index of the MCP: 1 = BNAv1, 2 = BNAv2, 3 = X25 If ARY [3].[47:01] = 1, then ARY [3].[03:04] = the offset in words to the network name (in substandard form). ARY [ARY[3].[03:04]] contains the network name.

Call

```
B := GETSTATUS (2 & 15 [15:Ø8], Ø, Ø, ARY);
```

Results Returned

GETSTATUS returns information in the following format:

ARY Word	Contents
ARY [4] through ARY [6]	Network name in substandard form
ARY [7].[47:08]	Reserved for future use
ARY [7].[39:04]	Network state: • 0 = NOT RUNNING • 1 = INITIALIZING • 2 = FROZEN • 3 =TERMINATING
ARY [7].[35:04]	Network unit information: • 1 = *NULL • 2 = *DEFAULT • 3 = <file title> found in ARY[11] in display form
ARY[7].[31:16]	Reserved for future use
ARY[7].[15:08]	Controller port index of the network
ARY[8] through ARY[10]	The primary library function name
ARY[11] through ARY[53]	The next NETINIT file title in display form

Examples

To inquire on a network by its network index:

```
ARY[Ø] := 2:
ARY[1] := Ø;
ARY[3] := NETINDEX;     % 1 = BNAv1, 2 = BNAv2, 3 = X25
GETSTATUS (2 & 15 [15:Ø8], Ø, Ø, ARY);
```

To inquire on BNAv2 with network name:

```
ARY[Ø] := 2;
ARY[1] := Ø;
ARY[3] := Ø & 1[47:Ø1] & 4[Ø3:Ø4];
REPLACE POINTER (ARY[4]) BY 48"Ø5", 8"BNAV2.";
GETSTATUS (2 & 15 [15:Ø8], Ø, Ø, ARY);
```

SUBTYPE 18 Calls (Print Information)

For SUBTYPE 18 calls, GETSTATUS returns information about printing, such as the number of requests that are printing, waiting, waiting for forms, and so on. This GETSTATUS call corresponds to the system command *PS STATUS*.

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	2

Call

```
B := GETSTATUS (2 & 18 [15:Ø8], Ø, Ø, ARY);
```

Results Returned

The array returned by GETSTATUS contains the following information:

Word	Description
Word 3	Number of requests printing
Word 4	Number of requests waiting
Word 5	Number of requests waiting for forms
Word 6	Number of requests in an exception state
Word 7	Number of requests that are stopped

SUBTYPE 19 Call (Memory Dump Type)

For SUBTYPE 19 calls, GETSTATUS returns the memory dump type setting (refer to the MDT system command).

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	2

Call

```
B := GETSTATUS (2 & 19 [15:Ø8], Ø, Ø, ARY);
```

Results Returned

The array returned by GETSTATUS contains the following information:

ARY Word	Value or Description
2	Code value: • 0 = COMPLETE • 1 = ALLINUSE • 2 = PARTIAL

SUBTYPE 51 Calls (Halt/Load Unit)

The SUBTYPE 51 call corresponds to the system command HLUNIT.

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	2

Call

```
B := GETSTATUS (2 & 51 [15:08], Ø, Ø, ARY);
```

Results Returned

GETSTATUS returns information in the following format:

ARY Word	Contents
ARY [0]	Number of entries * 2
ARY [1] through ARY [ARY[0]–1]	Each entry is a pair of words, as follows: • ARY [AI] = Unit number • ARY [AI + 1] = PROCMASK (A 15 systems only)

For A 10 dual processor systems, the results stored in ARY are as follows:

ARY Word	Contents
ARY[0]	Number of entries * 2 + 16.
ARY [1] through ARY [Number of entries * 2 – 1]	Each entry is a pair of words, as follows: • First word = Unit number. • Second word [15:16] = PROCMASK. • ARY [ARY[0] – 16] for 16 = The last 16 words contain halt/load information that is used after reconfiguration, indexed by the processor number offset by A [0] – 16 (for example, A[0] – 16 + 10 equals halt/load unit information for processor 10 after reconfiguration). The format for each word is as follows: – [47:32] = Undefined – [15:16] = Unit number

SUBTYPE 54 Calls (BOOTCODE File)

For SUBTYPE 54 calls, GETSTATUS returns information about the selection of a BOOTCODE file during system initialization. This call has two SUBCLASS types.

SUBCLASS 0 Calls (MB)

Use this type of SUBCLASS call for the equivalent of the system command MB (Make Boot). For additional information, refer to the *System Commands Reference Manual*.

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	2

Call

```
B := GETSTATUS (2 & 54 [15:Ø8], Ø, Ø, ARY);
```

Results Returned

GETSTATUS returns information in the following format:

ARY Word	Contents
ARY [1].[23:12]	Last valid word in the array row.
ARY [1].[11:12]	Number of entries.
ARY [2] through ARY [ARY [1].[11:12]+1]	Each entry is as follows: • [47:08] = Family index. • [38:07] = Priority number. • [31:04] = Processor ID. • [27:12] = Pointer to the standard form file title. • [15:12] = Pointer to the family name. • [02:01] = Family index information is present. • [01:01] = A BOOTCODE file is on the unit. • [00:01] = The file is active.
ARY [ARY [1].[11:12]+2] to the end	File titles and family names in standard form.

SUBCLASS 1 Calls (MB ON <family name>)

Use this type of SUBCLASS call for the equivalent of the system command MB ON <family name>. For further information about the MB system command, refer to the *System Commands Reference Manual*.

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	2
ARY [1]	2
ARY [2] through ARY [5]	Family name in substandard form

Call

```
B := GETSTATUS (2 & 54 [15:Ø8], 1, Ø, ARY);
```

Results Returned

GETSTATUS returns information in the following format:

ARY Word	Contents
ARY [1]	Pointer to file title in standard form.
ARY [2]	Entry information, as follows: • [47:08] = Family index. • [38:07] = Priority number. • [31:04] = Processor ID. • [27:12] = Pointer to the file title in standard form. • [15:12] = Pointer to the family name. • [02:01] = Family index information is present. • [01:01] = A BOOTCODE file is on the unit. • [00:01] = The file is active.
ARY [3]	MB timestamp.
ARY [4]	MCP update timestamp.
ARY [5]	Microcode level information.
ARY [ARY [1]] to the end	Standard form file titles and family names.

SUBTYPE 56 Calls (Automatic Power Schedule)

For SUBTYPE 56 calls, GETSTATUS returns information about the automatic power schedule for A 1, A 2, A 3, A 4, and A 5 machines. This call has two SUBCLASS types.

SUBCLASS 1 Calls

A SUBCLASS 1 call is a query about the automatic power schedule.

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	8

Call

```
B := GETSTATUS (2 & 56 [15:Ø8], 1, Ø, ARY);
```

Results Returned

For SUBCLASS 1 calls, the response in ARY contains a schedule for an entire week in 7 words. Each word contains information for one day. Word 1 of the array ARY is 0, Word 2 contains the entry for Sunday, Word 3 contains the entry for Monday, and so on. The format of each word is as follows:

Word	Description
[47:06]	Reserved for future use
[41:01]	On-time valid indicator
[40:01]	Off-time valid indicator
[39:08]	Power off delay time in minutes
[31:08]	On-time hour
[22:08]	On-time minute
[15:05]	Off-time hour
[07:08]	Off-time minute
[47:48]	0 if this function is not supported

SUBCLASS 2 Calls

A SUBCLASS 2 call is a mode check request.

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	8

Call

```
B := GETSTATUS (2 & 56 [15:Ø8], 2, Ø, ARY);
```

Results Returned

For SUBCLASS 2 calls, the response in Word 1 of the array ARY contains the mode check word, as follows:

Word	Description
[47:08]	Reserved for future use
[39:08]	Power-off time remaining. If MODE equals any type of power-off pending, this field contains the number of minutes left before power is actually shut off. For other modes, this field is 0.
[07:08]	MODE indicator, as follows: • [05:01] = Blower failure warning • [04:01] = Thermal overload warning • [03:01] = Thermal overload power-off pending • [02:01] = Power clock master • [01:01] = Scheduled power-off pending • [00:01] = Unscheduled power-off pending

SUBTYPE 63 Calls (Resident Program)

For SUBTYPE 63 calls, GETSTATUS returns a list of the programs that have been subjected to the RP (Resident Program) system command.

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	2

Call

```
B := GETSTATUS (2 & 63 [15:Ø8], Ø, Ø, ARY);
```

Results Returned

A[1].[32:17] is the index to the start of data in array ARY. If you index the value of A[1].[32:17], GETSTATUS returns the following:

Word	Description
Word 0	Number of resident programs
Word 1	Number of words for this entry
Word 2	Number of running copies of the program
Word 3 through end	Program name

If there are more resident programs, entries continue by repeating Words 1 through 3 for the remaining number of resident programs.

SUBTYPE 69 Calls (Initialize Data Comm)

For SUBTYPE 69 calls, GETSTATUS returns information about the data comm subsystem. This call is equivalent to the ID system command.

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	2
ARY [1]	0

Use the following MASK bits to obtain information:

MASK Bit	Description of Result
15	Link to data comm network information file (NIF) prefix
20	Link to next (if any) data comm NIF prefix
30	MAXPSEUDO stations
32	Data comm audit options
33	Next value of MAXPSEUDO stations

Call

```
B := GETSTATUS (2 & 69 [15:Ø8], Ø, MASK, ARY);
```

Results Returned

A[1].[32:17] is the index to the validity mask in the array ARY. If a validity mask bit is on, then the information for the corresponding MASK bit is present in the array ARY. Each item is located in ARY [ARY [1].[32:17] + bit number + 1].

SUBTYPE 70 Calls (HYPERchannel Adapters)

For SUBTYPE 70 calls, GETSTATUS returns information about a specific HYPERchannel adapter. The call is rejected if the adapter is owned and the caller is not the owner of the adapter. There is no corresponding system command for this call. The call has five subclasses as follows:

SUBCLASS Number	Description
SUBCLASS 1	Returns statistics in formatted form
SUBCLASS 2	Returns statistics in unformatted form
SUBCLASS 3	Returns and clears statistics in formatted form
SUBCLASS 4	Returns and clears statistics in unformatted form
SUBCLASS 5	Returns information in response to a SENSE ADAPTER command

For SUBCLASS calls 1, 2, 3, or 4, the array ARY must be at least 16 words long. For SUBCLASS 5 calls, the array ARY must be at least 174 words long.

Input

Use the following input information for the array ARY:

ARY	Value or Description
ARY [0]	2
ARY [1]	0
ARY [3]	<adapter number>

Call

```
B := GETSTATUS (2 & 70 [15:08], 1, 0, ARY);  % for SUBCLASS 1
B := GETSTATUS (2 & 70 [15:08], 2, 0, ARY);  % for SUBCLASS 2
B := GETSTATUS (2 & 70 [15:08], 3, 0, ARY);  % for SUBCLASS 3
B := GETSTATUS (2 & 70 [15:08], 4, 0, ARY);  % for SUBCLASS 4
B := GETSTATUS (2 & 70 [15:08], 5, 0, ARY);  % for SUBCLASS 5
```

Information Returned for SUBCLASS 1 and 3 Calls

For SUBCLASS 1 or 3, the response in array ARY is returned as follows:

ARY Number	Description
ARY [3]	Count of message propers and data buffers sent on trunk 0, in the range 0 to (2**24)–1, or 16,777,215.
ARY [4]	Count of message propers and data buffers sent on trunk 1, in the range 0 to (2**24)–1, or 16,777,215.
ARY [5]	Count of message propers and data buffers sent on trunk 2, in the range 0 to (2**24)–1, or 16,777,215.
ARY [6]	Count of message propers and data buffers sent on trunk 3, in the range 0 to (2**24)–1, or 16,777,215.
ARY [7]	Count of canceled operations, in the range 0 to (2**12)–1, or 4095.
ARY [8]	Count of terminations due to hardware malfunction or deadman timeout, in the range 0 to (2**12)–1, or 4095.
ARY [9]	Count of retransmissions on trunk 0, in the range 0 to (2**24)–1, or 16,777,215.
ARY [10]	Count of retransmissions on trunk 1, in the range 0 to (2**24)–1, or 16,777,215.
ARY [11]	Count of retransmissions on trunk 2, in the range 0 to (2**24)–1, or 16,777,215.
ARY [12]	Count of retransmissions on trunk 3, in the range 0 to (2**24)–1, or 16,777,215.
ARY [13]	Model number of the adapter from which statistics were taken. This number is a 4-hexadecimal-digit number that is right-justified in the word.
ARY [14]	Revision level of the adapter from which statistics were taken. This number is in the range 0 to 255, inclusive.
ARY [15]	Adapter trunk address of the adapter from which statistics were taken. This number is in the range 0 to 255, inclusive.

Information Returned for SUBCLASS 2 and 4 Calls

For SUBCLASS 2 or 4 calls, the response is placed beginning in word 3 of the array parameter ARY. This response is identical to the result returned by the adapter itself in response to the corresponding operations to the adapter.

A soft error code word is returned in array ARY [1] if the adapter does not exist on the local system or there was an error in reading the statistics.

Information Returned for SUBCLASS 5 Calls

For SUBCLASS 5 calls, array ARY must be at least 174 words long. GETSTATUS places the response beginning in A [3] of the array parameter. This response is identical to the information returned by the adapter in response to a SENSE ADAPTER command. Refer to the appropriate Network Systems Corporation adapter reference manual for the format of the data.

SUBTYPE 71 Calls (Support Libraries)

For SUBTYPE 71 calls, GETSTATUS returns information about the support libraries defined with the system command SL (Support Library). For more information, refer to the *System Commands Reference Manual*.

SUBCLASS 1 Calls

Use SUBCLASS 1 calls to return information for all functions defined in the SL table.

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	2
ARY [1]	0

Call

```
B := GETSTATUS (2 & 71 [15:Ø8], 1, 0, ARY);
```

Results Returned

For SUBCLASS 1 calls, GETSTATUS returns SL information in array ARY as follows:

ARY Number	Value or Description
ARY [0]	2
ARY [1]	0
ARY [2]	SL table information.
ARY [2].[39:20]	The number of entries in the table.

continued

continued

ARY Number	Value or Description
ARY [3] through end	Contains table entries. Each entry is as follows: • ARY [n] = Information about the function: – ARY [n].[47:08] = The length of the entry in words, including this word. – ARY [n].[38:01] is the SYSTEMFILE bit. – ARY [n].[36:01] = 1 if a pending title is present. – ARY [n].[35:01] is the MCPINIT bit. – ARY [n].[34:01] is the TRUSTED bit. – ARY [n].[33:01] is the ONEONLY bit. – ARY [n].[31:04] contains the LINKCLASS. • ARY [n+1] through ARY [n+3] = Function name in substandard form. • ARY [n+4] through ARY [n+ARY [n].[47:08]–1] = The file title (or titles if a pending title is present) in standard form. Each title starts at a word boundary.

SUBCLASS 5 Calls

Use SUBCLASS 5 calls to return information for a designated function.

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	2
ARY [1]	0
ARY [2] through ARY [4]	Function name in substandard form

Call

```
B := GETSTATUS (2 & 71 [15:Ø8], 5, 0, ARY);
```

Results Returned

For SUBCLASS 5 calls, GETSTATUS returns soft error 216 if the named function is not defined; otherwise, GETSTATUS returns the SL information in array ARY as follows:

ARY Number	Value or Description
ARY [0]	2
ARY [1]	Contains information about the function: • ARY [1].[47:08] = The length of the entry in words, including this word. • ARY [1].[38:01] is the SYSTEMFILE bit. • ARY [1].[36:01] = 1 if a pending title is present. • ARY [1].[35:01] is the MCPINIT bit. • ARY [1].[34:01] is the TRUSTED bit. • ARY [1].[33:01] is the is the ONEONLY bit. • ARY [1].[31:04] contains the LINKCLASS.
ARY [2] through ARY [4]	Function name in substandard form.
ARY [5] through end	File title (or titles if a pending title is present) in standard form. Each title starts at a word boundary.

See the example of a SUBTYPE 71 GETSTATUS call at the end of this section.

SUBTYPE 76 Calls (Logging)

A SUBTYPE 76 call corresponds to the system command LOGGING. SUBCLASS 1 retrieves all LOGGING specifications. SUBCLASS 2 retrieves a specific LOGGING <major> and <minor> type.

Input for SUBCLASS 1 Calls

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	2

Call

```
B := GETSTATUS (2 & 76 [15:08], 1, 0, ARY);  % for SUBCLASS 1
```

Input for SUBCLASS 2 Calls

Use the following input information for array ARY:

ARY	Value or Description
ARY [0]	2
ARY [1]	0
ARY [2]	MAJOR type

continued

continued

ARY	Value or Description
ARY [3]	MINOR type

Call

```
B := GETSTATUS (2 & 76 [15:Ø8], 2, Ø, ARY);  % for SUBCLASS 2
```

Results Returned

ARY[0] contains a pointer to the log-setting stream. ARY [ARY [0]] through the end contains the log-setting stream. The log-setting stream is a sequence of 16-bit items.

For SUBCLASS 1, the log-setting stream is returned in the following format:

Number of Items	Size of Each	Description
1	16-bit	Number of MAJOR types defined. Number = M. The specification "LOGGING *" is an exception if M is greater than or equal to 4"FF00".
M	16-bit	A MAJOR-type specification. The value for the MAJOR type is S. If S > = 4"FF00", this log setting is a MAJOR-type. S – 4"FF00" is the log setting for all MINOR types that are associated with the MAJOR type. If S < 4"FF00", this item is a pointer to a MINOR-type specification at POINTER (A [A[0]]) + S.

The MINOR-type specification is in the format that follows.

Number of Items	Size of Each	Description
1	16-bit	Number of MINOR types defined. Number = m.
m	8-bit	A MINOR-type log setting whose value = s; s is the setting for the combination M,m.

For SUBCLASS 2, word 1 of array ARY contains the log setting for MAJOR type followed by MINOR type.

For both cases (SUBCLASS 1 or SUBCLASS 2), the log setting is an 8-bit item that reflects the logging for the applicable major type or combination of MAJOR type and MINOR type. The lower 4 bits reflect logging to the SUMLOG, while the upper 4 bits reflect logging to the job file. Every log record has a result indicator to indicate the type of record. All 8-bit log setting values are independent for each applicable MAJOR type or combination of MAJOR type and MINOR type.

The bit combinations and their results are as follows:

SUMLOG	Job File	Log Record Result Indicator
Bit 0	Bit 4	If turned on, success records are being logged.
Bit 1	Bit 5	If turned on, failure records are being logged.
Bit 2	Bit 6	If turned on, security-relevant records are being logged.
Bit 3	Bit 7	If turned on, security violation records are being logged.

SUBTYPE 78 Calls (SUPPRESSWARNING)

This call retrieves a copy of the system warning array or part of the array in which the system stores the list of warnings to be suppressed. The information stored in the suppress warning array is either zero, or one or more 2-byte entries terminated by two bytes of all zeros.

Each 2-byte entry is either a warning number to be suppressed or the first or last number in a range of warnings to be suppressed. The last number in a range of warnings has bit [15:01] on. To tell whether an individual 2-byte entry is the first number of a range of warnings to be suppressed or a single warning to be suppressed, check bit [15:01] of the next 2-byte entry. If it is on, then the two 2-byte entries represent a range of warnings to be suppressed. Otherwise, the first 2-byte entry represents a single warning to be suppressed.

Since it is normally small, you can usually retrieve a copy of either the entire warning array or any portion of it. In either case, use array ARY[1] (word 1) to indicate the starting index of the information that you want to retrieve from the system warning suppression array. To retrieve the entire array, set ARY[1] equal to zero. GETSTATUS then copies as many words from the system warning array as will fit in the array parameter array, starting at the word in the system warning array that you specified.

If GETSTATUS retrieves the entire warning array, or the remainder of the array, it stores a zero in array ARY[0]. Otherwise, it stores the index of the next word in the system warning array in array ARY[0]. In this case, to retrieve the rest of the array, put that index value into ARY[1] and repeat the GETSTATUS call.

TYPE Parameter

The TYPE parameter has the following values:

Field	Value
[07:08]	2 (TYPE)
[15:08]	78 (SUBTYPE)

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY[0]	2
ARY[1]	0 or starting index in system warning array

Call

```
B := GETSTATUS (2 & 78 [15:08], 0, 0, ARY);
```

Example

This example displays the numbers of suppressed warnings. In this example, the array parameter has been made large enough to hold the largest possible copy of the system warning array.

```
BEGIN
 BOOLEAN RSLT;       % ERROR INDICATOR FROM GETSTATUS
 ARRAY ARY [0:7286];   % ARRAY PARAMETER FOR GETSTATUS
   POINTER PARY;        % POINTER INTO "ARY"
 ARRAY D [0:2];  POINTER PD;

     ARY [0] := 2;
     ARY [1] := 0;
     IF RSLT := GETSTATUS (2 & 78 [15:8], 0, 0, ARY) THEN
       PROGRAMDUMP (ARRAYS)     % ERROR
     ELSE
       BEGIN
         PARY := POINTER (ARY [3]);
         IF PARY = 48"0000" THEN     % -NONE-
           DISPLAY ("NONE")
         ELSE
           DO       % LOOP OVER THESE WARNING NUMBERS
             BEGIN
               REPLACE POINTER (D) BY " " FOR 18;
               REPLACE PD:POINTER (D) BY REAL (PARY, 2) FOR * DIGITS;
               PARY := *+2;
               IF BOOLEAN (REAL (PARY, 2)).[15:1] THEN
                 BEGIN     % IT'S ARY RANGE OF WARNING NUMBERS
                   REPLACE PD BY " TO ",
                          (REAL (PARY, 2).[14:15]) FOR * DIGITS;
                   PARY := * + 2;    % SKIP EXTRA 2 BYTES
                 END;
               DISPLAY (POINTER (D));

             END
            UNTIL REAL (PARY, 2) = 0;
       END;
END.
```

SUBTYPE 80 Calls (Time Zone Name)

Use this SUBTYPE to return the name and information about the time zone specified at the installation. Refer to the TR (Time Reset) system command for an explanation of how to set the date, the time, and the time zone.

TYPE Parameter

The TYPE parameter has the following values:

Field	Value
[7:08]	2 (TYPE)
[15:08]	80 (SUBTYPE)
[37:01]	0 OR 1 (MLS indicator). Indicates whether the time zone name should be returned in a particular language. If the MLS indicator is 0, the time zone name is returned in the language of the calling task. If the MLS indicator is 1, place the name of the desired language in substandard format starting in word A [SIZE (A) – 6].

Input

Use the following input information for array ARY:

ARY	Value or Description
A[0]	12

Call

```
B := GETSTATUS (Ø & 2 [7:Ø8] & (8Ø) [15:Ø8]
                & (MLS indicator) [37:Ø1], Ø, Ø, A);
```

Examples

The following example shows the code to use to retrieve the time zone name in Spanish:

```
A[Ø] := 12;
REPLACE A [SIZE (A) -6 ] BY 48"Ø7", "ESPANOL";
B := GETSTATUS (Ø & 2 [7:Ø8] & (8Ø) [15:Ø8]
                & 1 [37:Ø1], Ø, Ø, A);
```

The following example shows the code to use to retrieve the time zone name in the language of the task (or the default system language):

```
A[Ø] := 12;
B := GETSTATUS (Ø & 2 [7:Ø8] & (8Ø) [15:Ø8]
                & Ø [37:Ø1], Ø, Ø, A);
```

Results Returned

If the system does not have a time zone specified, then GETSTATUS stores zeros in words 1 through 12 of the array parameter. Otherwise, GETSTATUS returns the following information for SUBTYPE 80 calls:

ARY	Value or Description
A [1]	Predefined time zone number or zero. (Refer to the TR Call (Extended Version) subsection of Section 9, SETSTATUS Request Type 2 Miscellaneous Requests) in this manual for a list of predefined time zones.)
A [2]	Offset direction and time zone validity indication: • 0 = No time zone configured. • 1 = Add offset to universal time (UT) to produce local time. • 2 = Subtract offset from UT to get local time:
A [3]	Hours offset (0 through 24).
A [4]	Minutes offset (0 through 59).
A [5] through A [12]	Time zone name in substandard form followed by the time zone abbreviation in substandard form (as designated for custom names; or translated into requested language, task language, or system language for predefined time zones).

SUBTYPE 85 Calls (SYSTEMACCOUNTING)

Use this SUBTYPE to return information about attributes for dependent task accounting and file accounting. On InfoGuard systems, only security administrators can use this option.

Input

Use the following input information for array ARY:

ARY	Value or Description
ARY [1]	2

Call

```
B := GETSTATUS (2 & 85 [15:Ø8], Ø, Ø, ARY);
```

Results Returned

GETSTATUS returns the following values for array ARY:

ARY	Value or Description
ARY [1]	This field contains one of the following values: • 0 = UNSPECIFIED dependent task accounting • 1 = ANONYMOUS dependent task accounting • 2 = IDENTIFIED dependent task accounting
ARY [2]	This field contains one of the following values: • 0 = UNSPECIFIED file accounting • 1 = ANONYMOUS file accounting • 2 = IDENTIFIED file accounting

Examples of Miscellaneous SUBTYPE 1, 2, 12, and 71 GETSTATUS Calls

```
BEGIN
DEFINE
    XSLINKF = [32:17] #,
    BASE = ARY[1].XSLINKF #,
    GSITEM (ARY,BITNO) = ARY [BASE+BITNO+1] #,
    GSMASK (ARY,BITNO) = BOOLEAN (ARY[BASE].[BITNO:1]) #,
    GSNAME (ARY,BITNO) = (BASE + GSITEM (ARY,BITNO).[32:17]) #;
ARRAY ARY[Ø:255], BUF[Ø:22], SCRATCH[Ø:22];
POINTER P,Q;
REAL TEMP,COUNT,J,Z;
BOOLEAN GSRSLT;
FILE LINE(KIND=PRINTER);

PROCEDURE MASKTOLIST(MSK1,MSK2,TOPBITNO,P);
  VALUE MSK1,MSK2,TOPBITNO;
  REAL MSK1,MSK2,TOPBITNO;
  POINTER P;
BEGIN
    REAL L,K,N;
    BOOLEAN ONOFF;
    LABEL START;
    START:
    FOR L := Ø STEP 1 UNTIL TOPBITNO DO
        BEGIN
        IF NOT(BOOLEAN(MSK1).[L:1] EQV ONOFF) THEN
            BEGIN
            IF (ONOFF := NOT ONOFF) THEN
                REPLACE P:P BY (N := L+K) FOR * DIGITS
            ELSE
                BEGIN
                IF N ISNT (N := L+K-1) THEN
                    REPLACE P:P BY "-", N FOR * DIGITS;
                REPLACE P:P BY ",";
                END;
            END;
        END;
    IF MSK2 ISNT Ø THEN
        BEGIN
        MSK1 := READLOCK(Ø,MSK2);
        K := TOPBITNO+1;
        GO TO START;
        END;
    IF ONOFF AND BOOLEAN(MSK1.[TOPBITNO-1:1]) THEN
        REPLACE P:P BY "-", (N := TOPBITNO+K) FOR * DIGITS;
    IF P-1 = "," THEN
        REPLACE P-1 BY " ";
END MASKTOLIST;
```

```
%------------------------ RETURN MIXCOUNT ---------------------------

  ARY[Ø] := 2;
  ARY[1] := Ø;
  GSRSLT := GETSTATUS(Ø & 2 [7:8] & 1 [15:8],Ø,Ø & 1 [3:1],ARY);
  TEMP := GSITEM(ARY,3);
  REPLACE BUF BY " " FOR 132;
  IF TEMP = Ø THEN
       REPLACE BUF BY "THE MIX IS EMPTY"
  ELSE
  IF TEMP = 1 THEN
       REPLACE BUF BY "THERE IS ONE JOB IN THE MIX"
  ELSE
       REPLACE BUF BY "THERE ARE ",TEMP FOR * DIGITS,
                      " JOBS & PROGRAMS IN THE MIX";
  WRITE(LINE,14,BUF);

%-----------------------CU SYSTEM COMMAND----------------------------

  ARY[Ø] := 2;
  ARY[1] := Ø;
  GSRSLT := GETSTATUS(Ø & 2 [7:8] & 2 [15:8],Ø,Ø & 1 [8:1],ARY);
  TEMP := GSNAME(ARY,8);
  COUNT := ARY[TEMP]+ARY[TEMP+1]+ARY[TEMP+2];
  REPLACE BUF BY " " FOR 132;
  REPLACE BUF BY "AVAIL=", ARY[TEMP] FOR * DIGITS,
                 " OLAY=", ARY[TEMP+1] FOR * DIGITS,
                 " SAVE=", ARY[TEMP+2] FOR * DIGITS,
                 " * TOTAL=", COUNT FOR * DIGITS;
  WRITE(LINE,14,BUF);

%-----------------------ID SYSTEM COMMAND----------------------------

  ARY[Ø] := 2;
  ARY[1] := Ø;
  GSRSLT := GETSTATUS(Ø & 2 [7:8] & 1 [15:8],Ø,Ø & 1 [15:1],ARY);
  REPLACE BUF BY " " FOR 132;
  REPLACE P:POINTER(BUF) BY "DC PREFIX : ";
  IF GSMASK(ARY,15) THEN
      STANDARDTODISPLAY (POINTER( ARY [GSNAME (ARY, 15)]),P)
  ELSE
      REPLACE P:P BY "NOT SPECIFIED ";
  WRITE(LINE,14,BUF);
```

```
%-----------------------SC INFORMATION-------------------------------

  ARY[Ø] := 2;
  ARY[1] := Ø;
  GSRSLT := GETSTATUS(Ø & 2 [7:8] & 2 [15:8],Ø,
            Ø & 1 [1:1] & 1 [2:1] & 1 [11:1], ARY);
  REPLACE BUF BY " " FOR 132;
  REPLACE BUF BY "=SYSTEM CONFIGURATION=";
  WRITE(LINE,14,BUF);

  REPLACE BUF BY " " FOR 132;
  TEMP := GSITEM(ARY,1);
  REPLACE P:POINTER(BUF)+5 BY ONES(TEMP) FOR * DIGITS," PROCESSOR";
  IF ONES(TEMP) > 1 THEN
      REPLACE P:P BY "S";
  REPLACE P:P BY " => ";
  MASKTOLIST(TEMP,Ø,19,P);
  WRITE(LINE,14,BUF);

  REPLACE BUF BY " " FOR 132;
  TEMP := GSITEM(ARY,2);
  REPLACE P:POINTER(BUF)+5 BY ONES(TEMP) FOR * DIGITS,
      " I/O PROCESSOR";
  IF ONES(TEMP) > 1 THEN
      REPLACE P:P BY "S";
  REPLACE P:P BY " => ";
  MASKTOLIST(TEMP,Ø,19,P);
  WRITE(LINE,14,BUF);

  REPLACE BUF BY " " FOR 132;
  SCRATCH[Ø] := GSITEM(ARY,11).[19:48];
  REPLACE P:POINTER(BUF)+5 BY "DO = ", POINTER(SCRATCH,4)
      FOR 5 WITH HEXTOEBCDIC;
  WRITE(LINE,14,BUF);
  REPLACE BUF BY " " FOR 132;

%----------------------SF SYSTEM COMMAND-----------------------------

  ARY[Ø] := 2;
  ARY[1] := Ø;
  GSRSLT := GETSTATUS(Ø & 2 [7:8] & 1 [15:8],Ø,
              Ø & 1 [7:1] & 1 [3Ø:1],ARY);
  Z := GSITEM(ARY,7);
  REPLACE BUF BY " " FOR 132;
  J := GSITEM(ARY,3Ø);
  REPLACE BUF BY "FACTORS: OLAY GOAL=", Z.[31:16] FOR * DIGITS,
                 "; AVAILMIN=", Z.[47:16] FOR * DIGITS,
                 "; FACTOR=", Z.[15:16] FOR * DIGITS,
                 "; MEM PRIORITY=", J FOR * DIGITS;
  WRITE(LINE,14,BUF);
```

```
%-----------------------OP SYSTEM COMMAND----------------------------

  ARY[Ø] := 2;
  ARY[1] := Ø;
  GSRSLT := GETSTATUS(Ø & 2 [7:8] & 1 [15:8],Ø,Ø & 1 [Ø:1],ARY);
  TEMP := GSITEM(ARY,Ø);
  REPLACE BUF BY " " FOR 132;
  REPLACE P:POINTER(BUF) BY "SYSTEM OPTIONS SET: ";
  MASKTOLIST(TEMP,Ø,47,P);
  WRITE(LINE,14,BUF);

%-----------------------SI SYSTEM COMMAND----------------------------

  ARY[Ø] := 2;
  ARY[1] := Ø;
  GSRSLT := GETSTATUS(Ø & 2 [7:8] & 1 [15:8],Ø,Ø & 1 [1Ø:1],ARY);
  REPLACE BUF BY " " FOR 132;
  REPLACE P:POINTER(BUF) BY "INTRINSICS : ";
  IF GSMASK(ARY,1Ø) THEN
      STANDARDTODISPLAY (POINTER (ARY [GSNAME (ARY, 1Ø)]), P)
  ELSE
      REPLACE P:P BY "NOT SPECIFIED.";
  WRITE(LINE,14,BUF);

%-----------------------WM SYSTEM COMMAND----------------------------

  ARY[Ø] := 2;
  ARY[1] := Ø;
  GSRSLT := GETSTATUS(Ø & 2 [7:8] & 1 [15:8],Ø,
            Ø & 1 [16:1] & 1 [17:1] & 1 [19:1] & 1 [21:1]
              & 1 [22:1] & 1 [25:1] & 1 [26:1],ARY);
  REPLACE BUF BY " " FOR 132;
  IF GSMASK(ARY,16) THEN % VALID MCP NAME
      BEGIN
      REPLACE P:POINTER(BUF) BY "MCP :";
      STANDARDTODISPLAY (POINTER (ARY [GSNAME (ARY, 16)]), P);
      TEMP := GSITEM(ARY,19);
      REPLACE P:P BY " ", TEMP.[47:16] FOR * DIGITS, ".",
          TEMP.[31:16] FOR * DIGITS, ".",
          TEMP.[15:16] FOR * DIGITS;
      WRITE(LINE,14,BUF);
      REPLACE BUF BY " " FOR 132;
      END;
```

```
IF GSMASK(ARY,17) THEN % COMPILE TIME OPTIONS PRESENT
    BEGIN
      REAL COUNT;
    REPLACE P:POINTER(BUF) BY "COMPILETIME OPTIONS:";
    Q := POINTER (ARY [GSNAME (ARY, 17)]);
    WHILE REAL (Q, 1) ISNT Ø DO
        BEGIN
        REPLACE P:P BY Q:Q UNTIL = 48"ØØ", ",";
        Q := Q+1;
        IF COUNT := *+1 GEQ 3 THEN
          BEGIN
            WRITE (LINE, 14, BUF);
            REPLACE BUF BY " " FOR 132;
            COUNT := Ø;
            P := POINTER (BUF) + 21;
          END;
        END;
    IF COUNT GTR Ø THEN
      WRITE(LINE,14,BUF);
    REPLACE BUF BY " " FOR 132;
    END;

P := POINTER(BUF);
IF GSMASK(ARY,21) THEN % HALT/LOAD REASON
    BEGIN
    REPLACE P:P BY "H/L REASON: ", POINTER(ARY[GSNAME(ARY,21)])
        FOR GSITEM(ARY,21).[15:16], "; ";
    END;
IF GSMASK(ARY,22) THEN %  HALT/LOAD UNIT
    BEGIN
    REPLACE P:P BY "H/L UNIT: ", GSITEM(ARY,22) FOR * DIGITS, "; ";
    END;
WRITE(LINE,14,BUF);

REPLACE P:= POINTER (BUF) BY " " FOR 132;
IF GSMASK(ARY,26) THEN % SYSTEM SERIAL NUMBER
    BEGIN
    REPLACE P:P BY "SYSTEM SERIAL NUMBER: ",
        GSITEM(ARY,26) FOR * DIGITS, " ; ";
    END;
IF GSMASK(ARY,25) THEN % CATALOG LEVEL
    BEGIN
    REPLACE P:P BY "CATALOG LEVEL: ",
        GSITEM(ARY,25) FOR * DIGITS;
    END;
WRITE(LINE,14,BUF);
```

```
%-----------------------CS SYSTEM COMMAND----------------------------

  ARY[Ø] := 2;
  ARY[1] := Ø;
  GSRSLT := GETSTATUS(Ø & 2 [7:8] & 1 [15:8],Ø,Ø & 1 [8:1],ARY);
  REPLACE BUF BY " " FOR 132;
  REPLACE P:POINTER(BUF) BY "SUPERVISOR : ";
  IF GSMASK(ARY,8) THEN
      STANDARDTODISPLAY (POINTER (ARY [GSNAME (ARY, 8)]), P)
  ELSE
      REPLACE P:P BY "NOT SPECIFIED.";
  WRITE(LINE,14,BUF);

%-----------------------SL SYSTEM COMMAND----------------------------

  ARY[Ø]:= 2;
  ARY[1]:= Ø;
  REPLACE POINTER (ARY[2]) BY 48"ØC" 8"USERFUNCTION";
  GSRSLT := GETSTATUS (Ø & 2  [7:8] & 71 [15:8],     % GET FUNCTION
                       5, Ø, ARY );
  IF NOT GSRSLT THEN
      BEGIN
      REPLACE P:POINTER(BUF) BY "SL USERFUNCTION=";
      STANDARDTODISPLAY (POINTER (ARY[5]), P);
      IF ARY[1] ISNT Ø THEN
          BEGIN
          REPLACE P:POINTER(BUF) BY "; ";
          STANDARDTODISPLAY (POINTER(ARY[5+((ARY[5].[47:8]+5) DIV 6)]),
                             P);
          REPLACE P:POINTER(BUF) BY " PENDING";
          END
      END
  ELSE
      REPLACE P:POINTER(BUF) BY "FUNCTION USERFUNCTION IS NOT DEFINED";
  WRITE(LINE,14,BUF);
```

```
%------------------------ MCP LANGUAGES -----------------------------

  REPLACE BUF BY " " FOR 132;
  ARY[Ø] := 2;
  ARY[1] := Ø;
  GSRSLT := GETSTATUS( Ø & 2 [7:8] & 12 [15:8],
                       Ø, Ø & 1 [11:1], ARY);
  IF NOT GSRSLT THEN
   IF GSMASK (ARY, 11) THEN
    BEGIN
       REAL LANGCNT, AI;
      WRITE (LINE, <"MCP MLS LANGUAGES">);
      AI := GSNAME (ARY, 11);
      LANGCNT := ARY[ AI ];
      AI := * + 1;
      FOR J:= Ø STEP 1 UNTIL LANGCNT-1 DO
          BEGIN
            REPLACE POINTER (BUF) BY "   ",
                POINTER( ARY[ AI + J*3 ], 8 ) + 1
                          FOR ARY[ AI + J*3 ].[47:Ø8];
            WRITE(LINE,14,BUF);
            REPLACE BUF BY " " FOR 132;
          END;
    END;
END.
```

Output from Examples

```
THERE ARE 1Ø6 JOBS & PROGRAMS IN THE MIX
AVAIL=27544448 OLAY=3450875 SAVE=2183285 * TOTAL=8388608
DC PREFIX : MPA15/C
=SYSTEM CONFIGURATION=
     1 PROCESSOR => 5
     1 I/O PROCESSOR => Ø
     DO = Ø21ØØ
FACTORS:  OLAY GOAL=Ø; AVAILMIN=Ø; FACTOR=1ØØ; MEM PRIORITY=Ø
SYSTEM OPTIONS SET: 2,4-4,8,12,14,16,19-21,24-27,4Ø
INTRINSICS : SYSTEM/INTRINSICS
MCP :SYSTEM/ASD/MCP/38Ø42/DIAG  3.8.42
COMPILETIME OPTIONS:TRACE,DIAGNOSTICS,EXPERIMENTAL,
                    LINEINFO,LOCKTRACE,READLOCK,
                    READLOCKTIMEOUT,RESTART,FREEZEOUTTC,
                    ASD,
H/L REASON: HARDWARE    ; H/L UNIT: 75;
SYSTEM SERIAL NUMBER: 8 ; CATALOG LEVEL: Ø
SUPERVISOR : SUPPRESS/MIX/NOS
FUNCTION USERFUNCTION IS NOT DEFINED
MCP MLS LANGUAGES
   ENGLISH
```

Section 4
GETSTATUS Request Type 3 (Directory Calls)

GETSTATUS directory calls can be used to retrieve information about disk directories, catalogs, disk families, disk files, and cataloged tape files.

The general GETSTATUS call is as follows:

```
B := GETSTATUS (TYPE, SUBCLASS, MASK, ARY);
```

The following information describes parameter definitions for TYPE. The parameters SUBTYPE, MASK, and ARY are described in detail later in this section.

TYPE

This parameter generally selects the specific case and subcase within the GETSTATUS intrinsic. For directory requests, the fields within this word are as follows:

TYPE.TYPEF [07:08] = 3

SUBTYPEF = [15:08]

The field SUBTYPEF designates the SUBTYPE of the directory search These SUBTYPEs are as follows:

SUBTYPE	Description
0	This request searches for one or more specified files or directory. SUBTYPE 0 corresponds to the system command *PD <file name> ON <family>*. PD stands for Print Directory.
1	This initial request searches for all files under a specified directory name. SUBTYPE 1 corresponds to the system command *PD <directory name/=> ON <family>*.
2	This request is referred to as a *continuation request* and can be used after a SUBTYPE 1 or a SUBTYPE 2 call that has returned an indication that more files exist under the initial directory of the designated file name.
3	Reserved.
4	This SUBTYPE is also a continuation request. You can use it after a SUBTYPE 1 or SUBTYPE 4 call has returned an indication that more files exist under the directory.
5	This subtype requests that a copy of the volume library be made.

continued

continued

SUBTYPE	Description
6	This request returns the total amount of available space on a family or a family member.
7	This SUBTYPE requests that a complete family directory be copied to a designated file.
8	This request is no longer available.
9	This request copies all volume directory data records. You can use this request only if you are a privileged user.
10	This request copies only volume directory data records whose FAMILYOWNER matches the usercode of the process.
11	This request reads an archive directory record for a given file on a given disk family.
12	This request copies all archive data records from the archive directory for a given family to a new disk file.
13	This request copies archive records for files under a designated usercode or an asterisk (*) usercode from the archive directory for a given family to a new disk file.
14	This request returns information about disk files that are currently open or in use by the system or a program. SUBTYPE 14 corresponds to the system command SHOWOPEN.

Other bits are used in the TYPE parameter for SUBTYPEs 0, 1, 2, and 4. They are as follows:

RETURNRESIDENTF = [37:01]

If this bit is turned off for SUBTYPES 1, 2, and 4, GETSTATUS reports only on resident disk files and on directory names that have a level equal to the MAXLEVEL designated in the SUBCLASS parameter. Otherwise, GETSTATUS reports on resident disk files, nonresident archived disk files, nonresident cataloged disk files, cataloged tape files, and directories.

ONLYSYSTEMFILESF = [38:01]

This bit limits the GETSTATUS search to system files (the directory for nonusercoded files) and is applicable only to SUBTYPEs 1, 2, and 4.

DISPLAYFORMNAMEF = [39:01]

If this bit is turned off, the file and directory names passed to GETSTATUS and returned by GETSTATUS are in standard form. For information about standard form, refer to the DISPLAYTOSTANDARD function in the *DCALGOL Reference Manual*.

If the DISPLAYFORMNAMEF bit is turned on, it indicates that the requested and returned <file names> are in display form format for SUBTYPEs 1, 2, and 4. If this bit

is on, bit 40 (RETURNFULLNAMEF) must also be on. If RETURNFULLNAMEF is off, GETSTATUS returns hard error 51. Refer to Appendix A in this manual for a listing of hard error messages.

Each display form name returned is in the following format:

```
<two byte-length characters> <display form name>
```

The two byte-length characters contain the length of the display form name in binary form. The normal decimal point is not appended as a delimiter to the display form name variable.

RETURNFULLNAMEF = [40:01]

This bit instructs GETSTATUS to return the full file name instead of the simple form name list.

If RETURNFULLNAMEF is turned off for SUBTYPEs 1, 2, and 4, GETSTATUS returns a file status word for each file and directory encountered. For example, if files A/B/C/X and A/B/C/Y exist and a SUBTYPE 1 call for A/B is issued, A, B, C, X, and Y are reported.

If RETURNFULLNAMEF is turned on for SUBTYPEs 1, 2, and 4, GETSTATUS reports only on full file names and directories limited by the SUBCLASS.MAXLEVELF parameter. For example, if A/B/C/X and A/B/C/Y exist and a SUBTYPE 1 call for A/B is issued, files A/B/C/X and A/B/C/Y are reported.

USERCODEONLYF = [41:01]

This bit inhibits the secondary search of the system directory (the directory for nonusercoded files). It does so if the specified file or directory is not found under the process usercode directory and the original security byte was 1 (it had no explicit asterisk or usercode). USERCODEONLYF has effect only if the process is running under a usercode or if the file name or directory name is specified with an asterisk (*) or an explicit usercode.

RETAINUSERCODEF = [42:01]

If RETAINUSERCODEF is turned on, GETSTATUS reports the true file name of the files located. If RETAINUSERCODEF is off, the process is running under a usercode, and the caller designated a file or directory name without a usercode or an asterisk (*), GETSTATUS returns the file name without the process usercode directory name under which the file was found.

WAITFORFILEF = [43:01]

This bit instructs GETSTATUS to wait for the designated family <on part> to be mounted if the family is not present at the time of the GETSTATUS call.

GSLINKINONPARTF = [45:01]

If this bit is turned on, GETSTATUS appends the family name to the file name reported when TYPE.RETURNFULLNAMEF is turned on. The family name will be the actual family name that was used after applying family substitution.

SUBCLASS Parameter for SUBTYPE 0, 1, 2, and 4 Calls

The following definitions pertain to the SUBCLASS parameter for SUBTYPE 0, 1, 2, and 4 calls:

MAXCATLEVELF = [47:08]

This field is applicable only if MASK bit 16 is turned on. This field allows the caller to limit the number of file generations returned from the catalog system. For details, refer to Table 4–2. The value 0 in this field indicates that all file entries are to be returned. Otherwise, the number is designated with this field.

CDROMF = [39:01]

When this field is equal to 1, the family name found within the ARY GETSTATUS parameter refers to a CD-ROM instead of a disk pack.

ORGLEVELF = [38:19]

For a SUBTYPE 2 continuation request, this field indicates the number of levels in the continuation name that were in the original SUBTYPE 1 request. The number of levels can be retrieved from ARY[1].LEVELF after a TYPE.SUBTYPEF = 1, 2, or 4 call.

MAXLEVELF = [19:20]

For SUBTYPEs 1, 2, and 4, when TYPE.MAXLEVELF is not equal to 0, the level of searching is limited to the value in TYPE.MAXLEVELF names above the number of levels of the original directory name. For example, if the original directory name was A/B and MAXLEVELF equals 1, GETSTATUS restricts its search to all names under the initial directory of A/B plus one more level; that is, if the files A/B/C/D and A/B/E are present, GETSTATUS returns information about the directory A/B/C and the file A/B/E, but not about the file A/B/C/D. This call corresponds to the system command *PD A/B/=1*.

MASK Parameter for SUBTYPE 0, 1, 2, and 4 Calls

Each bit in the MASK word for SUBTYPE 0, 1, 2, and 4 calls indicates a specific piece of data that is to be returned about the designated files (refer to Table 4–1). If MASK is 0 or 1, GETSTATUS returns a one-word summary of each file or directory located. Otherwise, for each file located, GETSTATUS also returns the attributes that correspond to the MASK bits selected.

Note: All available attributes can easily be retrieved by turning on all the bits in MASK. However, some of the information, such as the image of the disk file header (bit 8) or the row address words (bit 33), might consume a large number of words in ARY. If information for several files is to be retrieved with a single GETSTATUS call, it is more efficient to use MASK bits to select only the information that is needed.

Array Parameter for SUBTYPE 0, 1, 2, and 4 Calls

ARY is used to pass parameter information to GETSTATUS. GETSTATUS then stores result information in the array ARY. The input data in ARY is used by GETSTATUS to determine the files and directories to retrieve. The structure of the input values for ARY is explained in the discussion for the specific SUBTYPES:

- Searching for a specific file
- Searching for all files under a given directory
- Continuing a directory search

  The array ARY can be segmented. In addition to the general limits on the values that GETSTATUS accepts in ARY [0].[19:20] and ARY [0].[39:20], directory requests have other limits (refer to "ARY Limits" in Section 1). For all directory GETSTATUS calls, SIZE (ARY) must be less than 21846, and ARY [0].[19:20] must be less than 2048. Otherwise, hard error 40, Boolean (1 & 40 [11:08]), is returned. Refer to Appendix A for a listing of hard errors.

  The following definitions pertain to the format of various words in the array ARY.

  - ERRORF = [47:01]
  - ERRORVALUEF = [46:08]
  - ADDLINFOF = [46:08]
  - VALUEF = [38:06] This field is divided into three subfields:
    - SUBVALUE2F = [38:02]
    - SUBVALUE3F = [36:01]
    - SUBVALUE1F = [35:03]
  - LINKF = [32:17]

    In pointer or link words, this field usually contains a character or word index.
  - INFOF = [15:16]

    In pointer or link words, this field usually contains the length of the referenced item in characters or words.
  - NEXTLEVELLINKF = [15:11]
  - RESIDENTSTATEF = [04:01]
  - LEVELF = [03:04]

Family Name Information for SUBTYPE 0, 1, 2, and 4 Calls

The family name on which the directory or files are to be found should be included in the display form or standard form names supplied in the array ARY. If the caller does not supply the family name, GETSTATUS uses the family name DISK. If the process has an active family substitution and the family name to be searched matches the family substitution target name, GETSTATUS applies family substitution. If the requested file or directory cannot be located after substituting the primary name, GETSTATUS searches the secondary substitute (if present). To search for cataloged tape files, use the family name TAPE.

A potential four-way search can be performed under the following circumstances:

- If the process is running under a usercode, the file name to be located does not specify a usercode or an asterisk (*).
- If both TYPE.ONLYSYSTEMFILESF and TYPE.USERCODEONLYF are turned off.
- If GETSTATUS invokes family substitution with an OTHERWISE clause.

The order of searching used by GETSTATUS is not the normal order that would be used. GETSTATUS tries four searches in the following order:

1. Under the task usercode on the primary family
2. Under the task usercode on the secondary family
3. Under the asterisk (*) directory on the primary family
4. Under the asterisk (*) directory on the secondary family

The previous sequence differs from normal system searches because steps 2 and 3 are reversed. The first search that succeeds terminates the search.

General Format of Results from SUBTYPE 0, 1, 2, and 4 Directory Calls

GETSTATUS reports on the status of a file with one of three structured words. In this section, the three kinds of words used are called *soft error words, file status words,* and *fixed info link words*. When the information to report for a file cannot fit in one word, GETSTATUS uses a fixed info link word to point to the extra information.

Soft Error Word

If bit 47 of the word is turned on, an error caused GETSTATUS to terminate the report for the file. Soft error words are structured as follows:

- ERRORF = [47:01]

  This bit is turned on; it distinguishes soft error words from file status words and fixed info link words.

- ERRORVALUEF = [46:08]

  The ERROFVALUEF field contains the soft error number. Refer to Appendix B for a listing of soft errors.

File Status Word

If bit 47 of the word is turned off and the field [3:04] contains a value greater than 0, the word is a file status word. A file status word indicates the general status of the file, directory, or both and contains a link to the file or directory name.

File status words are structured as follows:

- [47:01] = 0
- ADDLINFOF = [46:08]

  For resident disk files, this field contains the FILEKIND.
- VALUEF = [38:06]

  VALUEF is divided into three subfields:

  - SUBVALUE2F = [38:02]

    The value in this subfield indicates the kind of entry represented by this word, as follows:

Value	Description
0	Reserved.
1	This entry is a file.
2	This entry is a directory.
3	This entry is a file and a directory.

  - SUBVALUE3F = [36:01]

    The number 1 in this subfield indicates that the file is in use.

  - SUBVALUE1F = [35:03]

    If TYPE.RETURNFULLNAMEF is turned on, this field contains a copy of the lower three bits of the qualification byte of the standard file name.

    If TYPE.RETURNFULLNAMEF is turned off for SUBTYPE 0, this field returns a 2 if the file is from the system directory or a 3 if the file is from a usercode directory.

    Otherwise, this field is 0 when the LEVELF field does not contain the value 1. If the LEVELF field does contain the value 1, the field contains the following values:

Value	Description
0	Reserved.
1	The file is from the usercode directory of the process, and TYPE.RETAINUSERCODEF is turned off.
2	The file is from the system directory.

continued

continued

Value	Description
3	The first name of the file is a usercode.

- LINKF = [32:17]

  This field is an absolute character index into the array ARY and points to the file name or subpart thereof for SUBTYPE 14. If TYPE.RETURNFULLNAMEF is on, GETSTATUS returns the full file name. If the call is a SHOWOPEN call (SUBTYPE 14), GETSTATUS also returns the full file name.

- NEXTLEVELLINKF = [15:11]

  This field links together the subnames of the last file name so that the last name returned can be recreated. These links are generated only if TYPE.RETURNFULLNAMEF is OFF and SUBTYPEF equals 1, 2, or 4. If a subsequent SUBTYPE 4 call is made, the information in these links is used by GETSTATUS to determine the continuation point.

  RESIDENTSTATEF = [04:01]

  This bit is turned on if the file is resident and the TYPE.RETURNRESIDENTF parameter bit is on.

- LEVELF = [03:04]

  If TYPE.RETURNFULLNAMEF is turned off, this field indicates the level of the name pointed to by the LINKF field. If TYPE.RETURNFULLNAMEF is turned on, this field contains the number of levels in the name (excluding the <on part> variable).

  The number of simple form names in a file name represents the number of levels in the file name. For example, the following file name (in display form) has three levels:

  ```
  *A/B/C
  ```

  If you create this same name in standard form, it appears as 48"09020301C101C201C3", where the third byte contains the number of levels (3) in the file name. When an <on part> construct is part of the original file name, <on part> becomes an additional level to the file name. To indicate that the <on part> construct is present in a standard form name, bit 2 of byte 1 is turned on. For example, the display form file name of *AA/BB ON C appears in standard form as 48"0B060302C1C102C2C201C3". The security byte (byte 1) contains a 6 (with bit 2 turned on) and the number of levels (byte 2) is 3.

Fixed Info Link Word

Fixed info link words stored in the array ARY by GETSTATUS point to the attributes and status information for the corresponding file. A fixed info link word is returned instead of a file status word for SUBTYPE 0 calls or when GETSTATUS is reporting information for a file (fixed info link words are not used for directories) and if bits other than bit 0 are turned on in the MASK parameter. When a fixed info link word is returned, the second word of the fixed information is the file status word. Refer to Figure 4–1 and Table 4–1 for detailed information about fixed info link words.

If bit 47 of the word is turned off and field [3:04] contains 0, the word is a link to file information words (refer to Table 4–1 and Figure 4–1). Fixed info link words are structured as follows:

- ERRORF = [47:01]

  This value is 0.

- VALUEF = [38:06]

  FOR SUBTYPE 0 requests, VALUEF retains the original value supplied by the caller in the file pointer word.

- LINKF = [32:17]

  If LEVELF within this word is 0, this field is an absolute word index into the array and points to the validity mask word (refer to Figure 4–1).

- LEVELF = [3:04]

  The value is 0.

Figure 4–1 is a representation of the words in the array ARY that correspond to the MASK bits described in Table 4–1.

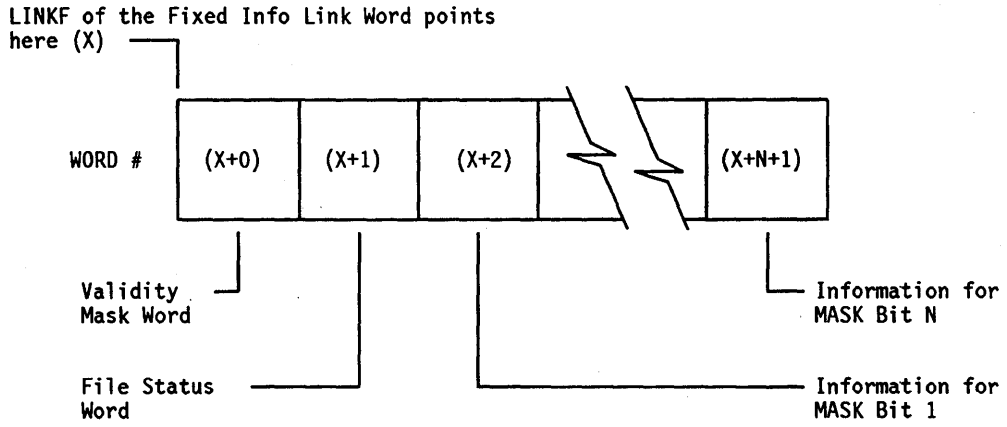

Figure 4–1. Format of Fixed Information for a File

GETSTATUS places information in words (X+2) through (X+N+1) only if the corresponding bit is turned on in the MASK parameter. Words (X+0) and (X+1) are always returned regardless of MASK bits.

When additional information is requested about a file (MASK bits other than bit 0 are turned on), the corresponding bit in the the validity mask word is turned on for each piece of data that GETSTATUS could return. The validity mask bits have a one-to-one mapping correspondence with the original MASK word. Bit 0 of the validity mask is always on, which means that word (X+1) contains a valid file status word.

When catalog information, archive information, or both catalog and archive information for a file exists, some or all of the file attributes for the backed up versions of the file might differ from the attributes for the resident file. For example, the FILEKIND

reported for the resident file in the file status word (refer to word 1 in Figure 4–1) might differ from the FILEKIND reported from a backed up catalog version of the file (refer to word 4 in Table 4–3), and they both might differ from the FILEKIND for a backed up version of the file listed in the archive directory (refer to word 1 in Table 4–5). The resident version and various backed up versions of a file are reported together because they all have the same file name. They do not necessarily have to be versions of the *same* file. Only selected information such as the file status word and information for MASK bits 9 and 16, MASK bits 13 and 34, or both is available if a file is not resident on disk, and a record for the file exists in the archive directory, or if the file is a cataloged file and a record exists for it in the catalog.

If the request is for information on more than one file and the array is not large enough to contain the complete information requested, bit ARY[0].[47:01] is turned on. GETSTATUS does not return the value TRUE for this case. ARY[0].[19:20] points one word beyond the last valid pointer word that was processed. If the array ARY is not large enough to contain the information for the first file pointer word, the GETSTATUS call returns hard error 41 (refer to Appendix A for a listing of hard error messages).

Tables 4–1 through 4–5 describe the information that can be returned when some parameter MASK bits are turned on. Table 4–1 shows the information returned in the array ARY for MASK bit The caller can turn bits on or off in the MASK parameter to indicate specific file information that is to be retrieved. The system takes the MASK parameter for each file and validates that the requested information is available and then indicates the pieces of information that were successfully retrieved by storing bits in the validity mask for the file at word ARY [X]. Table 4–2 shows the format of the returned catalog information. Table 4–3 describes the cataloged file generation information that is returned. Table 4–4 shows the format of archive information returned. Table 4–5 shows the format of archive backup information returned.

In Tables 4–1 through 4–5, the phrase *link to* means that the word points to information returned elsewhere in the array ARY. The LINKF field [32:17] of a normal link word contains the character index of the first byte of the information. The INFOF field [15:16] of a normal link word contains the length in bytes of the information.

Table 4–1. File Information

MASK Bit	Fixed Information Returned
[00:01]	MASK bit 0 is not used.
[01:01]	File attribute CREATIONDATE.
[02:01]	Disk blocking is HEADER [3] and contains the following information: • [47:16] = BLOCKSIZE in physical units • [31:16] = MINRECSIZE in physical units • [15:16] = MAXRECSIZE in physical units (See MASK Bit [14:01] later in this table for a definition of physical units.)

continued

Table 4–1. File Information (cont.)

MASK Bit	Fixed Information Returned
[03:01]	File attribute SAVEFACTOR.
[04:01]	Size (in words) of the disk file header that is returned if MASK bit 8 is turned on.
[05:01]	ROWSIZE of each disk area (in segments).
[06:01]	Bit [00:01] = File attribute IAD. Bit [01:01] = File attribute CRUNCHED. Bit [02:01] = File has a guardfile. Bit [04:01] = Files in resident program code file. Bit [05:01] = File has been subjected to an SL system command. Bit [06:01] = File is a system file. Bit [07:01] = File is a temporary file.
[07:01]	Number of rows allocated. A row is allocated if disk space is assigned to it. The number of allocated rows might be less than the number of rows reserved for the file (compare with MASK bit 10).
[08:01]	Link to an image of the disk file header of the file. Refer to the *I/O Subsystem Programming Guide* for the layout of the header. Be sure to note the warnings that accompany the description of the layout. Because the layout of the header is likely to change over time, Unisys recommends, in general, that you use this MASK bit only if it is necessary to obtain information from a header that cannot be obtained by setting other MASK bits.
[09:01]	Bit [01:02] = 1. This is a file. Bit [01:02] = 2. This is a directory. Bit [01:02] = 3. This is both a file and a directory. Bit [02:01] = 1. The disk file header for the file is available. Bit [03:01] = 1. The information is for a tape file (cataloged file on the TAPE family) Bit [04:01] = 1. The information is for a file on an offline disk family.
[10:01]	Number of rows available—that is, file attribute AREAS.
[11:01]	End-of-file (in segments), exclusive of a possible partial final segment.
[12:01]	End-of-file uses this number of bits in the last segment. This final segment is not counted in the word requested by MASK Bit 11.

continued

Table 4–1. File Information (cont.)

MASK Bit	Fixed Information Returned
[13:01]	SECURITYTYPE file attribute in field [19:20] through the following mapping: CASE (X.[19:20]) OF (1,2,3,0). SECURITYUSE file attribute in field [39:20] through the following mapping: CASE (X.[39:20]) OF (3,1,2,0).
[14:01]	This MASK bit is being replaced by bits [18:01] and [24:01]. It will be deimplemented in a subsequent release. Turning this bit on in the Mark 3.9 release causes the system to issue a run time warning. For the Mark 3.9 release, this bit contains the file information block word TANKDATA1 that consists of the following: • [42:03] = EXTMODE file attribute (see the note that follows) • [39:01] = UNITS file attribute • [38:04] = FILETYPE file attribute • [34:03] = SIZEMODE file attribute • [31:16] = SIZEOFFSET file attribute • [15:16] = SIZE2 file attribute The physical units of BLOCKSIZE, MAXRECSIZE, and MINRECSIZE are determined by the attribute UNITS in [39:01] of the above word. ***Note:*** *In the Mark 3.9 release, if the actual value of EXTMODE exceeds the value of ASCII (5), GETSTATUS returns the value of ASCII in this field. GETSTATUS returns the actual value of EXTMODE in field [15:16] associated with MASK bit [18:01].*
[15:01]	USEDATE file attribute (access date).
[16:01]	Special link to catalog information. The VALUEF field [38:06] = 1 if the resident file is cataloged. The LINKF field contains the word index of the validity mask for the catalog information. Refer to Table 4–2 for details.
[17:01]	Link to guardfile TITLE (in standard form).

continued

Table 4–1. File Information (cont.)

MASK Bit	Fixed Information Returned
[18:01]	The values of BLOCKSIZE, MINRECSIZE, and MAXRECSIZE are expressed in FRAMESIZE units. • [47:01] = UNITS file attribute. • [46:01] = 1. The CCSVERSION file attribute has been turned on. • [43:04] = FILETYPE file attribute. • [39:16] = CCSVERSION file attribute if bits [46:01] = 1. • [23:08] = Value of FRAMESIZE for the file. • [15:16] = EXTMODE file attribute.
[19:01]	File attribute VERSION.
[20:01]	File attribute CYCLE.
[21:01]	File TIMESTAMP. The format of the value is the same as that of the value returned by the ALGOL TIME(6) intrinsic function.
[22:01]	Number of segments allocated, computed with regard to the value of the file attribute CRUNCHED.
[23:01]	File is an APL file; the MA system command has been performed on it.
[24:01]	Size information for BLOCKSIZE = VARIABLEOFFSET for files. • [47:16] = SIZEMODE file attribute • [31:16] = SIZEOFFSET file attribute • [15:16] = SIZE2 file attribute
[25:01]	USETIME file attribute.
[26:01]	USERINFO file attribute.
[27:01]	ALTERDATE file attribute.
[28:01]	ALTERTIME file attribute.
[29:01]	CREATIONTIME file attribute
[30:01]	Unit number of the base pack of the family.
[31:01]	Link to an image of segment 0 of a code file. ***Note:*** *The segment 0 information is intended for use by system software. Unisys maintains the right to change the contents of segment 0 from release to release.*

continued

Table 4–1. File Information (cont.)

MASK Bit	Fixed Information Returned
[32:01]	Link to the list of deimplementation message numbers. The list consists of 16-bit binary numbers, with the last number being 0.
[33:01]	Link to an image of the row address (mass-address) words. The first word pointed to gives the total number of row address words (zero or more). Subsequent words contain the row address words beginning with row address word zero. Bit 43 of each row address word can be interrogated to determine if space has been allocated for the row. The format of each row address word is as follows: • [47:01] = 1 if the DMSII read lock flag is turned on for the row. • [46:01] = 1 if the DMSII write lock flag is turned on for the row. • [45:01] = 1 if the row is on a unit that is not write-enabled. • [44:01] = 1 if the family member on which the row was allocated has been deleted from the family by an operator RC (Reconfigure Disk) system command. • [43:01] = 1 if space has been allocated for the row. • [42:01] = 1 if the family index field was explicitly set requesting that space for the row be allocated on a specific family member. • [41:08] = The family index of the family member on which the row is or will be allocated (family indexes count from one). • [33:01] = 1 if the row is allocated and the family member on which it is allocated is online. In the row address words returned by this request, this bit is always 0. • [32:33] = If the row is allocated, this field contains the actual disk address of the first segment of the row.
[34:01]	FILEORGANIZATION file attribute.
[35:01]	Link to the NOTE file attribute.
[36:01]	Bit 1 is turned on if the file is restricted. Bit 2 is turned on if the base disk volume is restricted. Bit 3 is turned on if the base disk unit is restricted. Bit 4 is turned on if the disk family is restricted. Bit 5 is turned on if the file is a read only file, such as a system log file.
[37:01]	Link to the license key information.
[38:01]	Link to the RELEASEID attribute.

continued

Table 4–1. File Information (cont.)

MASK Bit	Fixed Information Returned
[39:01]	Link to the list of user defined disk file attributes. Refer to the description of the header layout in the *I/O Subsystem Programming Guide*.
[40:01]	Bits [47:01] = 1 if the BLOCKSTRUCTURE file attribute is valid. Bits [46:07] = Reserved. Bits [39:20] = The number of bytes between the end of an uncrunched area and the end of the sector in which the area ends. Bits [19:08] = BLOCKSTRUCTURE file attribute. If the file was created on an operating system that does not support the FILESTRUCTURE attribute (Mark 3.8 and earlier releases), bits [11:12] = 0. If the file was created on an operating system that supports the FILESTRUCTURE attribute, the following is applicable: • Bits [11:04] = 1 if the file was created with direct I/O. • Bits [11:04] = 2 if the file was not created with direct I/O and a value of FILESTRUCTURE was not explicitly designated when the file was created. • Bits [11:04] = 3 if the file was not created with direct I/O and a value of FILESTRUCTURE was explicitly designated when the file was created. • Bits [07:08] = FILESTRUCTURE attribute.
[41:01]	FILELENGTH file attribute.
[42:01]	Link to KEYEDIO information.
[43:01]	Link to a word list containing the privilege bits of the designated code file, as found in the disk file header. The first word corresponds to the special handling bits. The remaining words are the security bits. The special handling bits are turned on by the system commands *CP*, *SUPPRESS*, *LP*, and *RP*. The security bits are turned on by the system command *PP*. Refer to the *I/O Subsystem Programming Guide* for the layout of the bits.
[44:01]	DOCUMENTTYPE and PERMITTEDACTIONS file attributes: • [47:08] = The DOCUMENTTYPE attribute. • [39:01] = The DOCUMENTTYPE validity flag, if TRUE, indicates that the DOCUMENTTYPE has been set for the file. • [10:11] = The PERMITTEDACTIONS file attribute.
[45:01]	Reserved.

continued

Table 4–1. File Information (cont.)

MASK Bit	Fixed Information Returned
[46:01]	Link to a list of mix numbers of users who have the file open.
[47:01]	Special link to archive information. The LINKF field contains the word index of the validity mask for the archive information. Refer to Tables 4–4 and 4–5 for details.

Table 4–2 contains the catalog information corresponding to MASK bit 16 (the link to catalog information). If X points to the beginning of the fixed info link words, then Y is given by the following formula:

```
Y := ARY[X+17].LINKF;
```

This value points to the catalog information. For SUBTYPE 0 requests, you can limit the catalog information on a file-by-file basis by using ADDLWORD(1) as a selection mask.

Table 4–2. Format of Returned Catalog Information

Word	Contents
ARY[Y+0]	The validity mask. The bits that are turned on in this word indicate the words in the catalog information that are valid. Bit 0 corresponds to the first word of catalog information (catalog block number), bit 1 corresponds to the second word (serial number), and so on.
ARY[Y+1]	The block number of the catalog record for the file in the system catalog file.
ARY[Y+2]	The serial number of the disk family or tape volume that contains this file.
ARY[Y+3]	The number of file entries or generations under the catalog block.
ARY[Y+4]	A special link to the file or generations: • LINKF contains an absolute word index to the first file generation reported. • INFOF contains the number of file generations that are present. • MAXCATLEVELF of SUBCLASS controls the number of file entries returned. • ADDLINFOF contains the size of each entry in words. For details, refer to Table 4–3.

Table 4–3 contains the catalog information referenced in ARY[Y+4]. Each referenced file entry is structured according to the values in Table 4–3, where Z is the index of the first word of the file entry. The index of the first file entry is as follows:

```
Z := ARY[Y+4].LINKF;
```

Table 4–3. Format of File Generations Returned

Word	Contents
ARY[Z+0]	The validity mask. The bits that are turned on in this word indicate the words in the file generation information that are valid. Bit 0 corresponds to the first word of the generation information (number of reels), bit 1 corresponds to the second file entry (resident indicator), and so on.
ARY[Z+1]	Number of reels associated with this file (for tape files only).
ARY[Z+2]	Resident indicator. The value is 1 if this generation is for the resident disk file. The value is 0 if this file generation is for a tape file or a nonresident generation of a disk file.
ARY[Z+3]	Version and cycle of this generation.
ARY[Z+4]	For disk files, the FILEKIND attribute. For tape files, the content is 192.
ARY[Z+5]	Link to the backup list. ARY[Z+5].LINKF is a character index that points to one or two backup entries. ARY[Z+5].INFOF contains the total number of characters pointed to as follows: `4"0301" <kind>  4"0802" <serial number>` Each backup entry is 11 bytes long. The variable <kind> is the KIND value of the unit on which the backup appears and is a 1-byte binary number. The variable <serial number> is the SERIALNO value of the volume on which the backup appears and is 6 EBCDIC characters long.
ARY[Z+6]	TIMESTAMP attribute for this entry. The format returned is the same as for the ALGOL TIME (6) intrinsic.
ARY[Z+7]	Reserved.
ARY[Z+8]	Reserved.
ARY[Z+9]	For tape files, the last access date and time of access. The format returned is the same as for the ALGOL TIME (6) intrinsic.

Table 4–4 contains the archive information that corresponds to MASK bit 47, the link to archive information. If X points to the beginning of the fixed info link words, then R is given by the following formula:

```
R := ARY [X+48].LINKF;
```

This value points to the archive information.

Table 4–4. Format of Returned Archive Information

Word	Contents
ARY [R+0]	The validity mask. The bits that are turned on in this word indicate the words in the archive information that are valid. Bit 0 corresponds to the first word of archive information (timestamp), bit 1 corresponds to the second word (link to backup information), and so on.
ARY [R+1]	The date and time when the archive record for this file was last updated (or when the record was initially entered into the archive directory). The format returned is the same as for the ALGOL TIME (6) intrinsic.
ARY [R+2]	Special link to archive backup information for the file: • LINKF contains an absolute word index to the archive backup information. • INFOF contains the number of sets of backups returned. • ADDLINFO contains the size in words of each backup set. For details, refer to Table 4–5.

Table 4–5 contains the archive backup information referenced in ARY [R + 2]. The archive backup information gives a description of some of the file attributes of the named file at the time when the file was backed up or rolled out with a WFL *ARCHIVE* statement. The currently resident file, if any, might actually be a different file with completely different file attribute values, but it can still have the same name as the backed up file. The archive backup information also describes the tape media to which the backup was made. Archive backup information is structured according to the values in Table 4–5, where G is the index of the first word in the entry. The index G of the backup information can be calculated with the following formula:

```
G := ARY [R+2].LINKF;
```

Note that an archive record can contain references to one or two tape backups. In Table 4–5, the references to backups come in sets of 4 words, such and 9, 10, 11 and 12 or 13, 14, 15, and 16. The first word of each set corresponds to one backup. The second word of each set corresponds to the another backup, and so on. The backup entries are not ordered in the record. Any of the four backups can be missing. When all four are present, the most recent backup can be in the first, second, third, or fourth slot. You can use the date and time that appear in words 17 through 20 to determine which backup is most recent.

Table 4–5. Format of Archive Backup Information

Word	Contents
ARY [G+0]	The validity mask. The bits that are turned on in this word indicate the words in the archive backup information that are valid. Bit 0 corresponds to the first word of archive backup information (FILEKIND and security), bit 1 corresponds to the second word (SAVEFACTOR and organization), and so on.
ARY [G+1]	Bits [47:01] contain the FILEKIND value. Bits [35:08] contain an internal code for SECURITYTYPE as follows: • 0 = PUBLIC • 1 = GUARDED • 2 = CONTROLLED • 3 = PRIVATE Bits [27:08] contain an internal code for SECURITYUSE as follows: • 0 = IO • 1 = IN • 2 = OUT • 3 = SECURED
ARY [G+2]	Bits [47:08] contain the FILEORGANIZATION. Bits [19:20] contain the SAVEFACTOR.
ARY [G+3]	TIMESTAMP attribute for the backed up file. The format returned is the same as for the ALGOL TIME (6) intrinsic.
ARY [G+4]	If CYCLE (and VERSION) were designated for the file, bits [21:14] contain CYCLE and bits [07:08] contain VERSION.
ARY [G+5]	REAL number (in floating point form): file size in megabytes.
ARY [G+6]	File creation date and time. The format returned is the same as for the ALGOL TIME (6) intrinsic.
ARY [G+7]	Last access date and time. The format returned is the same as for the ALGOL TIME (6) intrinsic.
ARY [G+8]	Equal to 1 if the backed up file does not match the residential file. Equal to 0 if no resident file exists or if the resident and backed up versions of the file do match.
ARY [G+9]	Bits [47:08] contain the tape reel number on which one backup copy of this file starts.

continued

Table 4–5. Format of Archive Backup Information (cont.)

Word	Contents
	Bits [37:06] contain a code for the type of ARCHIVE backup: • 0 = EMPTY (The backup slot is available for the next ARCHIVE backup or merge entry for the file.) • 1 = FULL • 2 = DIFFERENTIAL • 3 = INCREMENTAL • 4 = ROLLOUT • 32 = MERGE Bit [31:01] is on if the backup tape has been overwritten. Each backup description has the TYPE, tape name, date and time of creation, and serial number list.
ARY [G+10]	Has the same format as ARY [G+9] for another backup.
ARY [G+11]	Has the same format as ARY [G+9] for another backup.
ARY [G+12]	Has the same format as ARY [G+9] for another backup.
ARY [G+13]	Link to the backup tape name in substandard form.
ARY [G+14]	Link to another backup tape name in substandard form.
ARY [G+15]	Link to another backup tape name in substandard form.
ARY [G+16]	link to another backup tape name in substandard form.
ARY [G+17]	Date and time when the archive process started. For installations that use the volume directory facility (such as the system command *SECOPT TAPECHECK=AUTOMATIC*, this value matches the value stored in word 15 of the volume directory record for the backup tape. The format returned is the same as for the ALGOL TIME (6) intrinsic.
ARY [G+18]	Same format as ARY [G+17] for another backup tape.
ARY [G+19]	Same format as ARY [G+17] for another backup tape.
ARY [G+20]	Same format as ARY [G+17] for another backup tape.
ARY [G+21]	Link to the tape SERIALNO list used for the backup file. Each SERIALNO in the list is 6 EBCDIC characters long.
ARY [G+22]	Same format as ARY [G+21] for another backup tape.
ARY [G+23]	Same format as ARY [G+21] for another backup tape.
ARY [G+24]	Same format as ARY [G+21] for another backup tape.

SUBTYPE 0 Calls (Searching for a Specific File)

SUBTYPE 0 can be used to retrieve status and attribute information about one or more disk files or cataloged tape files. For SUBTYPE 0 calls, be aware of the following:

- The SUBCLASS.MAXCATLEVELF parameter can be used to limit the number of cataloged file generations reported (refer to Table 4–3).
- You can use the MASK parameter to select the file attributes or status items that are to be reported (refer to Table 4–1).
- The array ARY must contain the names of the files. ARY can optionally contain dynamic MASK and SUBCLASS values. These dynamic values can be used as substitutes for the SUBCLASS and MASK parameters. Such a technique permits the caller to retrieve different attributes for different files with a single call. ARY can optionally contain an ADDLWORD for one or more of the file pointer words. For each file pointer word, ADDLWORD(1) supplies a mask to limit the specific catalog information retrieved (refer to Table 4–2); if ADDLWORD(1) is not present, all available catalog information is retrieved.

Input for SUBTYPE 0 Calls

Figure 4–2 shows a diagram of input with subtype 0 for the array ARY.

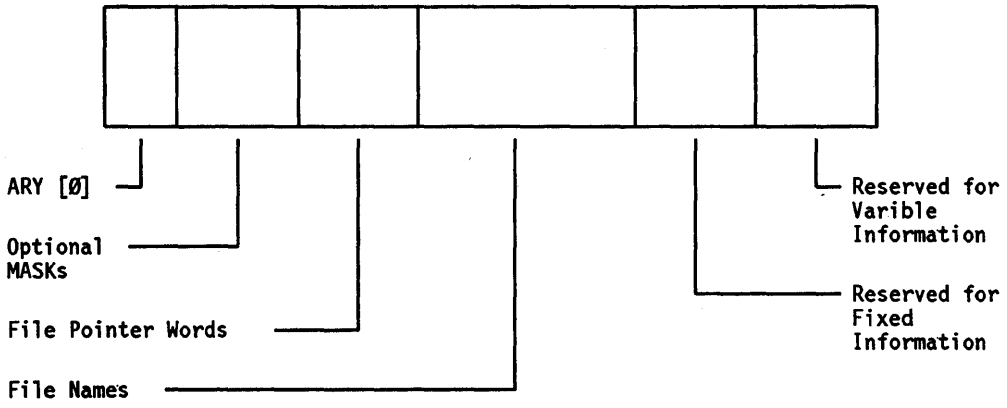

Figure 4–2. Input Diagram for SUBTYPE 0 Calls

For SUBTYPE 0 calls, the information in the array ARY should be structured as described in the following text.

ARY[0]

This word contains two numbers:

- ARY[0].[19:20]

  This field contains the count of the total number of file pointer words and ADDLWORD words plus 1. If ADDLWORD words are not used, ARY[0].[19:20] contains the number of file pointer words plus 1.

- ARY[0].[39:20]

  This field contains a count of the number of dynamic MASK and SUBCLASS words present plus 1. ARY[0].[39:20] should be 0 or 1 if dynamic MASK and SUBCLASS words are present; otherwise, it should be an odd number (an even number plus 1 because each reference requires a pair of words).

Optional MASK and SUBCLASS Values

These words contain optional dynamic MASK and SUBCLASS values. The number of words occupied by these values must be indicated by the value in ARY[0].[39:20].

File Pointer Words

The file pointer words and optional ADDLWORDs immediately follow the last dynamic MASK and SUBCLASS pair. If no dynamic MASK words exist, the first file pointer word must be in ARY [1]. Each file pointer word must be structured with the LINKF, VALUEF, and ADDLINFOF fields. In the following description, suppose that the file pointer word (FPW) is the next file pointer word in ARY.

FPW.LINKF must contain the character offset to the start of the file name. GETSTATUS locates the first byte of the file name for a given file pointer word with the following formula:

```
character index = ARY[Ø].[19:2Ø] * 6 + FPW.LINKF
```

FPW.VALUEF indicates whether or not an optional ADDLWORD(1) follows the file pointer word. If FPW.VALUEF contains a 0, all catalog information is returned for the file if MASK bit 16 is turned on (refer to Table 4–2). If FPW.VALUEF contains a 1, an ADDLWORD(1) is in the next word of ARY. That word is used as a mask to limit the catalog information returned.

FPW.ADDLINFOF indicates the optional dynamic MASK and SUBCLASS pairs that are to be used for the file. If FPW.ADDLINFOF is 0, the value of MASK and SUBCLASS is not changed. If ARY[0].[39:20] is greater than 1 and FPW.ADDLINFOF is not 0, the MASK and SUBCLASS are changed as follows:

```
MASK := ARY [FPW.ADDLINFOF];
SUBCLASS := ARY [FPW.ADDLINFOF + 1];
```

File Names

The file names should be placed in the same order as that of the corresponding file pointer words, immediately after the last file pointer word.

Fixed Information

GETSTATUS uses these reserved words to return simple file attributes and links to variable-length file attributes. GETSTATUS places the fixed information for all files, beginning in the word immediately after the first file name. The number of words required per file name depends on the MASK parameter and the amount of requested catalog information. For example, if the highest bit that is on in the MASK word is 38, then 39 words will be required for each file plus the words for catalog information. Catalog information for each file might require an additional 5 + 10 = 15 words (refer to Table 4–2 and Table 4–3).

Variable Information

GETSTATUS uses these reserved words to return variable-length file attributes. Sufficient space must be left at the end of the array ARY so that GETSTATUS can store variable-length information. For example, depending on the MASK parameter, GETSTATUS places guardfile names, row address words, catalog backup lists, and so on at the end of the array ARY.

Results Returned for SUBTYPE 0

If bit 0 of the Boolean result returned by GETSTATUS *B* equals 1,one of the following types of errors has occurred:

- B.[11:08] is not 0. GETSTATUS detected a hard error. Refer to Appendix A for a listing of hard errors.
- If B.[11:08] equals 0, a pointer word (other than ARY[0].[19:20]) within the array contains the ERRORF bit set, and subsequent investigation of the ERRORVALUE field provides the soft error number. Refer to Appendix B for a listing of soft errors.
- For example, if the first file is not present and ARY [0].[39:20] equals 0, ARY[1].ERRORF contains a 1 and ARY[1].ERRORVALUEF contains a 49 (NOFILE).

If all the file pointer words are processed, ARY [0].[19:20] is left unchanged. If processing stops because ARY does not have enough room for all the requested information for all the files, bit 47 of ARY [0] is turned on and ARY [0].[19:20] is made equal to the index of the file pointer word where processing stopped. The information for that file was not completed and should not be used.

GETSTATUS searches for each of the files referenced by file pointer words. GETSTATUS replaces each file pointer word processed with either a soft error word or a fixed info link word.

SUBTYPE 0 Examples

The following SUBTYPE 0 examples depend on these declarations:

```
DEFINE DISPLAYFORMNAMEF = [39:1] #,
       RETURNRESIDENTF  = [37:1] #,

       ERRORF    = [47:1] #,
       ADDLINFOF = [46:8] #,
       VALUEF    = [38:6] #,
       LINKF     = [32:17] #,
       INFOF     = [15:16] #;

 REAL FILW, GLW;          % FIXED INFO LINK WORD,  GENERATION LINK WORD
 REAL J, X, Y, Z, LEN, TIMESTAMP;
 BOOLEAN B;
 ARRAY ARY [0:4000];
```

This example shows how to retrieve all the information available about a file:

```
   ARY [0] := 2;                  % WORD INDEX OF FIRST FILE NAME
   ARY [1] := 0;                  %  FILE POINTER WORD
   REPLACE POINTER (ARY [2]) BY   %   FILE NAME
         "SYSTEM/DCALGOL ON SYS37.";
   B := GETSTATUS (3 & 1 DISPLAYFORMNAMEF, 0, REAL (NOT FALSE), ARY);
```

After GETSTATUS has been called, the results are available in B and ARY.

The example code that follows shows how to retrieve the TIMESTAMP entry for each generation of the cataloged file. The catalog generation TIMESTAMP attribute was chosen for the example because it illustrates how to code several layers of error checking and linking. Most file attributes are easier to access because they are available at the first level and do not require multiple links. To locate the TIMESTAMP entry for each generation of the file, proceed as follows. In Table 4–1, observe that bit 16 (word 17) links to the catalog information. In Table 4–2, observe that bit 3 (word 4) links from the catalog information to the generation information. In Table 4–3, observe that bit 5 (word 6) contains the TIMESTAMP.

```
IF NOT B OR REAL (B.[11:8]) = Ø THEN  % GETSTATUS CALL WAS SUCCESSFUL
  BEGIN
    FILW := ARY [1];     % PICK UP "FIXED INFO LINK WORD"
    X := FILW.LINKF;
    IF NOT BOOLEAN (FILW).ERRORF THEN  % FILE ENTRY WAS FOUND
      IF BOOLEAN (ARY [X].[16:1]) THEN   % CATALOG INFO IS THERE
        BEGIN
          Y := ARY [FILW.LINKF+17].LINKF;   % PICK UP CATALOG LINK
          IF BOOLEAN (ARY [Y].[3:1]) THEN   % GENERATION INFO OK
            BEGIN
              GLW := ARY [Y + 4];  % GET LINK TO GENERATION INFO
              Z := GLW.LINKF;      %  INDEX OF FIRST GENERATION
              LEN := GLW.ADDLINFOF;  % SIZE OF EACH GENERATION
              THRU GLW.INFOF DO       % LOOP OVER ALL GENERATIONS
                IF BOOLEAN (ARY [Z].[5:1]) THEN  % GOOD TIMESTAMP
                  BEGIN
                    TIMESTAMP := ARY [Z+6];   % GET TIMESTAMP

% % % % % % % % PROCESS TIMESTAMP OF THAT GENERATION AS NECESSARY

                    Z := Z + LEN;       % NEXT GENERATION
                  END;
            END;       % OF "IF GENERATION OK ..."
        END;        % OF "IF CATALOG INFO OK..."
  END;           % OF "IF GETSTATUS CALL OK..."
```

The next example illustrates how to retrieve information for three specific files with a single GETSTATUS call. This example also illustrates how to use dynamic MASK and SUBCLASS values in SUBTYPE 0 calls and how to use ADDLINFO(1) to selectively limit the catalog information to be retrieved.

Suppose that the three files are SYSTEM/SUMLOG ON DISK, (X)DATAFILE ON PACK, and SYSTEM/ALGOL ON SYS37. Suppose that you want to determine the file size (in segments) of the SUMLOG, the number of catalog generations for (X)DATAFILE, and the creation date of ALGOL. From Table 4–1, observe that MASK bit 22 corresponds to file size and bit 1 corresponds to creation date. To get catalog generations, MASK bit 16 selects catalog information, and from Table 4–2, ADDLWORD(1) bit 3 selects the count of generations. Turn on the RETURNRESIDENTF bit of the TYPE parameter because to get a count of the generation of (X)DATAFILE even if a resident generation does not exist.

```
 ARY [Ø] := 9              % LAST FILE POINTER WORD IS IN ARY [9-1]
                            % TWO PAIRS OF DYNAMIC "MASK" & "SUBCLASS"
           & 5 [39:2Ø];     % PRECEDE FIRST FILE POINTER IN ARY [5].

 ARY [1] := Ø & 1 [16:1]; % MASK BIT FOR CATALOG INFO
 ARY [2] := Ø;             % DUMMY SUBCLASS VALUE
 ARY [3] := Ø & 1 [1:1];  % MASK BIT FOR CREATION DATE
 ARY [4] := Ø;             % DUMMY SUBCLASS VALUE

% FILE POINTER WORDS
 ARY [5] := Ø & (6 * (1Ø - 9)) LINKF; % SUMLOG
 ARY [6] := Ø & (6 * (2Ø - 9)) LINKF  % (X)DATAFILE
              & 1 ADDLINFOF             % LINK TO DYNAMIC "MASK"
              & 1 VALUEF;               % REF TO ADDLWORD (1)
 ARY [7] := Ø & 1 [3:1];                % ADDLWORD (1) FOR DATAFILE
 ARY [8] := Ø & (6 * (3Ø - 9)) LINKF  % SYSTEM/ALGOL
              & 3 ADDLINFOF;           % LINK TO DYNAMIC "MASK"

% NOTE -- THERE CAN BE SPACES BETWEEN THE FILENAMES AND THERE CAN BE
%         SPACE BETWEEN THE LAST FILE POINTER WORD
%         AND THE FIRST NAME.
 REPLACE POINTER (ARY [1Ø]) BY "SYSTEM/SUMLOG.";
 REPLACE POINTER (ARY [2Ø]) BY "(X)DATAFILE ON PACK.";
 REPLACE POINTER (ARY [3Ø]) BY "SYSTEM/ALGOL ON SYS37.";

 B := GETSTATUS (3 & 1 DISPLAYFORMNAMEF & 1 RETURNRESIDENTF,
                 Ø, Ø & 1 [22:1], ARY);
```

For additional examples of SUBTYPE 0 calls, refer to "Hypothetical Examples for File and Directory Searches" in this section.

SUBTYPE 1, 2, and 4 Calls (Searching for Files under a Given Directory)

SUBTYPEs 1, 2, and 4 return information about files and directories. The caller indicates the files and directories that are to be reported by designating one major directory name and family name under which GETSTATUS should search. GETSTATUS returns file status words for directories that are found twice under the specified directory. GETSTATUS also returns file status words and other file information (as selected by the MASK parameter) for files that are found twice under the specified directory, and file status words and other file information for nodes found that are both directories and file names. Each file status word that is returned indicates the name and level of the node and whether it describes a directory, a file, or a file and a directory.

Request Format for SUBTYPE 1, 2, and 4 Calls

You can designate that the directory to be searched in one of four different ways:

- Searching a specifically named subdirectory
- Searching the entire usercode directory
- Searching the entire system directory
- Searching both the entire usercode directory and the entire system directory

If you want to search a specifically named subdirectory, you can designate the name of the directory in either display form (see TYPE.DISPLAYFORMF) or standard form.

You can also designate that the entire usercode directory (containing all files under usercodes), the entire system directory (containing all files without usercodes and directory), or both be searched. In these cases, the input directory name is special and must be in standard form. For information about standard form, refer to information on the DISPLAYTOSTANDARD function in the *DCALGOL Reference Manual*. The usercode information field of the <qualification> variable in the standard-form name should be as follows:

Variable	Description
3	Searches all usercode directories. A process that is not privileged and was not started from an ODT will receive a "NO FILE" soft error for this request.
2	Searches the nonusercode directory, and, depending on the privileges of the process, all usercode directories. If the process is not privileged and was not started from an ODT, only the directory for nonusercoded files is reported. Otherwise, if the process is privileged or was started from an ODT, all files (usercoded and nonusercoded) are reported.
1	Searches the usercode directory, nonusercode directory or both. A process that is not running under a usercode, is not privileged, and was not started from an ODT gets only files from the nonusercode directory. A process that is not running under a usercode but is privileged or was started from an ODT gets files from both the usercode and nonusercode directories. A process that is running under a usercode gets files from under the process usercode directory if any exists. If no files exist under the process usercode directory, files from the nonusercode directory are reported.

These three standard form directory names are respectively 48"030300", 48"030200", and 48"030100".

If the family name is also specified, the <total length> variable in the standard form name should be increased, bit [2:01] of the <qualification> byte should be turned on, the <number of identifiers> should be 1, and the <ID length> and <identifiers> for the family name should be appended to the code for the directory.

For example, in order to search the nonusercode directory on PACK, the standard form name would be the following:

```
48"ØØØ6Ø1Ø4" 8"PACK"
```

Input Array ARY Requirements for SUBTYPE 1, 2, and 4 Calls

Figure 4–3 shows a diagram of input for SUBTYPEs 1, 2, and 4 of the array ARY.

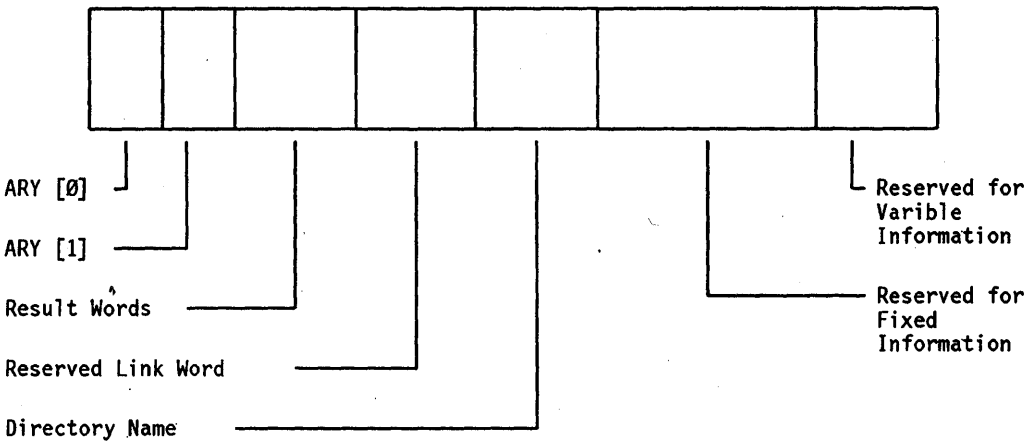

Figure 4–3. Input Diagram for SUBTYPE 1, 2, and 4 Calls

For SUBTYPE 1, 2, and 4 calls, the caller should structure the array ARY as follows:

ARY[0]

ARY[0].[19:20] should contain the number plus 2 of files, directories, or both to be reported.

ARY[1]

ARY[1].LINKF should contain the character offset to the name of the directory to be searched. GETSTATUS locates the start of the directory name as follows:

```
POINTER (ARY) + (ARY [Ø].[19:2Ø] *6 + ARY [1].LINKF)
```

Normally, ARY [1].LINKF should be 6 so that one word is reserved for the family link that is to be placed after the result words.

Result Words

These words are reserved so that GETSTATUS can place file status words, fixed info link words, or both for the files and directories found by GETSTATUS. The number of result words reserved is given by ARY[0].[19:20]–2.

Reserved Link Word

GETSTATUS returns the family name of the family that was searched. The word immediately after the last word reserved for result words is used by GETSTATUS to return a link to the family name used.

If a word is not reserved for this link, then after GETSTATUS picks up the directory name, GETSTATUS overwrites the first word of the name with the link to the family name.

Directory Name

The name of the directory to be searched is located, as previously described under the explanation of ARY[1].

Fixed Information

These reserved words are used by GETSTATUS to return simple file attributes and links to variable-length file attributes. GETSTATUS places the fixed information for all files beginning in the word immediately after the first directory name. The number of words required per directory name depends on the MASK parameter and the amount of catalog information requested. For example, if the highest bit that is on in the MASK word is 38, then 39 words are required for each file plus the words for catalog information. Catalog information for each file might require an additional 5 + 10 = 15 words (refer to Table 4–2 and Table 4–3).

Variable Information

These reserved words are used by GETSTATUS to return variable-length file attributes. Sufficient space must be left at the end of the array ARY so that GETSTATUS can store variable-length information. For example, GETSTATUS uses these words to return the family name and file names. Also, depending on the MASK parameter, GETSTATUS places guardfile names, row address words, catalog backup lists, and so on at the end of the array ARY.

Results Returned for SUBTYPE 1, 2, and 4 Calls

If the GETSTATUS call returns a Boolean 0, no errors were detected in the input parameters, and no file errors are reported in the array. If bit 0 of the Boolean result returned is turned on, either a hard error or a soft error was detected. If bits [11:08] of the Boolean result are not all 0, a hard error was detected, and any information returned in the array should not be relied upon.

When no hard errors are detected, GETSTATUS stores special information in ARY[0], as follows:

- Bit ARY[0].[47:01] is turned on if there were not enough reserved words in the array for GETSTATUS to return information about all the directories and files under the requested directory. Use a continuation request to retrieve information for the remaining files.
- ARY [0].[19:20] contains the value of the index plus 1 of the last result word returned.

GETSTATUS stores special information in ARY[1]. If an error such as "NO FILE" is detected on the initial name, ARY[1].ERRORF equals 1, and ARY[1].ERRORVALUEF indicates the type of soft error that occurred. Refer to Appendix B for a list of soft errors. Otherwise, GETSTATUS stores the following information in ARY[1]:

- ARY[1].[43:11] contains a word index that points to a link word. LINKF of that word is a character index that points to the family name stored as a simple form name. GETSTATUS returns the name of the actual family used (after family substitution).
- If TYPE.RETURNFULLNAMEF equals 0, ARY[1].NEXTLEVELLINKF (ARY[1].[15:11]) points to a file status word for the first node of the last complete file name. The last node in the list contains a NEXTLEVELLINKF field value of 0.
- ARY[1].LINKF is controlled through TYPE.RETURNFULLNAMEF as follows:
  - If TYPE.RETURNFULLNAMEF equals 0, LINKF is not altered from the original input call.
  - If TYPE.RETURNFULLNAMEF equals 1, LINKF points to the last file name for which the requested information was completely returned.
- ARY[1].LEVELF contains the number of levels that were present in the original directory name. To make a continuation call (SUBTYPEs 2 and 4), you should retrieve this number and pass it in the SUBCLASS.ORGLEVELF parameter.

GETSTATUS stores either one soft error word, one file status word, or one fixed info link word into the array ARY for each file or directory that is located. These words are placed in the area reserved for result words in the array ARY, starting at ARY[2]. GETSTATUS stores a number in ARY[0].[19:20] that is one greater than the index of the last soft error word, file status word, or fixed info link word returned.

GETSTATUS places file status words in the reserved area for each directory located. GETSTATUS also places file status words in the reserved area for each file located if the MASK parameter does not have any bits turned on other than bit 0. If other MASK bits are on, GETSTATUS stores fixed info link words in the reserved area for each file that is located.

SUBTYPE 2 and 4 Calls (Continuation of a Directory Search)

Because the array ARY must have a practical length, the GETSTATUS call might not have enough array space to provide all the names (with or without additional MASK information) that are present in a specified directory. After a call where SUBTYPE equals 1, 2, or 4, GETSTATUS turns on bit ARY[0].[47:01] if one of the following occurs:

- Not enough result words were reserved to report all the files and directories present. That is, the value supplied in ARY[0].[19:20] was not large enough.
- Not enough words were reserved at the end of the array ARY for GETSTATUS to place all the information available for the files that were located.

GETSTATUS does not return an error (bit 0 of the Boolean result returned is turned off) for these out-of-space conditions.

If there are not enough reserved words in the array for GETSTATUS to return complete information for the first file or directory located, hard error 41 is returned. Refer to Appendix A for a listing of hard error codes.

If GETSTATUS does not return a hard error, it returns a value in ARY[0].[19:20] that is 1 greater than the last complete result word that was placed in the array. The caller can examine and and process those result words and, if bit ARY[0].[47:01] is on, request information about the remaining files with a continuation request. By repeatedly issuing continuation requests, the caller can eventually retrieve information for all the files in a directory.

The two methods for continuing a GETSTATUS directory search are the following:

- TYPE.SUBTYPEF = 2
- TYPE.SUBTYPEF = 4

SUBTYPE 2 Calls

A SUBTYPE 2 continuation call is quite similar to an initial SUBTYPE 1 call. However, there are two differences:

- The SUBCLASS.ORGLEVELF parameter should contain a count of the number of levels in the original directory name. The proper value can be retrieved from ARY[1].LEVELF after a successful SUBTYPE 1, 2, or 4 directory call.
- The last file name or directory name reported for the processing directory call should be supplied in place of the original directory name.

For a SUBTYPE 2 call, the TYPE, SUBCLASS, and MASK parameters are the same as for a SUBTYPE 1 call, except that TYPE.SUBTYPEF = 2, and SUBCLASS.ORGLEVELF must be set up as previously described. Furthermore, as with SUBTYPE 1, a count of the number of files or directories to report plus 2 should be stored in ARY[0]; ARY[1].LINKF should point to the file or directory name from which the continuation search is to proceed.

Note: Coding a SUBTYPE 2 call is as easy as coding a SUBTYPE 1 call. A SUBTYPE 2 call can be used to start a search in the middle of a directory. It is not required that a SUBTYPE 2 call be preceded by another SUBTYPE 2 or SUBTYPE 1 call.

SUBTYPE 4 Calls

Unlike a SUBTYPE 2 continuation call, a SUBTYPE 4 continuation call requires that the result information in the array ARY from the preceding SUBTYPE 1 or SUBTYPE 4 call be intact. For SUBTYPE 4, GETSTATUS traverses the array ARY and automatically determines the file name to continue searching from. The contents of ARY[0].[19:20] should be restored to the count of files to be reported plus two before the SUBTYPE 4 call is issued. The value of the TYPE parameter passed for the SUBTYPE 4 call must be the same as it was in the initial SUBTYPE 1 call, except that the SUBTYPEF field is changed to 4. The array should not be altered between the TYPE.SUBTYPEF = 1 and the TYPE.SUBTYPEF = 4 calls, except for reinstating ARY[0].[19:20].

SUBTYPE 1, 2, and 4 Examples

The information that follows contains two kinds of examples: real programming (code) examples and hypothetical examples. (Refer to Appendix D for a complete example of the GETSTATUS directory interface, including continuation calls and attribute gathering.)

Programming Examples

The following SUBTYPE 1, 2, and 4 examples depend on these declarations:

```
DEFINE RETURNFULLNAMEF  = [40:1] #,
       DISPLAYFORMNAMEF = [39:1] #,
       ONLYSYSTEMFILESF = [38:1] #,
       RETURNRESIDENTF  = [37:1] #,
       SUBTYPEF         = [15:8] #,

       ORGLEVELF = [38:19] #,
       MAXLEVELF = [19:20] #,

       ERRORF    = [47:1] #,
       ADDLINFOF = [46:8] #,
       VALUEF    = [38:6] #,
       LINKF     = [32:17] #,
       INFOF     = [15:16] #,
       LEVELF    = [3:4] #;

 REAL FILW;              % FIXED INFO LINK WORD
 REAL J;
 BOOLEAN B;
 ARRAY ARY [0:4000];
```

The following example illustrates how to search all files under a directory and how to process the results. Initially, a SUBTYPE 1 directory search call is issued, and then repeated SUBTYPE 4 continuation directory search calls are made until all files have been reported. The example shows a directory search for all files under the directory A/=. The information retrieved by GETSTATUS is processed to count the total number of files (resident and nonresident) under the directory A/= and to count the total number of segments in use by resident files under the directory A/=.

```
REAL FILES, SEGS, SUBTYPE;
LABEL LOOP;

 REPLACE POINTER (ARY [53]) BY   %   DIRECTORY NAME
          48"Ø5Ø1Ø1Ø1" "A";
 ARY [1] := Ø & 6 LINKF;
 SUBTYPE := 1;                  % FIRST CALL IS SUBTYPE 1

LOOP:    % CONTINUE WITH MORE

 ARY [Ø] := 52;                 % ROOM TO REPORT 5Ø FILES PER CALL

 % RETURNRESIDENTF IS ON SO THAT BOTH RESIDENT AND CATALOGED
 % NONRESIDENT FILES WILL BE REPORTED.  MASK BIT 22 REQUEST COUNT OF
 % SEGMENTS IN USE BY FILE.

 B := GETSTATUS (3 & SUBTYPE SUBTYPEF & 1 RETURNRESIDENTF,
                 Ø, Ø & 1 [22:1], ARY);

 IF NOT B OR REAL (B.[11:8]) = Ø THEN
   IF NOT BOOLEAN (ARY [1].ERRORF) THEN
     BEGIN                                  % CALL WAS SUCCESSFUL

       J := ARY [Ø].[19:2Ø];      % INDEX OF LAST RESULT WORD PLUS 1
       WHILE J:= * - 1 GEQ 2 DO
         IF NOT BOOLEAN (ARY [J].ERRORF) THEN   % GOOD RESULT WORD?
           BEGIN
             IF ARY [J].LEVELF = Ø THEN    % FIXED INFO LINK WORD?
               BEGIN
                 FILES := *+1;
                 FILW := ARY [J];
                 IF BOOLEAN (ARY [FILW.LINKF]).[22:1] THEN  % SEGS OK
                   SEGS := * + ARY [FILW.LINKF + 23];
               END;
           END;         %  END OF LOOP OVER RESULT WORDS

       IF BOOLEAN (ARY [Ø].[47:1]) THEN    % MORE FILES TO COME
         BEGIN
           SUBTYPE := 4;
           GO LOOP;           % CONTINUE WITH NEXT BATCH OF FILES
         END;

     END;
```

The following example shows how to search for all files under the usercode directory of the process. If there are no files under the usercode directory of the process, files under the directory for nonusercoded files are reported instead. If the process is not privileged or was started from an ODT, only public nonusercoded files would be returned. If the process is not running under a usercode, then either all files (usercoded and nonusercoded) or all nonusercoded files are returned, depending on whether the process is privileged or is started from an ODT.

```
ARY [Ø] := 5Ø2;                  % ROOM TO REPORT 5ØØ FILES PER CALL
ARY [1] := Ø & 6 LINKF;

% QUALIFIER BYTE = 4 + 1 = 5:  "ON ..." + USERCODE, *, or both
REPLACE POINTER (ARY [5Ø3]) BY 48"Ø8Ø5Ø1Ø4" "PACK";
B := GETSTATUS (3 & 1 SUBTYPEF, Ø, Ø, ARY);
```

The following example shows how to determine that usercodes that have files on a family named PACK. The request will work only if the process is privileged or is started from an ODT; otherwise a "NO FILE" soft error would be returned.

```
ARY [Ø] := 5Ø2;                  % ROOM TO REPORT 5ØØ FILES PER CALL
ARY [1] := Ø & 6 LINKF;

% QUALIFIER BYTE OF NAME = 4 + 3 = 7:  "ON ..." + USERCODED
REPLACE POINTER (ARY [5Ø3]) BY 48 "Ø8Ø7Ø1Ø4" "PACK";

% SUBCLASS.MAXLEVELF IS USED SO THAT JUST FIRST LEVEL
% (USERCODE NAMES) ARE RETRIEVED.
B := GETSTATUS (3 & 1 SUBTYPEF, Ø & 1 MAXLEVELF, Ø, ARY);
```

The following example shows how to search for all files under the nonusercoded directory on the family named MCPMAST:

```
ARY [Ø] := 5Ø2;                  % ROOM TO REPORT 5ØØ FILES PER CALL
ARY [1] := Ø & 6 LINKF;

% QUALIFIER BYTE OF NAME = 4 + 2 = 6:  "ON ..." + *
REPLACE POINTER (ARY [5Ø3]) BY 48"ØB Ø6Ø1Ø7" "MCPMAST";

% USE TYPE.ONLYSYSTEMFILESF SO THAT USERCODED FILES WON'T BE REPORTED
% (OTHERWISE, THEY WOULD BE REPORTED IF THE PROCESS IS PRIVILEGED
% OR WAS STARTED FROM AN ODT).
B := GETSTATUS (3 & 1 SUBTYPEF & 1 ONLYSYSTEMFILESF,
                Ø, Ø, ARY);
```

The following example shows how to construct a SUBTYPE 2 continuation request that will continue searching from SYSTEM/COBOL under the SYSTEM/= directory. (Neither SYSTEM/COBOL nor files preceding it in the directory will be reported.)

```
ARY [Ø] := 5Ø2;                % ROOM TO REPORT 5ØØ FILES PER CALL
ARY [1] := Ø & 6 LINKF;

% NAME OF FILE FROM WHICH CONTINUATION IS TO COMMENCE
REPLACE POINTER (ARY [5Ø3]) BY "SYSTEM/COBOL ON DISK.";
B := GETSTATUS (3 & 2 SUBTYPEF & 1 DISPLAYFORMNAMEF
                  & 1 RETURNFULLNAMEF,
                Ø & 1 ORGLEVELF,              % SYSTEM/=
                Ø, ARY);
```

Hypothetical Examples for File and Directory Searches

The following examples illustrate the effects of the various parameters on directory GETSTATUS calls. Unless otherwise indicated, assume that, for each example, the process is running without a usercode and without a family substitution statement. Also assume that all bits in the TYPE, SUBCLASS, and MASK parameters are off (except that the TYPEF [7:08] field of the TYPE parameter is 3 and the SUBTYPEF [15:08] field of the TYPE parameter is as specified in the examples that follow).

The examples used here are mainly for SUBTYPEs 0 and 1 because SUBTYPEs 2 and 4 give the same results as SUBTYPE 1. In the examples that follow, the result is described as either a soft error word, a file status word, or a fixed info link word. Remember that whenever a file status word is returned in the array ARY, the word links to a node name or a complete title, depending on the TYPE.RETURNFULLNAMEF parameter. Whenever a fixed info link word is returned in the array ARY, the word links to additional information for the file, including a file status word.

In the examples that follow, the *level* of a file status word means the value in the LEVELF ([3:04]) field of the file status word.

SUBTYPE 0 Call with a Directory Name

Suppose that there is a resident file named A/B/C, and suppose that there is no file named A/B. The response to a SUBTYPE 0 call for file A/B would be a fixed information link word that links to a level 2 file status word for the directory A/B.

SUBTYPE 1 Call with a File Name

Suppose that there is a file named A/B and that there are no files under the A/B/= directory. A SUBTYPE 1 call for the directory A/B would return a level 1 file status word for the directory A and a level 2 file status word for the file B.

Suppose that there is a file named A and no files under the A/= directory. The response to a SUBTYPE 1 call for the directory A/B would be a soft error word with error code 49 ("NO FILE").

SUBTYPE 1 Call with MASK Bits

Suppose that there is a file named A/B/C that is not a code file. Suppose that a SUBTYPE 1 call for the directory A/= is issued with MASK bit 31 turned on (request for a copy of segment 0 of a code file) and all other MASK bits turned off. GETSTATUS would return a level 1 file status word for A, a level 2 file status word for B, and a fixed info link word for C. The validity mask word for the linked information would have bit 0 on for the level 3 file status word that follows. Bit 31 of the validity mask word would be off because segment 0 information is not available for noncode files.

SUBTYPE 1 Call with TYPE.RETURNFULLNAMEF Turned On

Suppose that there is a file named A/B/C. The response to a SUBTYPE 1 call for the directory A with TYPE.RETURNFULLNAMEF (TYPE.[40:01]) turned on and TYPE.DISPLAYFORMNAMEF (TYPE.[39:01]) turned on would be a level 3 file status word for A/B/C.

SUBTYPE 1 Call Showing the Effects of Usercode and Family Name on Level

TYPE.RETAINUSERCODEF increases by 1 the level reported for files found under the usercode directory of the process, but TYPE.LINKINONPARTF does not affect the level reported.

Suppose that the process is running under the usercode X and that there is a disk file named (X)A/B/C:

- A SUBTYPE 1 call for A would return a level 1 file status word for A, a level 2 file status word for B, and a level 3 file status word for C.
- A SUBTYPE 1 call for A with TYPE.RETAINUSERCODEF turned on would return a level 1 file status word for X, a level 2 file status word for A, a level 3 file status word for B, and a level 4 file status word for C.
- A SUBTYPE 1 call for (X)A with RETURNFULLNAMEF turned on but TYPE.RETAINUSERCODEF turned off would return a level 3 file status word for A/B/C.
- A SUBTYPE 1 call for A with TYPE.LINKINONPARTF turned on, TYPE.RETURNFULLNAMEF turned on, and TYPE.RETAINUSERCODEF turned off would return a level 3 file status word for A/B/C ON DISK.

SUBTYPE 1 Call Using SUBCLASS.MAXLEVELF

The results of SUBCLASS.MAXLEVELF are unaffected by RETAINUSERCODEF and LINKINONPARTF. Suppose that the process is running under usercode X and that there is a file named (X)A/B/C. For the examples that follow, TYPE.RETURNFULLNAMEF is turned on:

- A SUBTYPE 1 call for A with SUBCLASS.MAXLEVELF = 1 would return a level 2 File status word for A/B.
- A SUBTYPE 1 call for A with SUBCLASS.MAXLEVELF = 1 and TYPE.RETAINUSERCODEF turned on would return a level 3 file status word for (X)A/B:
- A SUBTYPE 1 call for A with SUBCLASS.MAXLEVELF = 1 and TYPE.RETAINUSERCODEF turned on, and TYPE.LINKINONPARTF turned on would return a level 3 file status word for (X)A/B ON DISK.
- A SUBTYPE 1 call for A ON DISK with SUBCLASS.MAXLEVELF = 1 would return a level 2 file status word for A/B.

Effects of TYPE.USERCODEONLYF on All SUBTYPEs

For the following cases, suppose that the process is running under the usercode X, that bit TYPE.USERCODEONLYF is on, and that there is a resident public file named *A/B but no directory named (X)A/= or no file named (X)A/B:

- A SUBTYPE 0 call for A/B would return a "NO FILE" soft error word.
- A SUBTYPE 0 call for *A/B would return a fixed info link word for *A/B.
- A SUBTYPE 1 call for A/B would return a "NO FILE" soft error word.
- A SUBTYPE 1 call for * A/B would return a level 1 file status word for *A and a level 2 file status word for B.

Effects of TYPE.SYSTEMFILESONLYF (SUBTYPE 1, 2, and 4 only)

For any SUBTYPE 1, 2 or 4 call, no file or directory with a usercode is returned if the TYPE.SYSTEMFILESONLYF bit is turned on.

Suppose that there is a resident public file named (X)A/B. A SUBTYPE 1 call for (X)A or (X)A/B with TYPE.SYSTEMFILESONLYF turned on would return a "NO FILE" soft error word. The result would still be "NO FILE" whether the process was running under usercode X, another usercode (privileged or not), or without a usercode.

SUBTYPE 5 Calls (Copying a Volume Library)

SUBTYPE 5 can be used to copy the entire contents of a volume library to a specific file. You do not need to have privileged status to use this call.

Input

The following is the array input needed for SUBTYPE 5:

ARY	Value or Description
ARY [0]	2
ARY [1]	0
ARY [2]	The beginning of the <file name> in which that copy is to be placed. The name can be in display or standard form, depending on TYPE.DISPLAYFORMNAMEF. You can also designate an <on part> construct.

Results Returned

The following information is returned when a volume library is copied:

- The file created has a MAXRECSIZE of 30 words. The row size is derived from the directory and is currently 1200 segments. The file is CRUNCHED.
- The first record in the copied file contains the same information as described in "SUBTYPEs 3 and 7 (Copying a System Directory)" earlier in this section. In addition, word 8 contains the length (in number of words) used beginning at record 1.
- Record 1 contains (one word each) the information necessary to locate valid rows in the copied file. The number of words to be used is determined by word 8 of record 0. This value can exceed 30 words. Each word contains the beginning record address of a valid row.
- The organization of the volume library allows the copying of certain rows from the catalog file of the system; therefore, rows that do not contain information exist within the copied file. In fact, these rows are not even allocated space. The information from record 1 must be used to determine which rows are valid.
- Within a valid row, all information within that same row of the catalog of the system is copied. This copying is necessary because of linking that may have been generated within the volume library. The format of the information within a specific row can be obtained from the description of the volume library.

SUBTYPE 6 Calls (Disk Utilization)

SUBTYPE 6 returns various information about sectors and areas on a family. This call corresponds to the *DU (Disk Utilization)* system command.

Input

The following array input information is needed for SUBTYPE 6:

ARY	Value or Description
ARY [0]	Index of the the first word of the family name. The value must be greater than or equal to 3. The family name must be in standard form.
ARY [1]	Family index of the unit within the family whose available space is to be returned. If 0 (zero), information about the entire family is returned.
ARY [2]	If SUBCLASS = 1, the value contained in ARY [2] is used as the specified size to base comparisons on. If SUBCLASS = 0, then the default size of 504 segments is used as the specified size.

Results Returned

The following information is returned by SUBTYPE 6:

Word	Contents
Word 1	The total number of available sectors.
Word 2	The size, in sectors, of the largest available area.
Word 3	The number of areas smaller than the specified size.
Word 4	The total number of sectors in areas smaller than the designated size.
Word 5	The number of areas larger than or equal to the designated size.
Word 6	The total number of sectors in areas larger than or equal to the designated size.
Word 7	The number of disk rows (areas) of the designated size that can be allocated.
Word 8	The default size used when SUBCLASS = 0. This is currently equal to 504 sectors.
Word 9	Total disk capacity in sectors of the family member designated in ARY [1]. If ARY [1] is 0, the value is the capacity of the entire family.

If the SIZE(ARY) is greater than 16, GETSTATUS returns the following additional information:

Word	Contents
Words 11 – 15	These words correspond exactly to words 3 through 7, except that the default number of sectors per area for STREAM and BLOCKED file structures is used as the designated size.
Word 16	The default number of sectors per area for STREAM and BLOCKED file structures that was used in constructing words 11 through 15.

SUBTYPE 7 Calls (Copying a System Directory)

SUBTYPE 7 can be used to copy a system directory. You must have privileged user status to use this call.

Input When Using the Unit Number

ARY	Description
ARY[0]	Value should be 6.
ARY[1]	0 for either a memory disk or a regular disk pack. 0 & 1 [15:08] for a memory disk. 0 & 17 [15:08] for a regular disk pack.
ARY[2]	The unit number.
ARY[3]	Not used.
ARY[4]	Not used.
ARY[5]	Not used.
ARY[6]	The beginning of the file name in which the copy is to be placed. The file name can be in display or standard form, depending on TYPE.[39:01]. You can also designate an <on part> construct.

Input When Using the Disk FAMILYNAME

ARY	Description
ARY[0]	Value should be 6.
ARY[1]	0 & 18 [32:17].
ARY[2]	Not used.
ARY[3] through ARY [5]	Disk FAMILYNAME in substandard form.
ARY[6]	The beginning of the file name in which the copy is to be placed. The file name can be in display or standard form, depending on TYPE.DISPLAYFORMNAMEF. You can also designate an <on part> construct.

Results Returned

The file created has a MAXRECSIZE of 30 words. The row size is derived from the directory and is currently 600 sectors.

The first record in the file contains the following:

Word	Description
Word 0	Serial number of the unit that contains the family that was copied.
Word 1	External device unit number from which the copy was performed.
Word 2	Date on which the copy operation was performed (in binary and in the form YYDDD).
Word 3	Time of day that the copy operation was performed (in 2.4-microsecond intervals of time).
Words 4 through 6	The family that was copied (in substandard form).
Word 7	Record address within the file that contains the first copied information.

The remainder of the file beginning at record 1 contains an exact copy of all the in-use headers in that family. In the following explanations, HDR is an array into which a record of the file has been placed. The version number of a header is contained in HDR[5].[47:04].

Examples of the contents of Version 6 headers are provided in the following list, with file attributes capitalized. The layout of the Version 6 header is described in the *I/O Subsystem Programming Guide*.

- The header in-use flag is 4"3F3F" in HDR[0].[47:16].
- The header block length is in HDR[0].[31:11], which is the length of the record in words.
- The FILEKIND is in HDR[1].[35:12].
- The title of the file in standard form is at POINTER (HDR [HDR[8].[35:12]], 8).
- The EXTMODE at file creation is stored in HDR[2].[42:3].
- HDR[2].[39:1] is turned on if the UNITS file attribute was TRUE at file creation.
- The BLOCKSIZE (in words or characters, as specified by UNITS) is in HDR[3].[47:16].
- The AREAS, or number of rows, of the file is HDR[5].[33:10]. Each area (or disk page) has one mass-address word. The mass-address word of area 0 is indexed by HDR[1].[19:8]. The mass-address word of area 1 is indexed by HDR[1].[19:8] + 1, and so on. The following is the format of a mass-address word:
  - Bit [43:1] indicates whether or not the area is allocated.
  - The disk segment address of the mass-address word is in bits [32:33].
  - Bit [42:1] is turned on if FAMILYINDEX was designated by the programmer at file creation.
- The ROWSIZE (the AREASIZE expressed in segments) is in HDR[5].[23:24].

- Bit HDR[7].[44:1] is turned on if the file is CRUNCHED.
- The end-of-file (in segments) is in HDR[9].[27:28]. If the end-of-file does not fall on a segment boundary, the count of bits used in the additional segment is in HDR[9].[47:20].
- The number of segments used in the last allocated area depends on whether or not the file is CRUNCHED. If the file is not CRUNCHED, the number of segments used in the last area is ROWSIZE. But if the file is CRUNCHED, the number of segments used in the last allocated area might be less than ROWSIZE. The following algorithm is used to determine the size of the last allocated area:

```
S := HDR[9].[27:28]                    %SEGMENTS
     + REAL(HDR[9].[47:20] NEQ 0);     %BITS IN LAST SEGMENT

A := CASE (HDR[2].[39:1]               %EXTMODE
          *HDR[2].[42.3])              %MODE
             OF (1, 1, 12, 8, 6, 6) * 30;
                        %SINGLE, DOUBLE, HEX, BCL, ASCII, EBCDIC

B := (HDR[3].[47:16] + A - 1) DIV A; %BLOCKS IN EXTMODE UNIT
```

- The number of segments occupied by a CRUNCHED file in its last allocated area is as follows:

```
(IF(S MOD B) = 0 THEN S ELSE (S - (S MOD B)) + B);
```

- After GETSTATUS has created a copy of the directory in a disk file, that disk file is locked as a CRUNCHED file with SECURITY = PRIVATE. The file should be processed with a FILETYPE of 8. Although MAXRECSIZE is 30 words, each in-use area might take more than 30 words, depending on the length specified in word 0 of the in-use area. Each in-use area begins on a segment boundary and uses an integer number of segments. The file is CRUNCHED.

SUBTYPE 9 and 10 Calls (Copying a Volume Directory)

SUBTYPEs 9 and 10 can be used to copy volume directory data records. You can use these SUBTYPEs only if the TAPECHECK = AUTOMATIC option of the system command SECOPT (Security Options) has been designated on the system.

Input

The following array input information is needed for SUBTYPEs 9 and 10:

ARY	Value or Description
ARY [0]	2
ARY [1]	0

continued

continued

ARY	Value or Description
ARY [2]	The beginning of the name of the file that the records from the volume directory are to be copied to. The name can be in display or standard form (depending on TYPE.DISPLAYFORMF). You can include a family name in the file title. The file that is created has a FILETYPE of 1. This file can be read by a logical file declaration that specifies FILETYPE = 1, DEPENDENTSPECS = TRUE, and INTMODE = SINGLE.

Results Returned

In the disk file that is created, the first word of each record is the count of words in the record. The words that follow this first word are an exact image of a volume directory data record. The format of volume directory records is documented in the *A Series Security Administration Guide*. The maximum record size of any record in the file is 2049 words.

If SUBTYPE = 9, all volume directory data records are copied. This form of the call can be used only by privileged users.

If SUBTYPE = 10, the volume directory data records that are copied are those whose FAMILYOWNER matches the usercode of the task. This form of the call can be used by privileged and nonprivileged users.

SUBTYPE 11 Calls (Reading an Archive Directory Record)

This call allows a nonprivileged process to retrieve a copy of an archive directory record by file title for any file owned by the usercode of the process. A privileged process can retrieve a copy of any selected archive directory record by file title (file name and family name).

Input

The following array input information is needed for SUBTYPE 11:

ARY	Value or Description
ARY [0]	2
ARY [1]	0
ARY [2] forward	Disk file TITLE in standard form (including family name)

Results Returned

This call returns the archive record for the file with the designated title from the archive directory for the disk family name included in the file title.

If the family name matches the target name of an active family substitution statement for the calling process, family substitution takes place.

If no usercode or asterisk (*) usercode is included in the title (the security byte = 1 or 5) and the process is running under a privileged usercode, GETSTATUS first searches under the process usercode. If GETSTATUS does not find a record with that title, GETSTATUS makes a second search under the asterisk (*) usercode directory.

GETSTATUS returns the record starting in word 2 of the array ARY and replacing the file title. The length in words of the archive directory record appears in bits [31:11] of the first word of the record (bits [31:11] of word 2 of the array ARY).

An archive record can be as long as 2047 words. Refer to the description of the format of archive directory records in the *A Series Disk Subsystem Administration and Operations Guide*.

GETSTATUS returns hard error 130 for the following reasons:

- The system cannot find an archive record for the named file.
- A nonprivileged process tries to retrieve an archive record for a file that does not belong to that user.
- The family is not online or does not have an active archive directory.

Example

The following example illustrates how to retrieve an archive record for a file titled MY/FILE ON WORKPACK.

```
  ARRAY ARY [0:2050]; BOOLEAN B;
ARY [0] := 2;
REPLACE POINTER (ARY [2]) BY
     48"140503", 48"02", 8"MY",
                 48"04", 8"FILE",
                 48"08", 8"WORKPACK";
B := GETSTATUS (3 & 11 [15:08], 0, 0, ARY);
```

SUBTYPE 12 and 13 Calls (Copying Records from an Archive Directory)

Use SUBTYPEs 12 and 13 to copy archive directory data records.

Input

The following array input information is needed for SUBTYPEs 12 and 13:

ARY	Value or Description
ARY [0]	6
ARY [1]	0 & 12 [32:17]

continued

continued

ARY	Value or Description
ARY [2] through ARY [5]	The family name (in substandard form) whose archive directory is to be copied.
ARY [6]	The beginning of the title of the file that the records from the archive directory are to be copied to. The title can be in display or standard form (depending on TYPE.DISPLAYFORMF). You can include a family name in the file title. The file that is created has a FILETYPE of 1. This file can be read by a logical file declaration that contains the following designations: • FILETYPE = 1 • DEPENDENTSPECS = TRUE • INTMODE = SINGLE

Results Returned

In the disk file that GETSTATUS creates, the first word of each record is the count of words in the record. The words that follow this first word are an exact image of an archive directory data record. The maximum record size of any record in the file is 2049 words. The format of archive records is documented in the *A Series Disk Subsystem Administration and Operations Guide.*

If SUBTYPE = 12, GETSTATUS copies all archive directory data records. This form of the call can be used only by privileged users, privileged programs, and processes started from the ODT.

If SUBTYPE = 13, GETSTATUS copies archive directory data records with file names that belong to the calling process (where ownership is determined by usercode). This form of the call can be used by privileged and nonprivileged users.

Examples

The following example shows how to copy archive records from the archive directory for the disk family named TESTPACK to a new file named MYARC/TESTPACK ON TESTPACK.

```
 ARY [Ø] := 6;
 ARY [1] := Ø & 12 [32:17]
% INPUT FAMILY NAME "TESTPACK"
 REPLACE POINTER (ARY [3]) BY 48"Ø8", 8"TESTPACK";
% OUTPUT FAMILY NAME "MYARC/TESTPACK ON TESTPACK"
 REPLACE POINTER (ARY [6]) BY 48"1BØ5Ø3"
           48"Ø5", 8"MYARC", 48"Ø8", 8"TESTPACK",
           48"Ø8", 8"TESTPACK";
 B := GETSTATUS (3 & 13 [15:Ø8], Ø, Ø, ARY);
```

The previous example can be continued with the following example that demonstrates how to read and process the records that GETSTATUS copied to the new file MYARC/TESTPACK ON TESTPACK. This example prints the names of the files

recorded in the file as well as the names and serial numbers of the library maintenance tapes on which the files have backup copies.

```
REAL BKNO, G, X;                %BKNO = Ø OR 1 (BACKUP NUMBER)
REAL SNS; POINTER PSNS;         %SERIAL NUMBERS
FILE PRT (KIND = PRINTER);
  ARRAY OUT [Ø:4ØØØ]; POINTER POUT;
FILE AIN (KIND = DISK, NEWFILE=FALSE, FILETYPE=1,
          DEPENDENTSPECS=TRUE, INTMODE=SINGLE,
          TITLE="MYARC/TESTPACK ON TESTPACK.");
ARRAY BUF [Ø:2Ø48];              %MAX RECORD SIZE IS 2Ø49 WORDS
  DEFINE ARBUF [X] = BUF [(X) + 1] #;  %BIAS OVER LENGTH WORD
  POINTER PIN;

% PARTIAL DEFINITION OF ARCHIVE RECORD STRUCTURE
DEFINE
 ARSRX = 3 #,
   ARSRF = [11:12] #,           %INDEX TO SUBRECORD

   ARSIZEF = [23:12] #,         %LINK TO FILE NAME
   ARLINKF = [11:12] #,         %INDEX OF CONTENT IN VARIABLE PART

 ARTITLEX = 6 #,                %LINK TO FILE NAME
 ARGBKSX = 9 #,                 %PAIR OF BACKUP INFO WORDS; AT
                                %  LEAST ONE BACKUP MUST EXIST OR
                                %  SUBRECORD (& RECORD) SHOULD BE
                                %  REMOVED FROM SYSTEM/ARCHIVE...
                                %  DIRECTORY.
   ARGTYPEF = [37:Ø4] #,        %TYPE OF BACKUP ENTRY
     ARGEMPTYV = Ø #,           %NO ENTRY OR DELETED ENTRY
 ARGNAMESX = 13 #,              %SETS OF FOUR 18-CHARACTER TAPE NAMES
 ARGSNSX   = 29 #;              %SETS OF FOUR LINKS TO SERIALNO LISTS

WHILE NOT READ (AIN, 2Ø49, BUF) DO
  BEGIN
   % PRINT FILE NAME
    REPLACE POUT := POINTER  (OUT) BY " " FOR 256;
    PIN := POINTER (ARBUF [ARBUF [ARTITLEX].ARLINKF]);
    STANDARDTODISPLAY (PIN, POUT);
    WRITE (PRT, 15, OUT);
    G := ARBUF [ARSRX].ARSRF;
```

```
% PRINT TAPE NAME AND SERIAL NUMBER LIST FOR ONE TO FOUR BACKUPS
 FOR BKNO := Ø STEP 1 UNTIL 3 DO
   IF ARBUF [G + ARGBKSX + BKNO].ARGTYPEF GTR ARGEMPTYV THEN
     BEGIN
       REPLACE POUT := POINTER (OUT) BY " " FOR 132;
       X := G + ARGNAMESX + 3*BKNO;
       REPLACE POUT: POUT + 5 BY POINTER (ARBUF [X]) + 1
            FOR ARBUF [X].[47:Ø8], ": ";
       X := G + ARGSNSX + BKNO;
       SNS := ARBUF [X].ARSIZEF;
       PSNS := POINTER (ARBUF [ARBUF [X].ARLINKF]);
      % PRINT AT MOST 6 SERIAL NUMBERS PER LINE
       THRU (SNS + 5) DIV 6 DO
         BEGIN
           THRU (MIN (SNS, 6)) DO
             BEGIN
               SNS  := *-1;
               REPLACE POUT:POUT BY PSNS:PSNS FOR 6, ", ";
             END;
           WRITE (PRT, 15, OUT);
           REPLACE POUT := (POINTER (OUT) BY " " FOR 132;
           POUT := POUT + 24;
         END;
       END;    % OF DOUBLE LOOP OVER BACKUP SLOTS
END;           % END OF READ FILE LOOP
```

SUBTYPE 14 Calls (Open File Information)

Use SUBTYPE 14 to implement the system command SHOWOPEN. Each request returns information about one open disk file. The file can be a permanent disk file (that is, recorded in the system directory) or a temporary disk file. Each response updates A[1] with the header index of the file that was found. This value should be left in the array A so that it can be used by GETSTATUS as a starting point to resume the search on the next call. If you set A[1] to 0, the next GETSTATUS call will restart the search through the headers.

Input

The following array input information is needed for SUBTYPE 14:

ARY	Value or Description
ARY [0]	8
ARY [1]	0 or the disk file header index returned by a preceding call. For the first call, A[1] should contain 0. For all subsequent calls, GETSTATUS will automatically use the value in A[1] to locate the next open disk file.
ARY [2].[7:8]	0 or the family index. If the value is 0, the report is for all open files on all disks in the family. If the value is greater than 0, the report is only for open disk files on that family index of the family.

continued

continued

ARY	Value or Description
ARY [3]	If the report of open files is to be limited to open files in an address range of a particular member of the family, A[3] should contain the starting sector address of that range.
ARY [4]	0 if the report of open files is not to be limited to an address range. If the report is to be limited to open files in an address range of a particular member of the family, A[4] should be the number of sectors in the address range (that is greater than 0) and A[2].[7:8] must be greater than 0.
ARY [5]	Family name in substandard form.

Results Returned

The information to be returned is specified in the MASK parameter (refer to Table 4–1). GETSTATUS returns either a soft error word, a file status word, or a validity mask word and fixed information as requested by the MASK parameter. For a description of the format of this information, refer to the appropriate areas discussed in "General Format of Results from Directory Calls (SUBTYPEs 0, 1, 2, and 4)" earlier in this chapter.

If an error is found in the parameters, if there is a security error or if the disk family is not online, GETSTATUS returns a soft error word in word 1 of the array ARY. If there are no (more) open files on the family (in the selected address range, if any), GETSTATUS returns soft error word 239 (NO MORE OPEN FILES).

If GETSTATUS does not detect any errors, and if you do not specify any MASK bits in the call or if you specify only MASK bit [0:1], GETSTATUS does not return any information about the file. But GETSTATUS does put the disk file header index into word 1 of the array.

If GETSTATUS does not detect an error, and if you specify MASK bits other than [0:1] in the call, GETSTATUS returns a validity mask word in word 8 of the array. This is followed by a file status word in word 9 and information for MASK bits 1 through 47 in word 10 through 56. These words in turn point to information, such as the file title, that is returned elsewhere in the array. Refer to Table 4–1 and Figure 4–1. GETSTATUS does not return any catalog or archive information for SUBTYPE 14 calls.

If there are no errors, GETSTATUS stores a disk file header index in Word 1 of the array ARY for use in a subsequent call for the next file.

Section 5
GETSTATUS Request Type 4 (Disk and Tape Volumes)

These calls allow you to retrieve information from the volume library or volume directory. For methods of obtaining complete copies of volume library or volume directory records, refer to "SUBTYPE 5 Calls (Copying a Volume Library)" and "SUBTYPE 9 and 10 Calls (Copying a Volume Directory)" in Section 4 of this manual.

SUBTYPE 0 Calls (Volume Library Information)

This call allows any process to retrieve information about any tape or disk volume listed in the volume library. The data retrieved includes information about the volume specified by serial number plus the serial numbers and status of other volumes in the same family. Tape volumes are in the same family if a file or files were written to them with intervening reel switches. This call corresponds to the system command PV (Print Volume). It can be used only if OP + CATALOGING has been specified on the system.

Input

The array ARY must be at least 23 words long and should be at least *21 + 2 times the number of volumes in the family* words long.

ARY	Value or Description
ARY [0]	4
ARY [1]	Serial number (6 EBCDIC characters)
ARY [2]	1 if the volume is a disk
	15 if the volume is a tape
	17 if the volume is a disk pack
ARY [3]	0

Call

```
B := GETSTATUS (4 & Ø [15:Ø8], Ø, Ø, ARY);
```

Results Returned

The information returned for SUBTYPE 0 calls is as follows:

Word Number	Description
Word 1	If bit [47:01] is turned on, ARY [1] contains a soft error word. Otherwise, ARY [1] contains a descriptor code for the requested volume as follows: • If bit [23:01] is turned on, the volume is a scratch tape. • If bit [17:01] is turned on, a WFL *VOLUME DESTROYED* statement has been executed for the volume. • If bit [16:01] is turned on, the volume is the base volume of the family.
Word 2	Count of volumes reported (starting in ARY [20]).
Word 4	Family name in substandard form.
Word 5	Continuation of family name in substandard form.
Word 6	Continuation of family name in substandard form.
Word 9	System serial number of the family creation site or 0.
Word 10	SAVEFACTOR or 0.
Word 11	Binary family creation date or 0.
Word 15	Count of volumes in the family.
Words 20 through 20 + 2 * ARY [2]	Each has two words for each volume in the family: • First word: SERIALNO value of volume (6 EBCDIC characters). • Second word: Bit codes as follows: – If bit [23:01] is turned on, the volume is a scratch tape. – If bit [17:01] is turned on, a WFL *VOLUME DESTROYED* statement has been executed for the volume. – If bit [16:01] is turned on, the volume is the base volume of the family.

Soft error word 126 is returned in word 1 if the serial number is error word 126 is returned in Word 1 if the serial number is not found in the volume library.

SUBTYPE 1 Calls (Volume Directory Record)

This call allows a privileged process to retrieve a copy of any selected volume directory record by serial number. This call corresponds to the system command TV (Type Volume). It can be used only if the TAPECHECK = AUTOMATIC option of the SECOPT (Security Options) system command has been specified on the system.

Input

The following array input is needed for SUBTYPE 1 calls:

ARRAY	Value or Description
ARY [0]	4
ARY [1]	Serial number of the volume whose volume directory record is being retrieved (6 EBCDIC characters)
ARY [2]	15 (tape)

Call

```
B := GETSTATUS (4 & 1 [15:Ø8], Ø, Ø, ARY);
```

Results Returned

The requested volume directory record is returned starting in word 4 of ARY. Refer to the *Security Administration Guide* for the format of volume directory records. Errors are returned in the normal manner. If a soft error occurs, GETSTATUS returns a Boolean value of TRUE with bits [11:08] of the Boolean result equal to 0, and GETSTATUS stores the soft error word in ARY [1].

Section 6
GETSTATUS Request Type 5 (Unit Requests)

GETSTATUS unit requests are grouped under Request Type 5. This GETSTATUS call retrieves information about peripheral units as well as about the volumes and sometimes the files on these units. The controller uses these GETSTATUS calls to retrieve information for such displays as those generated by the PER (Peripheral Status) and OL (Display Label and Paths) system commands.

For most GETSTATUS unit requests, you need to understand the 6-bit UNITTYPE codes. (See Appendix C for a list of unit type codes.) UNITTYPE codes often correspond to the values of the file attribute KIND. Some of the information returned for GETSTATUS unit requests uses the same numerical codes as information returned by SYSTEMSTATUS peripheral requests. It is necessary to refer to the explanations in the *A Series SYSTEMSTATUS Programming Reference Manual* to understand those values returned by GETSTATUS.

To use unit calls, the calling process must have privileged status or the call must be started from an ODT with a WFL *JOB* or *RUN* statement, but not with a primitive *??RUN* system command. Otherwise, a security error occurs and GETSTATUS returns hard error 43 (refer to Appendix A).

The general form of the GETSTATUS call is as follows:

```
B := GETSTATUS (TYPE, SUBCLASS, MASK, ARY);
```

Parameters

The parameters for Request Type 5 calls are described in the following pages.

TYPE

This parameter generally selects the specific case and subcase within the GETSTATUS intrinsic. For unit requests, the fields within the word are as follows:

- TYPE.TYPEF [7:08] = 5
- TYPE.SUBTYPEF [15:08]

TYPE.SUBTYPEF [15:08] can have one of the following values:

SUBTYPE	Description
0	Retrieves information for a specific list of one or more units by unit number. SUBTYPE 0 can be used for all kinds of units, including disks and packs.
1	Retrieves information for the units within a designated range of unit numbers.
2	Retrieves information for all units of designated kinds that are assigned to a designated process.
3	Reserved.
4	Not valid.
5	Retrieves information for a designated list (by unit number) of port and control units. The information to be retrieved is designated by parameter MASK bits (refer to Table 6–2).
20	Retrieves information for a specific list of disks and packs. SUBTYPE 20 retrieves some extra information that a SUBTYPE 0 call does not supply.
21	Retrieves information for disk units within a designated range of unit numbers. SUBTYPE 21 retrieves some extra information that a SUBTYPE 1 call does not supply.

SUBCLASS

If SUBCLASS is negative (the sign bit is turned on), GETSTATUS can report on the units assigned to invisible independent runners; if the sign bit is turned off, GETSTATUS does not report them.

MASK

For SUBTYPEs 0, 1, 2, 5, 20, and 21, the MASK parameter is used to select the unit attributes and status information to be retrieved by GETSTATUS. Each bit that is turned on in the MASK parameter instructs GETSTATUS to report the corresponding unit, volume, or file information (refer to Tables 6–1 and 6–2 for a listing of valid MASK bits and their uses).

Array ARY Parameter

The caller should place information such as unit numbers and so forth in the array ARY to indicate to GETSTATUS the units that are to be reported. In turn, GETSTATUS stores the retrieved status and attribute information into the array ARY. The exact format of the array parameter and result information is described later under the description of each particular SUBTYPE.

Request Type 5 Calls by SUBTYPE

The following Request Type 5 calls are described by SUBTYPE for the sake of functionality and common characteristics.

General Input Information

Some values reported in Table 6–1 correspond exactly to file attributes. Other values are related to units and file status. These latter values are mainly for internal use by the MCP and might change in future releases.

In the information tables that follow (Tables 6–1 and 6–2) , the link format for unit requests is as follows:

Field	Description
GSINFOF [15:16]	This value is the length in bytes of the item linked.
XSLINKF [32:17]	This value is the word offset to the beginning information in the array ARY. When this offset is added to the base, the offset gives the index of the information. Refer to Figure 6–1.

Many of these links point to a name that is in standard form. For a description of standard form, refer to the DISPLAYTOSTANDARD function in the *DCALGOL Reference Manual*.

Table 6–1. Status Information Reported for SUBTYPE 0, 1, 2, 20, and 21 Calls

MASK Bit	Description
0	Unit number.
1	Contents of MCP UNIT word. Bits [47:06] contain the UNITTYPE code (refer to the *SYSTEMSTATUS Reference Manual*). The UNITTYPE code is related to the file KIND attribute. For ODT units, bit [22:01] is turned on if the ODT is restricted.
2	Has the value 1 if the unit is ready or is assigned to a process. Refer to MASK bit 5.
3	Reliability factor. Refer to the system command RF (Reliability Factor).
4	UNITSTATISTICS word. Refer to the *SYSTEMSTATUS Reference Manual* for details.
5	If the unit is assigned to a single process, this bit contains the stack number of that process.
6	Disk or magnetic tape SERIALNO value in 6 EBCDIC characters.
7	For magnetic tape units, the FILESECTION value of the file at which the tape is positioned. For disk units, the FAMILYINDEX value of the volume.

continued

Table 6–1. Status Information Reported for SUBTYPE 0, 1, 2, 20, and 21 Calls (cont.)

MASK Bit	Description
8	For magnetic tape units, the SAVEFACTOR value of the file at which the tape is positioned. For disk units in SUBTYPE 20 or 21 calls, the following values apply: • Bit [0:01] is turned on if the unit is saved. • Bit [1:01] is turned on if the unit does not have a labeled volume. • Bit [2:01] is turned on if the unit is assigned to a process. • Bit [3:01] is turned on if an error occurred while the system was readying the disk. • Bits [45:03] contain the COMS COACTIVE transition state. • Bits [47:02] = 0 if the disk is not part of a family that has been designated for use by COMS COACTIVE files. • Bits [47:02] = 1 if the disk is in the Reader state and has been designated for use by COMS COACTIVE files. • Bits [47:02] = 2 if the disk is in the Writer state and has been designated for use by COMS COACTIVE files.
9	For magnetic tape, the creation date of the file at which the tape is positioned. For disks, the date that the original base pack of the family was reconfigured in binary. When printed in decimal, the date is in the form YYDDD, where YY is the year and DDD is from 1 to 366. For CD-ROM, the date that the CD-ROM was created.
10	For disks, a link to the FAMILYNAME in standard form. For external I/O units, link to stacks. For CD-ROM units, link to CD name in standard form. For other units, link to filename in standard form.
11	For magnetic tape, the numerical code for the DENSITY. For disk units in SUBTYPE 20 or 21 calls, a link to the host name of the writer system in standard form if the disk is part of a family that has been designated for use by COMS COACTIVE files.
12	For magnetic tape volumes, the LABELKIND value. Bit [47:01] is turned on if the volume is a library maintenance tape that contains files with license keys. For CD-ROM, bits [47:01] are turned on if the CD-ROM is in library maintenance format and contains files with license keys. Bits [46:01] are on if the CD-ROM is in library maintenance format.

continued

Table 6–1. Status Information Reported for SUBTYPE 0, 1, 2, 20, and 21 Calls (cont.)

MASK Bit	Description
13	EXTMODE attribute of the file on the unit.
14	For disk units, the following values apply: • Bit [0:01] is turned on if the volumes has been purged. • Bit [1:01] is turned on if the volume needs to be initialized, verified, and reconfigured. • Bit [2:01] is turned on if an error has occurred while the system was readying the unit (or if the system has not finished readying the unit). • Bit [3:01] is turned on if the volume SERIALNO value matches the serial number of another disk on the system. • Bit [4:01] is turned on if the disk unit has been closed. • Bit [5:01] is reserved. • Bit [6:01] is turned on if all I/Os to the unit are canceled. • Bit [7:01] is turned on if the unit is saved. • Bit [8:01] is turned on if the unit is reserved for maintenance. • Bit [9:01] is turned on if an error occurred while the system was readying the unit. • Bit [10:01] is turned on if the unit is saved. • Bit [11:01] is turned on if the volume is a member of the halt/load family. • Bit [12:01] is turned on if the system has read label information from the volume. • Bit [13:01] is turned on if the unit is part of a mirrored set. • Bit [14:01] is turned on if the unit is one on which the cache feature is being used. • Bits [47:04] contain one of the following codes: – 0 if the volume is an INTERCHANGE volume. – 1 if the volume is the basepack. – 2 if the volume is a continuation pack. – 3 if the volume label has not been (or could not be) read by the system.
15	If the unit is assigned to a visible process, a link to the NAME of that process in standard form.

continued

Table 6–1. Status Information Reported for SUBTYPE 0, 1, 2, 20, and 21 Calls (cont.)

MASK Bit	Description
16	For disk packs, 8 plus the unit subtype code that the system uses to distinguish between various kinds of disk packs. Refer to the *SYSTEMSTATUS Reference Manual* for a list of the code values.
17	For disk units, the capacity of each storage unit in sectors.
18	For disk units, the number of lock-out switches per storage unit.
19	For memory disks, the bit mask of switches locked out.
20	For disk packs, 1 if the family member is an IAD pack. For memory disks, the bit mask of IAD switches.
21	For mirrored disk units, a code for the mirror recovery option of the unit: • 0 = Discard • 1 = DMS Refer to the system command *MIRROR OPTION PK<unit no.>*
22	For disks, the volume SERIALNO value in binary of the original base pack of the family.
23	A link to the FORMID value of the unit or of the file on the unit in standard form.
24	For disks and CD-ROMs, a count of the number of open files and users of the disk or the number –1. To calculate this count, GETSTATUS must procure various directory locks. GETSTATUS must make present and examine the headers for all open disk files at the system. If the elapsed time of the GETSTATUS call exceeds 5 seconds, GETSTATUS skips the counting procedure and reports –1 instead. Refer also to MASK bit 32.
25	Link to the peripheral association list (refer to the system command PA). The first word of the list contains the count of entries. The entries begin in the second word, and each entry is two words long. The first word is the folded UNITTYPE code, and the second word is the unit number.
26	The count of the number of units associated with this unit. Refer to the system command PA (Peripheral Association).
27	File CYCLE attribute.
28	File VERSION attribute.

continued

Table 6–1. Status Information Reported for SUBTYPE 0, 1, 2, 20, and 21 Calls (cont.)

MASK Bit	Description
29	For data comm units (UNITTYPE = 16): • Bit [47:01] is turned on if bits [40:05] and [35:04] are valid: – Bits [40:05] contain the relative DCP number. – Bits [35:04] contain the relative cluster number. • Bits [31:08] contain the programmable read-only memory (PROM) level. • Bits [23:08] contain the PROM patch. • Bits [15:08] contain the firmware level. • Bits [7:08] contain the firmware patch.
30	For printer units: • Train ID of printer. • For disk units in SUBTYPE 20 or 21 calls, the sector size in bytes.
31	For HYPERchannel units: • Bits [47:08] contain the READPARTNER address. • Bits [39:08] contain the READPARTNER log address. • Bits [31:08] contain the WRITEPARTNER address. • Bits [23:08] contain the WRITEPARTNER log address. • Bit [14:01] is turned on if the unit is reserved. • Bit [13:01] is turned on if HYFILES partners have restrictions. • Bit [12:01] is turned on if the unit is for BNA. • Bits [11:04] contain the bit MASK of available trunks. Bit 11 is trunk 0, and so forth. • Bits [7:08] contain the adapter number. For disk units in SUBTYPE 20 or 21 calls, the total capacity of the unit in sectors.
32	For disks and CD-ROMs, either a count of the number of open files and users of the disk or the number –1. To calculate this count, GETSTATUS must procure various directory locks. GETSTATUS must make present and examine the headers for all open disk files at the system. If the elapsed time of the GETSTATUS call exceeds 60 seconds, GETSTATUS skips the counting procedure and reports –1 instead. Note the similarity with MASK bit 24.

continued

Table 6–1. Status Information Reported for SUBTYPE 0, 1, 2, 20, and 21 Calls (cont.)

MASK Bit	Description
33	For labeled tape and disk volumes, the system serial number in binary of the site that created the volume.
34	For magnetic tape, the creation date of the file at which the tape is positioned. For disks, the creation date of the original base pack of the family in binary. When printed in decimal, the date is in the form YYDDD, where YY is the year and DDD is from 1 to 366.
35	Creation time of the original base pack of the family in 2.4-microsecond intervals of time, or creation time of the CD-ROM.
36	For host control (HC) units: • Bits [36:05] contain 0 or the READPARTNER number plus 1. • Bits [28:05] contain 0 or the WRITEPARTNER number plus 1. • Bit [23:01] is turned on if there are restrictions on the HCFILES HUBNUMBER value. • Bit [22:01] is turned on if the unit is for BNA. • Bits [12:02] contain the mode code: – 0 = CLOSED – 1 = IN – 2 = OUT – 3 = IO • Bits [19:04] contain the HUBNUMBER value. • Bits [15:16] contain the HUBINDEX value.
37	For magnetic tapes: • Bit [47:01] is turned on if the BLOCKSIZE value of the file exceeds the maximum allowed on the unit. • Bits [43:20] contain the maximum possible BLOCKSIZE value in bytes. • Bits [19:20] contain the BLOCKSIZE value of the file in bytes. • For disk units in SUBTYPE 20 or 21 calls, bit [1:02] determines whether the M9710 disk drive is single- or dual-ported. If bit [1:02] = 0, the drive is single-ported. If bit [1:02] = 1 or 2, the drive is dual-ported.
38	0 if no ready paths exist to the unit.
39	Internal code for the type of DLP used by the unit.

continued

Table 6–1. Status Information Reported for SUBTYPE 0, 1, 2, 20, and 21 Calls (cont.)

MASK Bit	Description
40	Contains 1 if the unit is available, and 0 if the unit has been freed.
41	Index of path information for the unit. Refer to "I/O Path Information" later in this section.
42	Unit SUBTYPE value. Information on unit SUBTYPE values is located in the *SYSTEMSTATUS Reference Manual*.
43	1 if the unit is suspended.
44	Internal code for the type of DLP used by the unit or the number 10.
45	0 or the unit number of the base pack of the family.
46	For external input/output units (EIOUs), link to a copy of the unit information (UINFO) table information for the unit.
47	Link to mirror information. The mirrored disk information has the following format: • Word 0: – Bits [23:08] contain the number of pending mirrors. – Bits [15:08] contain the number of offline mirrors. – Bits [7:08] contain *m*, the number of online mirrors. • Words 1 through *m* contain unit numbers of online mirrors. • Words (m + 1) through *n* contain the unit numbers of pending mirrors.

I/O Path Information

The first word of the information format contains general information. Word 1 has the following format:

Field	Name	Description
[47:04]	SYSTEMTYPEF	Contains 2 for Entry and Medium Systems (EMS) and 4 for HDU systems.
[43:08]	DLPTYPEF	Contains the type of DLP.
[35:04]	RELUNITF	Contains the relative number of the unit with respect to the DLP.
[18:01]	UNITFREEDF	Contains 1 if the unit has been freed, but 0 if the unit has not been freed.

continued

continued

Field	Name	Description
[17:01]	SOMEPATHOUTBOARDF	Contains 1 if at least one path is through an outboard host and 0 if no paths are through an outboard host.
[16:01]	SOMEPATHHASFIRMWAREF	Contains 1 if at least one path has firmware and 0 if no paths have firmware.
[15:08]	UNITTYPEF	Contains the type of unit outboard of the DLP.
[7:08]	NUMPATHSF	Contains the number of existing paths to the unit.

The second and subsequent words contain information specific to each path, six words per path.

Word 1 of each path entry contains status information in the following format:

Field	Name	Description
[47:01]	PATHOFFLINEF	Contains 1 if the path is offline and 0 if it is not offline.
[45:01]	PATHRESERVEDF	Contains 1 if the path is reserved and 0 if the path is not reserved.
[44:01]	PATHRESERVEDMAINTF	Contains 1 if the path is reserved for maintenance and 0 if the path is not reserved for maintenance.
[43:01]	PATHBROKENF	Is not applicable for EMS. For host data units (HDUs), the field contains 1 if the path is broken and 0 if it is not.
[42:01]	PATHHASSIGNEDF	Contains 1 if maintenance is running on the path and 0 if maintenance is not running on the path.
[41:01]	PATHOUTBOARDF	Contains 1 if the path is through an outboard host and 0 if the path is not through an outboard host.
[40:01]	PATHHASFIRMWAREF	Contains 1 if the path has firmware and 0 if it does not.
[39:01]	PATHINUSEF	Contains 1 if the path is currently selected for all I/O operations to the unit and 0 if it is not.
[38:01]	PATHREADYF	Contains 1 if the host can see the unit ready through this path and 0 if it cannot.

continued

continued

Field	Name	Description
[37:01]	DLPFREEDF	Is not applicable for EMS. For HDUs, the field contains 1 if the DLP of the path has been freed and 0 if it has not.

Word 2 of each path entry contains the path name. Word 2 is formatted as follows:

Field	Name	Description
[47:01]	PATHNAMEF	Contains the path identifier in EMS and the DLP identifier in HDU systems.

Word 3 of each path entry contains firmware information. Word 3 is formatted as follows:

Field	Name	Description
[43:04]	PATHFWDIGITSF	Contains the number of digits in the firmware identifier.
[39:40	PATHFIRMWAREF	Contains the firmware identifier.

Word 4 of each path entry contains physical path information (machine dependent). Word 4 is formatted as follows:

Field	Name	Description
[23:04]	IOPF	Contains the I/O processor number.
[19:04]	PATHOUTBOARDF	Contains the host-dependent port (HDP) number for EMS. If the PATHOUTBOARDF value is 1, this field is 0 (meaningless). For HDUs, this field is not applicable.
[19:02]	MLIF	Is not applicable for EMS. For HDUs, it contains the port number.
[17:02]	HDPF	Is not applicable for EMS. For HDUs, it contains the HDP number.
[7:04]	LEMPORTF	Contains the address of the base on the line expansion module (LEM).
[3:04]	DLPADDRF	Contains the address of the DLP in the base.

Word 5 of each path entry contains the name of the base in which the DLP resides. Word 5 is formatted as follows:

Field	Name	Description
[47:48]	BASENAMEF	Is not applicable for EMS and contains the base name for HDUs.
[15:08]	BASEMAINTBUSADDRF	Contains the maintenance bus number of the base. This field is not applicable for HDUs.
[7:04]	BASEMAINTBUSNUMF	Contains the maintenance bus number of the bus. This field is not applicable for HDUs.
[3:04]	BASEEXTUNITNUMF	Contains the extension unit number of the base. This field is not applicable for HDUs.

Word 6 of each path entry contains the outboard host name if the path is through an outboard host. Word 6 is formatted as follows:

Field	Name	Description
[47:48]	OUTBOARDF	Contains the external unit name of the outboard host.

SUBTYPE 5 retrieves information by device number for a designated list of ports, controllers, and DCPs. For SUBTYPE 5, the MASK parameter selects information as described in Table 6–2.

Table 6–2. Information Returned for SUBTYPE 5 Calls

MASK Bit	Description
0	Index of path information for the unit. Refer to the information in "I/O Path Information" in this section.
1	Contents of MCP UNIT word. Bits [47:06] contain the UNITTYPE code (refer to the *SYSTEMSTATUS Reference Manual*). The UNITTYPE code is related to the file KIND attribute. For ODT units, bit [22:01] is turned on if the ODT is restricted.
2	Contains 1 if the unit is ready or is assigned to a process. Refer to MASK bit 3.
3	If the unit is assigned to a single process, the stack number of that process.

continued

Table 6–2. Information Returned for SUBTYPE 5 Calls (cont.)

MASK Bit	Description
4	For HYPERchannel units: • Bits [47:08] contain the READPARTNER address. • Bits [39:08] contain the READPARTNER log address. • Bits [31:08] contain the WRITEPARTNER address. • Bits [23:08] contain the WRITEPARTNER log address. • Bit [14:01] is turned on if the unit is reserved. • Bit [13:01] is turned on if HYFILES partners have restrictions. • Bit [12:01] is turned on if the unit is for BNA. • Bits [11:04] contain the bit MASK of available trunks. Bit 11 is trunk 0, and so forth. • Bits [7:08] contain the adapter number.
5	For host control (HC) units: • Bits [36:05] contain 0 or the READPARTNER number plus 1. • Bits [28:05] contain 0 or the WRITEPARTNER number plus 1. • Bit [23:01] is turned on if there are restrictions on the HCFILES HUBNUMBER value. • Bit [22:01] is turned on if the unit is for BNA. • Bits [12:02] contain the mode code: – 0 = CLOSED – 1 = IN – 2 = OUT – 3 = IO • Bits [19:04] contain the HUBNUMBER value. • Bits [15:16] contain the HUBINDEX value.

General Results Returned for SUBTYPE 0, 1, 2, 5, 20, and 21 Calls

GETSTATUS returns one of two kinds of result words in the array ARY for each unit that is reported.

Soft Error Word

If the GSERRORF bit ([47:01]) of the word is turned on, an error prevented GETSTATUS from reporting on the unit. This bit distinguishes a soft error word from a

unit info pointer word. The GSERRORVALUEF field ([46:08]) contains the soft error code (refer to Appendix B).

Unit Info Pointer Word

A unit info pointer word points to the information for the unit, file, or volume mounted on the unit and contains a code for the unit kind that identifies the unit, file, or volume mounted on the unit. Unit info pointer words are structured as follows:

- GSERRORF [47:01]

  The bit is OFF.

- GSVALUEF [38:06]

  This field contains the value 1.

- XSLINKF [32:17]

  If this field is not 0, the value is the index or base of the validity mask, and the attribute list returned for the unit follows the validity mask in the array ARY (refer to Figure 6–1). This field will not be 0 if the MASK parameter has any bits other than bit [0:01] turned on.

- GSUTYPEF [15:08]

  This field contains the code for the UNITTYPE or KIND of unit. (See Appendix C for a list of unit type codes.)

Figure 6–1 is a diagram of the information pointed to by the unit info pointer word. BASE is the value contained in the XSLINKF field of the unit info pointer word.

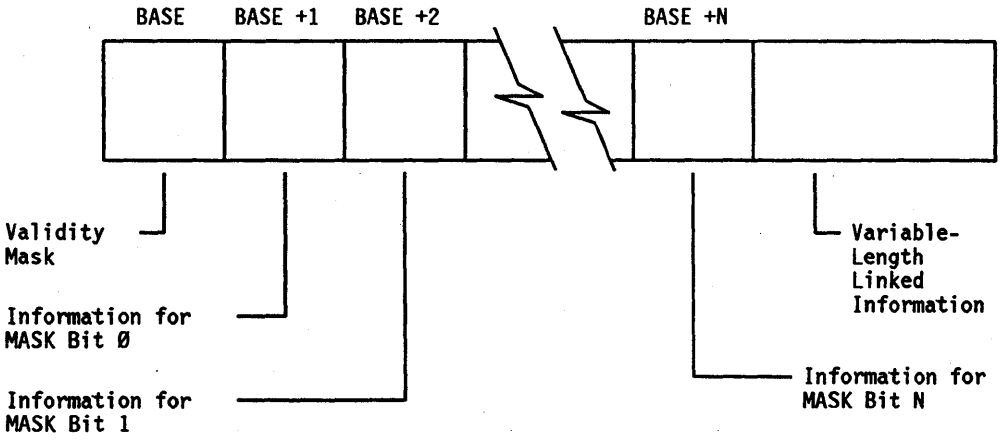

Figure 6–1. Information Pointed to by the Unit Info Pointer Word

A bit that is turned on in the validity mask corresponds to bits in the MASK parameter that are turned on for information that was successfully retrieved.

Each word after the validity mask for which the corresponding validity mask bit is turned on is either the file attribute or status information retrieved or is a link to the file attribute or status information retrieved.

The variable-length linked information follows the last word referenced by the validity mask. GETSTATUS places information in this area for items described in the MASK table (Tables 6–1 and 6–2) with the phrase *link to*. Given the value of the link word, the caller can locate the linked information with an index calculated with one of the following formulas:

```
INDEX := ARY [BASE + (MASK bit number + 1)].XSLINKF + BASE;

INDEX := (link word).XSLINKF + BASE;
```

XSLINKF = [32:17] and INDEX is the word index that locates the beginning of the information. For example, the link word for a disk pack family name (MASK bit 10) is contained in ARY [BASE + 11].

SUBTYPE 0 and 20 Calls (List by Unit Number)

SUBTYPEs 0 and 20 retrieve information for a specific list of one or more units by unit number. SUBTYPE 0 can be used for all kinds of units, including disks and packs. SUBTYPE 20 can be used only for disks and packs, but a SUBTYPE 20 call retrieves some extra information for disks that a SUBTYPE 0 call does not include.

Input

For both SUBTYPEs 0 and 20, the caller should store in ARY [0] the index plus 1 of the last unit number designated. Beginning in word ARY [1], the caller should store two words in the array ARY for each unit for which information is to be retrieved. Each pair of words should be as follows:

- First word

  [38:06] should contain a 1.

  [15:08] should contain the UNITTYPE code for the unit. (See Appendix C for a list of unit type codes.)

- Second word, ADDLWORD(1)

  This word contains the unit number.

Results Returned

GETSTATUS replaces the first word of each UNITTYPE/unit number pair with either a soft error word (with bit [47:01] turned on) or a unit info pointer word. If the array does not have enough room for GETSTATUS to return all the requested information, GETSTATUS either returns a hard error Boolean result (not enough room to report the first requested unit) or turns on bit [47:01] of ARY [0] and stores in ARY [0].[19:20] the index plus 1 of the last UNITTYPE/unit number pair that was processed.

SUBTYPE 1 and 21 Calls (Units within a Designated Range of Numbers)

SUBTYPEs 1 and 21 retrieve information for the units within a designated range of unit numbers. SUBTYPE 1 can be used for all kinds of units, including disks and packs. SUBTYPE 21 can be used only for disks and packs, but a SUBTYPE 21 call retrieves some extra information that a SUBTYPE 1 call does not include.

Input

SUBTYPEs 1 and 21 make use of the SUBCLASS parameter. If SUBCLASS is plus or minus 0, a unit is not reported unless it is either ready, assigned to a process, or rewinding. If SUBCLASS is plus or minus 1, units are reported even if they are not ready or in use.

For SUBTYPEs 1 and 21, the caller should set up the array ARY as follows:

- ARY [0]

  This word indicates how many units can be reported. ARY [0] should contain the index plus 1 of the last word in the array into which GETSTATUS can store a unit number. But note that for each unit reported, GETSTATUS uses two words.

- ARY [2]

  This word should contain the beginning unit number minus 1.

- ARY [3]

  This word should contain the last unit number plus 1.

- ARY [4]

  This field should contain a unit kind selector mask. A unit is not reported by GETSTATUS unless the bit that corresponds to the UNITTYPE of the unit is ON in the selector mask in ARY [4]. The correspondence of selector mask bits to UNITTYPE is as follows:

  Bit 0 corresponds to UNITTYPE 1 (memory disk), bit 1 corresponds to UNITTYPE 2 (ODT or SPO), and so on. UNITTYPE 0 is never reported. (See Appendix C for a list of unit type codes.)

Results Returned

Beginning at ARY [1] up to ARY [ARY [0] –1], GETSTATUS stores two words for each unit selected, according to the SUBCLASS, the beginning and ending unit numbers, and the folded UNITTYPE mask. The first word is either a soft error word or a unit info pointer word. The second word contains the unit number. Each unit info pointer word points to the information requested by the MASK parameter. GETSTATUS stores that information in the array ARY beginning at the word indexed by ARY [0]. If there are not enough words available in the array to report any units or if ARY [0] is less than 3, a hard error is returned. Otherwise, GETSTATUS changes the index contained in ARY [0] to be the index plus 1 of the word containing the last unit number processed. If there were not enough words in the array to report all the information available for all the requested units or if there were not enough words reserved between ARY [1] and

ARY [ARY [0] –1] to store all the requested unit info pointer words and unit numbers, GETSTATUS turns on bit [47:01] in ARY [0].

SUBTYPE 2 Calls (All Units Assigned to a Designated Process)

SUBTYPE 2 retrieves information for all units of designated kinds that are assigned to a designated process.

Input

SUBTYPE 2 reports on all units of designated kinds that are assigned to a designated stack number. Notice that some units, such as disk units, are not assigned to a stack even though a process stack might have files open on the unit. Input for SUBTYPE 2 is as follows:

- ARY [0]

  The value in this field indicates the number of units to report. The field should contain the index plus 1 of the last word in the array into which GETSTATUS can store unit info pointer words.

- ARY [1]

  Bits [15:16] should contain the stack number.

  Bits [40:25] should contain a unit kind selection mask. A unit is not reported by GETSTATUS unless the bit that corresponds to the folded UNITTYPE of the unit is ON in the selector mask in ARY [1]. The correspondence of selector mask bits to UNITTYPE is as follows: Bit 0 of the selector mask (bit 16 of ARY [1]) corresponds to UNITTYPEs 1 and 33, bit 1 of the selector mask (bit 17 of ARY [1]) corresponds to UNITTYPEs 2 and 34, and so on. UNITTYPEs 0, 26 through 32, and 48 through 63 are never reported. (See Appendix C for a list of unit type codes.)

Results Returned

GETSTATUS returns results as follows for SUBTYPE 2:

- If a hard error is detected, GETSTATUS returns a Boolean value of TRUE with the nonzero hard error code in bits [11:08] of the Boolean result.
- If the stack number is not valid, a soft error word is stored in ARY [1].

Otherwise, unit info pointer words, soft error words, or both are stored beginning in ARY [1] and continuing up to (at most) ARY [ARY [0] –1]. The unit info pointer words point to the information requested by the MASK parameter bits. This information is stored in the array beginning at word ARY [ARY [0]]. If there are not enough words available to return all the information for all the designated units, GETSTATUS turns on bit [47:01] of ARY [0]. Before returning, GETSTATUS stores in ARY [0].[19:20] the index plus 1 of the last word in which a unit info pointer word has been stored.

SUBTYPE 5 Calls (Specific List of Ports or Controls)

SUBTYPE 5 retrieves information for a specific list of one or more port or control units by unit number. This call corresponds to the *OL CTL <device number list>* and *OL PORT <port number list>* system commands.

Input

For SUBTYPE 5, the caller should store in ARY [0] the index plus 1 of the last unit number designated. Beginning in word ARY [1], the caller should store two words in the array ARY for each unit for which information is to be retrieved. Each pair of words should be as follows:

- First word

  [38:06] should contain a 1.

  [15:08] should contains the UNITTYPE code for the unit. (See Appendix C for a list of unit type codes.)

- Second word, ADDLWORD(1)

  This word contains the unit number.

Results Returned

GETSTATUS replaces the first word of each UNITTYPE/unit number pair with either a soft error word (with bit [47:01] turned on) or a unit info pointer word. If there is not enough room in the array for GETSTATUS to return all the requested information, GETSTATUS either returns a hard error Boolean result (not enough room to report the first requested unit) or turns on bit [47:01] of ARY [0] and stores in ARY [0].[19:20] the index plus 1 of the last UNITTYPE/unit number pair that was processed.

SUBTYPE 6 Calls (Reserved)

GETSTATUS unit requests of SUBTYPE 6 are reserved for internal product development. Do not perform GETSTATUS calls of this subtype, and do not use this GETSTATUS subtype in local patches.

Caution

Improper GETSTATUS calls of SUBTYPE 6 can cause an unintended halt/load.

Part 2
SETSTATUS Calls

Section 7
General Information on SETSTATUS

The SETSTATUS intrinsic provides the interface for control of MCP mix, unit, and operational functions from an object program. The SETSTATUS intrinsic can be called only by a DCALGOL MCS or a DCALGOL user program executing under a privileged usercode or a privileged program.

This part of the manual describes calls you can make with SETSTATUS intrinsic that are the equivalent of selected system commands. Each call corresponds to the system command of the same name unless otherwise noted. Refer to the *A Series System Commands Operations Reference Manual* for detailed information about the system commands.

The SETSTATUS intrinsic is referenced as a type Boolean procedure with four parameters:

```
RSLT := SETSTATUS (TYPE, SUBTYPE, VAL, ARRAYROW);
```

The TYPE, SUBTYPE, and VAL parameters are single-precision, REAL variables. ARRAYROW is a single-precision, REAL array row. The appropriate value for each parameter varies in each case. However, for all SETSTATUS calls, element [0] of ARRAYROW must be an integer greater than or equal to 2.

SETSTATUS Request Types

SETSTATUS functions are grouped into major categories by Request Type. The Request Type categories currently implemented are as follows:

Request Type	Description
0	Mix requests. These requests set information about processes in the system.
1	Reserved.
2	Miscellaneous requests.
3	Reserved.
4	Volume library requests. These requests set information about disk and tape volumes from the volume library and the volume directory.
5	Unit requests. These requests set information about peripheral I/O units.

Each Request Type category is further broken down into specific SUBTYPEs for obtaining specific information. These SUBTYPEs are discussed in detail as part of the explanation for each Request Type.

Unless otherwise indicated, the process that is called must have been initiated from an ODT or must be privileged to use Request Types 0, 2, or 5. Unless otherwise indicated, a nonprivileged process can use Request Type 4.

Errors and Results

The Boolean result returned by SETSTATUS is used to indicate two types of error: a hard error or a soft error. If SETSTATUS does not find any errors, it returns a value of FALSE.

A hard error indicates that the SETSTATUS request was incorrectly formed and that SETSTATUS was unable to complete it. An example of a hard error message is "INVALID VALUE IN TYPE.TYPEF", which means that an invalid Request Type was designated. After a hard error is detected, the SETSTATUS request is aborted.

A soft error indicates that SETSTATUS was unable to perform part of the original request, although some action might have been taken. An example of a soft error message is "INVALID NUMBER" (soft error 10), which means that an invalid mix number was designated or an invalid processor number was designated, and so forth. When a soft error is detected, SETSTATUS proceeds to the next request in the current invocation for multiple requests, or SETSTATUS returns to the calling program.

If either a hard or soft error has occurred, bit 0 of the result is turned on. Bits [11:8] indicate the hard error. If bits [11:8] are 0, a soft error has occurred.

To determine which soft error or errors have occurred, you must examine the returned array. The following algorithm describes the required search:

```
INX := 1;
LASTINX := A [Ø].[19:2Ø];
DO BEGIN
   IF BOOLEAN (A [INX].[47:Ø1]) THEN
       %Found a soft error
       %Soft error number in A [INX].[46:Ø8]
       HANDLE_SOFT_ERROR (A [INX].[46:Ø8]);
   INX := * + 1 + A [INX].[38:Ø6];
   END UNTIL INX GEQ LASTINX;
```

This process retrieves all soft errors found in the call by SETSTATUS.

Refer to Appendix A for a list of hard errors and Appendix B for a list of soft errors.

For some calls, SETSTATUS does not complete the processing of the request itself. Instead, it starts an independent system process to finish the request. In these cases, SETSTATUS returns to the calling program before the process finishes. SETSTATUS does not report the success or failure of the independent process. An example would be the SETSTATUS DD (Directory Duplicate) call.

Names

Parameters and results for some SETSTATUS calls use substandard form names. A substandard form name is a one-byte binary character count followed by the EBCDIC characters of the name. Parameters and results for other SETSTATUS calls use standard form names. Standard form names are coded in binary with lengths and codes. For more details about standard form names, refer to the information on the DISPLAYTOSTANDARD function in the *DCALGOL Reference Manual*.

Section 8
SETSTATUS Request Type 0 (Mix Entries)

You can use Request Type 0 SETSTATUS calls to pass information to processes in the mix or to change the status of processes in the mix. Most of these calls correspond to a system command and are arranged alphabetically by name.

The following is a summary list of Request Type 0 calls:

Call Name	SUBTYPE	Description
AX	12	Passes text to a program
DO	19	Turns diagnostic options on or off
DS	2	Terminates designated jobs and tasks
DUMP	8	Invokes the program dump procedure for designated programs
FA	21	Assigns a new or changed value to one or more file attributes of a task in the mix number list
FM	25	Restarts a program that has been suspended because it has tried to open a file whose FORMID attribute is turned on
FR	6	Designates that the input tape reel just read by a tape is the final reel of an unlabeled tape file
FS	0	Starts scheduled jobs or tasks
HI	10	Causes the EXCEPTIONEVENT task attribute to be invoked and optionally assigns a number to the TASKVALUE attribute of the task
IL	25	Designates that the file requested by a program resided on the designated unit, regardless of the label of the unit
LG	27	Controls selective logging of designated tasks
LJ	20	Enters message text into the job log and system log for the jobs or tasks designated in the mix number list
LP	22	Prevents the DS (Discontinue) and QT (Quit) system commands from terminating a designated task
OF	5	Causes the task designated by the mix number to proceed without the optional file
OK	4	Reactivates tasks if they have been suspended or stopped for some reason.

continued

continued

Call Name	SUBTYPE	Description
OU	25	Directs output from the designated task to a designated output device or to any available device of the designated type
PR	9	Assigns the priority of jobs or tasks
QT	3	Terminates the printing or punching of a backup file
RESTART	28	Discontinues and restarts a set of active jobs
RM	7	Responds to a "DUP LIBRARY" message
SM	13	Sends a message to an MCS or a database
STOP	1	Instructs a DCALGOL program running under a privileged usercode to stop a task.
THAW	4	Changes the status of a frozen library from permanent to temporary
UL	25	Responds to a "NO FILE" message

All mix type requests use the array parameter to designate the mix numbers of the process or processes to be acted on. As an optional safety check, the caller can also supply the usercode of the process to be affected. If the process that corresponds to a mix number given in the array does not have the designated usercode, the system returns soft error 10 ("INVALID NUMBER"). You can supply a usercode by turning on bit [47:01] of the TYPE parameter and placing the usercode in substandard form in the last three words of the array parameter A.

General Format for Array A

The following describes a general format of array A for Request Type 0 (Mix Request) SETSTATUS calls:

ARRAYROW	Description
A[0]	Length of valid information contained in A; 1 plus the length of the mix number list in words.
A[1] through A[A[0]–1]	List of mix numbers.

The layout of each entry in the list of mix numbers is as follows:

ARRAYROW	Description
A[n].[15:16]	Mix number.
A[n].[31:08]	Unit type (if applicable).
A[n].[38:06]	Number of words that follow as part of the current entry. The extra words for a mix entry are called ADDLWORDs.
A[n+1]. through A[A[n].[38:06]]	0 or more ADDLWORDs for A[n]—for example, the unit number.

For certain mix request SETSTATUS calls, special information should appear right after the last word used by the mix number list. For example, for an AX (Accept) call, the text response should start at word A[A[0]].

Common Soft Errors for Mix Requests

SETSTATUS can return the following soft error values if the mix number does not reference an executing visible process:

Soft Error Number	Message
1	IMPROPER STACK STATE
10	INVALID NUMBER (This message can be caused by one of the following reasons: • The mix number is not in the range 1 through 9999. • There is not a process with that mix number. • If bit [47:01] of the TYPE parameter is turned on, either the usercode of the process does not match the usercode passed to SETSTATUS in the array parameter A, or the process referenced by the mix number is an MCS.)
78	INVALID STACK TYPE (The mix number refers to a DBS, and either bit [47:01] of the TYPE parameter is turned on or the call is not a DS, LB, or SS SUBTYPE.)
79	INVISIBLE STACK

Examples of Mix Requests

The following show examples of input for two Request Type 0 calls.

Request Type 0, SUBTYPE 27 (LG call)

For the system command *3456,5678 LG*, use the following input:

```
A[Ø] := 3;                  % Length of request
A[1] := Ø & 3456 [15:16]    % First mix number
        & Ø [38:Ø6];        % No additional words for this entry
A[2] := Ø & 5678 [15:16]    % Second mix number
        & Ø [38:Ø6];        % No additional words for this entry
RSLT := SETSTATUS (Ø, 27, Ø, A);
```

Request Type 0, SUBTYPE 25 (UL call)

For the system command *6609 UL MT 30*, use the following input:

```
A[Ø] := 3;                  % Total length of request
A[1] := Ø & 66Ø9 [15:16]    % Mix number
        & 13 [31:Ø8]        % Tape
        &  1 [38:Ø6];       % One more word
A[2] := 3Ø;                 % Unit number
RSLT := SETSTATUS (Ø, 25, 2, A);
```

AX Call

Use the AX call to pass text to a program. This transfer of text can be in response to, or in anticipation of, a task that is requesting an ACCEPT message.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	12
VAL	0
ARRAYROW	Designates the mix number of the task and the ACCEPT task: • A[0] = Pointer to length of text • A[1] through A[A[0]–1] = Mix number list • A[A[0]] = Length of text in characters • A[A[0]+1] through end = Text

Call

```
RSLT := SETSTATUS (Ø, 12, Ø, A);
```

Results

SETSTATUS does not return any unique soft errors for this call. However, SETSTATUS can return hard errors 41 or 42 for an incorrectly structured array parameter or text string.

Examples

For the system command *1852 AX*, use the following input:

```
A[Ø] := 2;
A[1] := 1852;
A[2] := Ø;
A[3] := Ø;
RSLT := SETSTATUS (Ø, 12, Ø, A);
```

For the system command *2223 AX STOP PROCESSING*, use the following input:

```
A[Ø] := 2;
A[1] := 2223;
A[2] := 15;
REPLACE POINTER (A[3]) BY "STOP PROCESSING";
RSLT := SETSTATUS (Ø, 12, Ø, A);
```

For the system command *6131,6132 AX OK*, use the following input:

```
A[Ø] := 3;
A[1] := 6131;
A[2] := 6132;
A[3] := 2;
REPLACE POINTER (A[4]) BY "OK";
RSLT := SETSTATUS (Ø, 12, Ø, A);
```

DO Call

Use this call to turn various diagnostic options on or off. This call is only valid on A 12 and A 15 machines.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	19
VAL	0 = Turn off diagnostic options.
	1 = Turn on diagnostics options.
ARRAYROW	A[0] = 3
	A[1] = 0 & 1 [38:6]
	A[2].[45:1] = 1 (ADS option)
	A[2].[11:1] = 1 (DRH option)
	A[2].[10:1] = 1 (DDC option)

For detailed information about the previously mentioned options ADS, DRH, and DDC, refer to the *System Commands Reference Manual*.

Call

```
RSLT := SETSTATUS (Ø, 19, VAL, A);
```

Results

This call can return soft error 98 ("NO TEST JOB").

Examples

For the system command *DO + ADS*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:6];
A[2] := Ø & 1 [45:1];
RSLT := SETSTATUS (Ø, 19, 1, A);
```

For the system command *DO – DDC*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:6];
A[2] := Ø & 1 [1Ø:1];
RSLT := SETSTATUS (Ø, 19, Ø, A);
```

DS Call

Use this call to terminate designated jobs and tasks.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	2
VAL	0
ARRAYROW	A[0] = Length of request A[1] through A[A[O]–1] = Mix number and option list. The mix number list contains one or two words for each mix number designated. These words are formatted as follows: • A[n] = Mix number. If [38:6] equals 1, ADDLWORD A[n+1] contains the DS options for A[n]. • A[n+1].[7:1] = BASE • A[n+1].[8:1] = ARRAYS • A[n+1].[9:1] = CODE • A[n+1].[10:1] = FILES • A[n+1].[11:1] = PRESENT ARRAYS • A[n+1].[15:1] = DBS • A[n+1].[18:1] = SIBS • A[n+1].[19:1] = LIBRARIES • A[n+1].[20:1] = PRIVATELIBRARIES • A[n+1].[47:1] = If this bit is turned on, it causes the designated options to be combined with options already turned on by the task. The net result controls the program dump.

Call

```
RSLT := SETSTATUS (Ø, 2, Ø, A);
```

Results

This call can return the following soft errors:

Soft Error Number	Message
1	IMPROPER STATE
102	NOT DSED, LIBRARY NOT RESUMABLE
105	NOT DSED, NETWORK SERVICES LIBRARY

continued

continued

Soft Error Number	Message
106	NOT DSED, HOST SERVICES LIBRARY
135	LOCKED PROGRAM

Examples

For the system command *3451 DS*, use the following input:

```
A[Ø] := 2;
A[1] := 3451;
RSLT := SETSTATUS (Ø, 2, Ø, A);
```

For the system command *3451, 7890 DS * , CODE, BASE*, use the following input:

```
A[Ø] := 5;
A[1] := 3451 & 1 [33:1];
A[2] := Ø & 1 [7:1]
          & 1 [9:1]
          & 1 [47:1];
A[3] := 789Ø & 1 [33:1];
A[4] := Ø & 1 [7:1]
          & 1 [9:1]
          & 1 [47:1];
RSLT := SETSTATUS (Ø, 2, Ø, A);
```

DUMP Call

Use this call to invoke the program dump procedure for the designated programs. You can also designate that the program dump be invoked with specific options.

On a system with the InfoGuard security enhancements, when the PROGDUMPFILTER option is turned on, the dump contains only that data that belongs to the environment of the dumping program.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	8
VAL	0
ARRAYROW	A[0] = length of request A[1] through A[A[0]–1] = Mix number and option list. The mix number list contains one or two words for each mix number designated. These words are formatted as follows: • A[n] = Mix number. If A[n].[38:06] equals 1, ADDLWORD A[n+1] contains the DUMP options for A[n]. • A[n+1].[1:1] = FAULT • A[n+1].[2:1] = DSED • A[n+1].[7:1] = BASE • A[n+1].[8:1] = ARRAYS • A[n+1].[9:1] = CODE • A[n+1].[10:1] = FILES • A[n+1].[15:1] = DBS • A[n+1].[18:1] = SIBS • A[n+1].[19:1] = LIBRARIES • A[n+1].[20:1] = PRIVATELIBRARIES • A[n+1].[47:1] = If this bit is turned on, it causes the designated options to be combined with options already turned on by the task. The net result controls the program dump.

Call

```
RSLT := SETSTATUS (Ø, 8, Ø, A);
```

Results

SETSTATUS does not return any unique soft errors for this call.

Examples

For the system command *3451 DUMP*, use the following input:

```
A[Ø] := 2;
A[1] := 3451;
RSLT := SETSTATUS (Ø, 8, Ø, A);
```

For the system command *3451, 7890 DUMP DSED, ARRAYS*, use the following input:

```
A[Ø] := 5;
A[1] := 3451 & 1 [33:1];
A[2] := Ø & 1 [2:1]
          & 1 [8:1];
A[3] := 789Ø & 1 [33:1];
A[4] := Ø & 1 [2:1]
          & 1 [8:1];
RSLT := SETSTATUS (Ø, 8, Ø, A);
```

FA Call

Use this call to assign a new or a changed value to one or more file attributes of a task in the mix number list.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	21
VAL	0
ARRAYROW	A[0] = Index of the word in the array that contains the length of the file attribute assignments
	A[1] through A[A[0]–1] = Mix number list
	A[A[0]] = Length of the file attribute assignments in characters
	A[A[0]+1] through end = File attribute assignments

Call

```
RSLT := SETSTATUS (Ø, 21, Ø, A);
```

Results

This call can return the following soft errors:

Soft Error Number	Message
2	INVALID REPLY (The process was not expecting the FA reply.)
3	NO REPLY NEEDED (The process was not expecting a reply.)
101	WFL NOT INITIATED
113	INV ATTRIBUTE LIST

Examples

For the system command *7890 FA VERSION=12, CYCLE=13*, use the following input:

```
A[Ø] := 2;
A[1] := 789Ø;
A[2] := 2Ø;
REPLACE POINTER (A[3], 8) BY "VERSION=12, CYCLE=13";
RSLT := SETSTATUS (Ø, 21, Ø, A);
```

For the system command *3456 5678 FA TITLE = OBJECT/A*, use the following syntax:

```
A[Ø] := 3;
A[1] := 3456;
A[2] := 789Ø;
A[3] := 14;
REPLACE POINTER (A[4], 8) BY "TITLE=OBJECT/A";
RSLT := SETSTATUS (Ø, 21, Ø, A);
```

FM Call

Use this call to restart a program that has been suspended because it tried to open a file whose FORMID attribute is turned on.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	25
VAL	1
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–1] = List of mix numbers and units

The following is the layout of each request entry:

Array A	Description
A[n].[33:01]	1
A[n].[31:08]	Unit type (refer to Appendix C)
A[n].[15:16]	Mix number
A[n+1]	Unit number

Call

```
RSLT := SETSTATUS (Ø, 25, 1, A);
```

Results

SETSTATUS returns the following soft errors for this call:

Soft Error Number	Message
2	INVALID REPLY (The process was not expecting the FM reply.)
3	NO REPLY NEEDED (The process was not expecting a reply.)
15	UNIT NOT READY
28	UNIT CORRESPONDENCE (Invalid unit type designated.)
31	INVALID UNIT NUMBER

Example

For the system command *1234 FM LP4*, use the following input:

```
A[Ø] := 3;
A[1] := 1234 & 1 [38:Ø6] & 7 [31:Ø8];
A[2] := 4;
RSLT := SETSTATUS (Ø, 25, 1, A);
```

FR Call

Use this call to designate that the input tape reel just read by a process is the final reel of an unlabeled tape file.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	6
VAL	0
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–1] = Mix number list

Call

```
RSLT := SETSTATUS (Ø, 6, Ø, A);
```

Results

SETSTATUS returns the following soft errors for this call:

Soft Error Number	Message
2	INVALID REPLY (The process was not expecting the FR reply.)
3	NO REPLY NEEDED (The process was not expecting a reply.)

Example

For the system command *4567 FR*, use the following input:

```
A[Ø] := 2;
A[1] := 4567;
RSLT := SETSTATUS (Ø, 6, Ø, A);
```

FS Call

Use this call to start scheduled jobs or tasks, provided that enough memory is available. Note that the FS call cannot start a WFL job that is still in the job queue.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	0
VAL	0
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–1] = Mix number list

Call

```
RSLT := SETSTATUS (Ø, Ø, Ø, A);
```

Results

SETSTATUS returns soft error 1 ("IMPROPER STACK STATE") for this call if the process is not in the schedule.

Example

For the system call 9346 FS, use the following input:

```
A[Ø] := 2;
A[1] := 9346;
RSLT := SETSTATUS (Ø, Ø, Ø, A);
```

HI Call

Use this call to cause the EXCEPTIONEVENT of a task to take place and, optionally, to assign a number to the TASKVALUE attribute of the task.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	10
VAL	0
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–1] = Mix number and task value list. Each entry in the list consists of one or two words. The first word of an entry (A[n]) contains the mix number. If A[n].[38:06] equals 1, the ADDLWORD A[n+1] contains the new task value for A[n].

Call

```
RSLT := SETSTATUS (Ø, 1Ø, Ø, A);
```

Results

SETSTATUS can return soft error 1 ("IMPROPER STATE") for this call.

Examples

For the system command *1234 HI*, use the following input:

```
A[Ø] := 2;
A[1] := 1234;
RSLT := SETSTATUS (Ø, 1Ø, Ø, A);
```

For the system command *2468, 1357 HI 6*, use the following input:

```
A[Ø] := 5;
A[1] := 2468 & 1 [38:Ø6];
A[2] := 6;
A[3] := 1357 & 1 [38:Ø6];
A[4] := 6;
RSLT := SETSTATUS (Ø, 1Ø, Ø, A);
```

IL Call

Use this call to designate that the file requested by a program resides on the designated unit, regardless of the label of the unit.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	25
VAL	0
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–2] = Mix number list
	A[A[0]–2].[38:06] = 1
	A[A[0]–2].[31:08] = Unit type (refer to Appendix C)
	A[A[0]–1] = Unit number

Call

```
RSLT := SETSTATUS (Ø, 25, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
2	INVALID REPLY (The process was not expecting the IL reply.)
3	NO REPLY NEEDED (The process was not expecting a reply.)
15	UNIT NOT READY
28	UNIT CORRESPONDENCE (Invalid unit type designated.)
31	INVALID UNIT NUMBER

Example

For the system command *1234 IL MT 29*, use the following input:

```
A[Ø] := 3;
A[1] := 1234 & 1 [38:Ø6] & 13 [31:Ø8];
A[2] := 29;
RSLT := SETSTATUS (Ø, 25, Ø, A);
```

LG Call

Use this call to control selective logging of designated tasks. This call is available only if you have InfoGuard security enhancement software on your system.

If security administrator status is authorized on your system, only a security administrator usercode can invoke this call. If security administrator status is not authorized on your system, any privileged user can invoke this call.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	27
VAL	0
ARRAYROW	A[0] = Length of request A[1] through A[A[0]–1] = Mix number and log code value list. Each entry in the list consists of one or two words. The first word of an entry (A[n]) contains the mix number. If A[n],[38:06] equals 1, the ADDLWORD A[n + 1] contains the new log code value for A[n].

Call

```
RSLT := SETSTATUS (Ø, 27, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
221	REQUEST DENIED. SECADMIN USERCODE REQUIRED
229	SYSTEM HAS NO INFOGUARD AUTHORIZATION – HALT/LOAD TO INSTALL INFOGUARDSUPPORT
230	SYSTEM HAS NO INFOGUARD AUTHORIZATION – INFOGUARDSUPPORT LIBRARY NOT SL-ED
254	SYSTEM HAS NO INFOGUARD AUTHORIZATION –INFOGUARDSUPPORT LIBRARY INITIALIZING

Example

For the system command *3456 LG 5*, use the following input:

```
A[Ø] := 3;
A[1] := 3456 & 1 [38:Ø6];
A[2] := 5;
RSLT := SETSTATUS (Ø, 27, Ø, A);
```

LJ Call

Use this call to enter message text into the job log and system log (SUMLOG) for the jobs or tasks designated in the mix number list.

This SETSTATUS call corresponds to the following system command:

```
<mix number list> LJ <message text>
```

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	20
VAL	0
ARRAYROW	A[0] = Index of the word in the array that contains the length of the message in bytes A[1] through A[A[0]–1] = Mix number list A[A[0]] = Length of the message in bytes A[A[0]+1] through end = Message

Call

```
RSLT := SETSTATUS (Ø, 2Ø, Ø, A);
```

Results

SETSTATUS does not return any unique soft errors for this call.

Example

For the system call *1234, 5678, 9876 LJ THIS IS A COMMENT*, use the following input:

```
A[Ø] := 4;
A[1] := 1234;     % 1st mix number
A[2] := 5678;     % 2nd mix number
A[3] := 9876;     % 3rd mix number
A[4] := 17;
REPLACE POINTER (A[5]) BY "THIS IS A COMMENT" FOR 17;
RSLT := SETSTATUS (Ø, 2Ø, Ø, A);
```

LP Call

Use this call to prevent or permit the DS (Discontinue) and QT (Quit) system commands from terminating a designated task.

This SETSTATUS call with the VAL parameter greater than or equal to 0 corresponds to the following system command:

```
<mix number list> LP
```

This SETSTATUS call with the VAL parameter less than 0 corresponds to the following system command:

```
<mix number list> LP -
```

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	22
VAL	Less than 0 = Unlocks the program (allows DS or QT)
	Greater than or equal to 0 = Locks the program (prevents DS or QT)
ARRAYROW	A[0] = Length of the request
	A[1] through A[A[0]–1] = Mix number list

Call

```
RSLT := SETSTATUS (Ø, 22, VAL, A);
```

Results

If a designated program already has the locked or unlocked state that was requested, soft error 4 ("ALREADY DONE") is returned.

SETSTATUS returns soft error 1 ("IMPROPER STATE") if a designated program is not in the scheduled, active, suspended, or frozen state.

Example

For the system command *1234 LP*, use the following input:

```
A[Ø] := 2;
A[1] := 1234;    % mix number
RSLT := SETSTATUS (Ø, 22, 1, A);
```

NOTOK Call

Use this call to answer "no" to certain RSVPs such as "SORT FILE FULL, OK TO EXPAND."

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	4
VAL	1
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–1] = Mix number list

Call

```
RSLT := SETSTATUS (Ø, 4, 1, A);
```

Results

SETSTATUS returns the following soft errors for this call:

Soft Error Number	Message
2	INVALID REPLY (The process was not expecting the NOTOK reply.)
3	NO REPLY NEEDED (The process was not expecting a reply.)
10	INVALID NUMBER (The mix number of the process was invalid.)

Example

For the system command *2378 NOTOK*, use the following input:

```
A[Ø] := 2;
A[1] := 2378;
RSLT := SETSTATUS (Ø, 4, 1, A);
```

OF Call

Use this call to cause the task designated by the mix number to proceed without the optional file.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	5
VAL	0
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–1] = Mix number list

Call

```
RSLT := SETSTATUS (Ø, 5, Ø, A);
```

Results

If a soft error occurs, bit [47:1] of A[1] will be turned on and bits [46:8] of A[1] will contain a soft error number. Possible soft error numbers include the following:

Soft Error Number	Message
2	INVALID REPLY (The process was not expecting the OF reply.)
3	NO REPLY NEEDED (The process was not expecting a reply.)
10	INVALID NUMBER (The mix number of the process was invalid.)

Example

For the system command *2573 OF*, use the following input:

```
A[Ø] := 2;
A[1] := 2573;
RSLT := SETSTATUS (Ø, 5, Ø, A);
```

OK Call

Use this call to reactivate tasks if they have been suspended or stopped for some reason.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	4
VAL	0
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–1] = Mix number list

Call

```
RSLT := SETSTATUS (Ø, 4, Ø, A);
```

Results

SETSTATUS returns the following soft errors for this call:

Soft Error Number	Message
2	INVALID REPLY (The process was not expecting the OK reply.)
3	NO REPLY NEEDED (The process was not expecting a reply.)
10	INVALID NUMBER (The mix number of the process was invalid.)

Example

For the system command *2378 OK*, use the following input:

```
A[Ø] := 2;
A[1] := 2378;
RSLT := SETSTATUS (Ø, 4, Ø, A);
```

OU Call

Use this call to direct output from the designated task to a designated output device or to any available device of the designated type.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	25
VALUE	3
ARRAYROW	A[0] = Length of request
	A[1] through A[A[1]–1] = List of mix numbers and units

The layout of each request entry is as follows:

Array A	Value or Description
A[n].[33:1]	1
A[n].[31:8]	Unit type (refer to Appendix C)
A[n].[15:16]	Mix number
A[n+1]	Unit number (0 if unspecified)

Call

```
RSLT : = SETSTATUS(Ø, 25, 3, A);
```

Results

The unit designated by the unit number must be online and ready. If a soft error occurs, bit [47:1] of A[1] is turned on and bits [46:8] of A[1] contain a soft error number. Possible soft error numbers include the following:

Soft Error Number	Message
2	INVALID REPLY (The process was not expecting the OU reply.)
3	NO REPLY NEEDED (The process was not expecting a reply.)
10	INVALID NUMBER (The mix number of the process was invalid.)
15	UNIT NOT READY
28	UNIT CORRESPONDENCE (The unit type was invalid.)
31	INVALID UNIT NUMBER

Example

For the system command *2917 OU MT 49*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 2917 [15:16] & 15 [31:8] & 1 [38:6];
A[2] := 49;
RSLT := SETSTATUS (Ø, 25, 3, A);
```

PR Call

Use this call to assign the priority of jobs or tasks. Any changes made by this command are not preserved after a halt/load occurs. The value of the priority must be in the range from 0 to 99.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	9
VAL	0
ARRAYROW	A[0] = Length of request A[1] through A[A[0]–1] = Mix number entry list. The layout of each pair of words used for a mix number entry is as follows: • A[n],[15:16] = Mix number. • A[n].[38:06] = 1 • A[n + 1] = Priority value.

Call

```
RSLT := SETSTATUS (Ø, 9, Ø, A);
```

Results

This call can return soft error 10 ("INVALID NUMBER") because the priority value specified is out of range (0 to 99) or the mix number is invalid.

Example

For the system command *1234 PR 55*, use the following input:

```
A[Ø] := 3;
A[1] := 1234 & 1 [38:Ø6];
A[2] := 55;
RSLT := SETSTATUS (Ø, 9, Ø, A);
```

QT Call

Use this call to terminate the printing or punching of a backup file. The backup file is not removed from the directory. This command can be used only if the print tasks have not been locked with an *LP* system command or an LP SETSTATUS call.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	3
VAL	0
ARRAYROW	A[0] = Length of the request
	A[1] through A[A[0]–1] = Mix number list (one word for each mix number)

Call

```
RSLT := SETSTATUS (Ø, 3, Ø, A);
```

Results

SETSTATUS can return soft error 135 ("LOCKED PROGRAM") for this call.

Example

For the system command *2159 QT*, use the following input:

```
A[Ø] := 2;
A[1] := 2159;
RSLT := SETSTATUS(Ø, 3, Ø, A);
```

RESTART Call

Use this call to programmatically discontinue and restart a set of active jobs.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	28
VAL	0
ARRAYROW	A[0] = Number of mix entries plus 1
	A[1] through A[A[0]] = Mix numbers of the jobs that are to be restarted (one mix number per word)

Call

```
RSLT := SETSTATUS (Ø, 28, Ø, A);
```

Results

In result array A, SETSTATUS returns any errors that have occurred while the program was trying to restart a job. Each word in the array contains the mix number and whether or not an error occurred while the mix was being processed. Bits [15:16] contain the mix number, bit [47:01] (ERRORF) marks an error condition, and bits [46:08] (ERRORVALUEF) indicate what the error was.

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
1	IMPROPER STACK STATE
10	NOT A VALID NUMBER
78	INVALID STACK TYPE
79	INVISIBLE STACK

Example

For the SETSTATUS call *1234 RESTART*, use the following input:

```
A[Ø] := 2;
A[1] := 1234;
RSLT := SETSTATUS (Ø, 28, Ø, A);
```

RM Call

Use this call to respond to a "DUP LIBRARY" message. The call removes the disk file designated in the message and retains the new file.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	7
VAL	0
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–1] = Mix number list

Call

```
RSLT := SETSTATUS (Ø, 7, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
2	INVALID REPLY (The process was not expecting the RM reply.)
3	NO REPLY NEEDED (The process was not expecting a reply.)
10	INVALID NUMBER (The mix number of the process was invalid.)

Example

For the system command *1915 RM*, use the following input:

```
A[Ø] := 2;
A[1] := 1915;
RSLT := SETSTATUS (Ø, 7, Ø, A);
```

SM Call

Use this call to send a control command to an MCS or a database.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	13
VAL	0
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–1] = Mix number list
	A[n] = Characters in message
	A[n+1] = Message

Call

```
RSLT := SETSTATUS (Ø, 13, Ø, A);
```

Results

This call can return the following soft errors:

Soft Error Number	Message
1	IMPROPER STACK STATE
10	INVALID NUMBER
74	DATACOM INACTIVE
88	INVALID MCS

Example

For the system command *1234 SM QUIT*, use the following input:

```
A[Ø] := 2;
A[1] := 1234;
A[2] :=4;
REPLACE POINTER (A[3]) BY "QUIT";
RSLT := SETSTATUS (Ø, 13, Ø, A);
```

STOP Call

Use this call to instruct a DCALGOL program running under a privileged usercode to stop a task. This call has the same effect as the system command *ST*.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	1
VAL	0
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–1] = Mix number list

Call

```
RSLT := SETSTATUS (Ø, 1, Ø, A);
```

Results

SETSTATUS returns the following soft error for this call:

Soft Error Number	Message
10	INVALID NUMBER (The mix number of the process was invalid.)

Example

For the system command *1234 STOP*, use the following input:

```
A[Ø] := 2;
A[1] := 1234;
RSLT := SETSTATUS (Ø, 1, Ø, A);
```

THAW Call

Use this call to change the status of a frozen library from permanent to temporary. A permanently frozen library remains in the mix regardless of whether or not any users are attached to it. A temporary library stays in the mix only as long as a user is attached to it.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	4
VAL	2
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–1] = Mix number list

Call

```
RSLT := SETSTATUS (Ø, 4, 2, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
2	INVALID REPLY (The process was not expecting the THAW reply.)
3	NO REPLY NEEDED (The process was not expecting a reply.)
9	REQUEST DENIED (The process is not a frozen library.)
10	INVALID NUMBER (The mix number of the process is invalid.)

Example

For the system command *2378 THAW*, use the following input:

```
A[Ø] := 2;
A[1] := 2378;
RSLT := SETSTATUS (Ø, 4, 2, A);
```

UL Call

Use this call in response to a "NO FILE" message. The call assigns an unlabeled file on an indicated unit to a task.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	0
SUBTYPE	25
VAL	2
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–1] = List of requests

The layout for each request entry is as follows:

Array A	Value or Description
A[n].[31:08]	Unit type (refer to Appendix C)
A[n].[38:06]	1
A[n].[15:16]	Mix number
A[n + 1]	Unit number

SETSTATUS varies the value of *n* from1 by 2 until the value is greater than or equal to A[0]–1.

Call

```
RSLT := SETSTATUS (Ø, 25, 2, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
2	INVALID REPLY (The process was not expecting a UL reply.)
3	NO REPLY NEEDED (The process was not expecting a reply.)
10	INVALID NUMBER (The mix number of the process was invalid.)
15	UNIT NOT READY
28	UNIT CORRESPONDENCE (An invalid unit type was designated.)
31	INVALID UNIT NUMBER

Example

For the system command *1234 UL MT 30*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1234 [15:16] & 1 [38:Ø6]
          & 13   [31:Ø8];
A[2] := 3Ø;
RSLT := SETSTATUS (Ø, 25, 2, A);
```

Section 9
SETSTATUS Request Type 2 (Miscellaneous Requests)

This section contains miscellaneous SETSTATUS calls. Most of these calls correspond to a system command; they are arranged alphabetically by name.

The following is a summary list of Request Type 2 calls:

Call Name	SUBTYPE	Description
ACCOUNTING	85	Sets systemwide attributes for dependent task accounting and file accounting.
AD	20	Duplicates an active access structure or removes a duplicate structure.
AR	24	Starts the ARCHIVEHANDLER utility.
ARCCOPY	20	Makes an offline backup copy of the archive directory for a disk family.
ARCDUPLICATE	20	Creates a duplicate copy of an archive directory or deletes a duplicate copy of an archive directory.
ARCREPLACE	20	Replaces with a designated replacement directory the current archive directory for a family.
ASD	66	Allocates the factor that is used to calculate the size of the actual segment descriptor table at the next halt/load.
AUTORESTORE	83	Designates or changes the value of the system AUTORESTORE archiving option.
BNAVERSION	67	Designates the version of BNA to be brought up after the next halt/load.
CF	38	Establishes or removes a configuration file title from the system tables.
CHANGE SYSTEMFILE STATUS	86	Permits you to change the SYSTEMFILE status of the file *SYSTEM/FAULTLOG.
CM	6	Designates a new MCP.
COMPILERTARGET	58	Establishes the default TARGET for the system.
COPYCAT	20	Creates a copy of the catalog.
CP	8	Designates a code file as a control program.

continued

continued

Call Name	SUBTYPE	Description
CS	9	Designates a code file as a supervisor program or cancels a supervisor program.
DD	20	Creates or deletes a backup copy of the system directory.
DF	56	Transfers memory dumps from the dumpdisk file to a dumpfile or to tape.
DL	43	Designates that system files reside on a family other than the halt/load family.
DN	29	Controls the creation and assignment of the disk file used for the system memory dumps to disk.
DR	15	Changes the system date used by the MCP (use only for old programs).
DRC	79	Controls disk space usage on a per user basis.
DUMP	1	Dumps the entire contents of memory.
HLUNIT	53	Designates a new halt/load unit.
HOSTGROUP	69	Allocates the name of the host group to be used after the next halt/load.
HOSTNAME	41	Establishes a host name to take effect after the next halt/load.
HS	2	Stops or resumes initialization of new jobs or tasks.
HU	44	Designates a usercode for certain BNA Host Services requests if they come from an ODT that has no terminal usercode assigned to it.
ID	52	Initializes data comm, allocates data comm prefixes, allocates network support processor (NSP) audit options, and terminates data comm.
INSERT IN SUPERVISOR QUEUE	23	Inserts messages in the supervisor input queue.
LC	26	Enters a comment in the system log.
LOGGING	76	Changes the system logging options.
MB	61	Makes a designated BOOTCODE file usable for initialization or removes the eligible status of a BOOTCODE file.
MC	7	Designates a code file as a compiler code file or removes compiler status from a code file.

continued

continued

Call Name	SUBTYPE	Description
MESSAGESUPPORT Initialization	83	Initializes the MESSAGESUPPORT library.
MP	84	Associates an identity with a program.
MU	10	Causes a designated usercode to be entered in the USERDATAFILE as a valid usercode.
NETWORK INITIALIZATION AND TERMINATION	80	Initiates or terminates operation of BNA Version 1 and BNA Version 2. The call also can designate the NETINIT file to use.
NETEX	74	Initiates and terminates NETEX and optionally designates the NETEX init file.
OP	0	Turns on or off MCP run-time options.
PB	50	Prints or punches backup disk or tape files.
PP	8	Designates the privilege status of a code file as privileged, privileged transparent, or nonprivileged.
RB	4	Causes the MCP to build a new access structure for the system directory or the catalog.
RECONFIGURE	51	Regroups hardware resources.
RES	12	Allows an area to be returned to disk pack tables.
RESTRICT	77	Turns on or off any security-related restrictions.
RESTRICT VOLUME	77	Turns on or off any security-related restrictions on volumes.
RO	0	Turns off various system options.
RP	63	Makes a code file resident.
SB	25	Substitutes one backup medium for another.
SBP	47	Displays and sets the time interval used for computing system utilization information.
SECOPT	28	Allocates the security options word for the system.
SEGARRAYSTART	60	Allocates the array size beyond which the MCP segments an array by default.
SF	55	Changes the memory management parameters for the system.

continued

continued

Call Name	SUBTYPE	Description
SI	5	Changes the code file used for system intrinsics.
SL	75	Maps function names to library code files.
SO	0	Turns on various system options.
SQUASH	12	Moves in-use areas on disk so as to reduce fragmented disk allocation and to create larger available areas.
SS	19	Sends a message to data comm stations.
SUPPRESS	8	Prevents jobs or tasks from appearing in a mix display when they are active.
SUPPRESSWARNING	78	Modifies the list of warning messages that the system can display.
SUSPEND/RESUME	37	Suspends or resumes processes of DCALGOL programs running under privileged usercodes.
SYSTEMLANGUAGE	65	Designates the default language to be used for system messages.
TL	22	Releases the current system log file and starts a new one.
TR	14	Changes the system time of day or date used by the MCP. Use the enhanced version of this call for new programs and to designate or change the time zone.
XD	12	Eliminates from the available disk table any defective segments on disk.
XP	8	Marks a nonexecutable unsafe code file as an executable unsafe code file, or the reverse.

AD Call

Use the AD call to request duplication of directories or the catalog.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	20
VAL	1 = For the command AD (<number>).
	1 & 1 [10:1] = For the command AD– <family> (<number>).
	1 & 1 [10:1] & 1 [39:1] = For the command AD– (<number>).
ARRAYROW	Used to designate the family index number, the family name, and the file name. For possible details of this parameter, refer to the information that follows.

Use the following array values for AD and AD– without a family name:

Array A	Value or Description
A [0]	3
A [1]	Family index
A [3]	0
A [4], A [5], A [6]	Should contain 0

Use the following array values for AD– with a family name:

Array A	Value or Description
A [0]	3
A [1]	Family index
A [3]	18
A [4], A [5], A [6]	Family name in substandard form

Call

```
RSLT := SETSTATUS (2, 20, 1, A);
```

Results

If SETSTATUS does not find any errors in the request, a value of FALSE is returned to RSLT. This value means that an independent system process was started to process the request. The success or failure of that process is not reported by SETSTATUS. If SETSTATUS does detect an error with the request, bit [0:1] of the RSLT is turned on. If a soft error occurred (REAL (RSLT) IS 1), bit A [1].[47:1] is turned on and bits

A [1].[46:8] contain the soft error code. Possible soft error values for the AD call include the following:

Soft Error Number	Message
60	INVALID PARAMETER INDEX
63	PARAMETER LENGTH ERROR
69	INVALID VALUE
81	FIVE CHARS REQ
82	INVALID STANDARD FORM

Example

For the system command *AD–(3)*, use the following input:

```
A[Ø] := 3;
A[1] := 3;
A[3] := Ø;
REPLACE POINTER (A[4]) BY Ø FOR 18;
RSLT := SETSTATUS (2, 2Ø, 1 & 1 [1Ø:1] & 1 [39:1], A);
```

AR Call

Use this call to start the utility ARCHIVEHANDLER, which releases an old archive log and creates a new one. When the archive log is full (about 60,000 records), the MCP automatically releases it. When the archive log is released, it is crunched and marked as a nonsystem file.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	24
VAL	0
ARRAYROW	Designates the optional file title

Use the following array values for an AR call without a file title:

Array A	Value or Description
A [0]	2
A [1]	0
A [2]	0

Use the following array values for an AR call with a file title:

Array A	Value or Description
A [0]	2
A [1]	0
A [2] through end	File title in standard form

Call

```
RSLT := SETSTATUS (2, 24, Ø, A);
```

Results

If SETSTATUS does not find any errors in the request, a value of FALSE is returned to RSLT and an independent system process is started to process the request. The success or failure of that process is not reported by SETSTATUS. If SETSTATUS does detect an error with the request, bit [0:1] of the RSLT is turned on. If a soft error occurred (REAL (RSLT) IS 1), bit A [1].[47:1] is turned on and bits A [1].[46:8] contain the soft error code. Possible soft error values for SETSTATUS AR include the following:

Soft Error Number	Message
60	INVALID PARAMETER INDEX
63	PARAMETER LENGTH ERROR

continued

continued

Soft Error Number	Message
81	FIVE CHARS REQ
82	INVALID STANDARD FORM

Examples

For the system command *AR*, use the following input:

```
A[Ø] := 2;
A[1] := Ø;
A[2] := Ø;
RSLT := SETSTATUS (2, 24, Ø, A);
```

For the system command *AR *ARCHIVELOG/011889/141100 ON PACK*, use the following input:

```
A[Ø] := 2;
A[1] := Ø;
REPLACE POINTER (A[2]) BY 48"21Ø6Ø3",
                          48"ØA" "ARCHIVELOG",
                          48"Ø6" "Ø11889",
                          48"Ø6" "141ØØ",
                          48"Ø4" "PACK";
RSLT := SETSTATUS (2, 24, Ø, A);
```

ARCCOPY Call

Use this call to make an offline backup copy of the archive directory for a disk family.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	20
VAL	4
ARRAYROW	Designates the family name and the file title under which the copy is to be made

Use the following array values:

Array A	Value or Description
A [0]	3
A [1]	The family index of the disk to which the copy is to be made. The range can be from 1 through 255.
A [4] through A [6]	The name (in substandard form) of the disk family whose archive directory is to be copied.
A [7] through end	The file title under which the directory is to be copied.

Call

```
RSLT := SETSTATUS (2, 2Ø, 4, A);
```

Results

SETSTATUS returns errors if the family index, family name, or file title are not properly formed. If there are no errors, SETSTATUS starts the independent runner COPYDIR and returns. COPYDIR proceeds to copy the archive directory for the designated family to the designated file.

Example

For the system command *ARCCOPY WORKPACK *BACKUP/WORKPACK ON DISK*, use the following input:

```
 A[Ø] := 3;
 A[1] := 1;    %OUTPUT FAMILY INDEX
% INPUT FAMILY NAME "WORKPACK"
 REPLACE POINTER (ARY [4]) BY 48"Ø8", 8"WORKPACK";
%OUTPUT FILETITLE = "*BACKUP/ARCHIVE/WORKPACK/ØØ1 ON DISK"
 REPLACE POINTER (ARY [7]) BY 48"24Ø6Ø5",
           48"Ø6", 8"BACKUP", 48"Ø7", 8"ARCHIVE",
           48"Ø8", 8"WORKPACK", 48"Ø3", 8"ØØ1",
           48"Ø4", 8"DISK";
 RSLT := SETSTATUS (n, 2Ø, 4, A);
```

ARCDUPLICATE Call

Use this call to create a duplicate of an archive directory or to delete a duplicate of an archive directory.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	20
VAL	5 (for creating a new duplicate)
	5 & 1 [10:01] (for deleting an existing duplicate)
ARRAYROW	Designates the family name and the family index.

Use the following array values:

Array A	Value or Description
A [0]	3
A [1]	The family index of the disk to which the copy is to be made. The range can be from 1 through 255.
A [4] through A [6]	The name (in substandard form) of the disk family whose archive directory is to be copied or deleted.

Call

```
RSLT := SETSTATUS (2, 2Ø, VAL, A);
```

Results

SETSTATUS returns errors if the family index or family name are not properly formed. If there are no errors, SETSTATUS starts the independent runner COPYDIR and returns. COPYDIR proceeds to copy or to delete the duplicate archive directory as requested.

Examples

For the system command *ARCDUPLICATE WORKPACK (2)*, use the following input:

```
A[Ø] := 3;
A[1] := 2;    %NEW FAMILY INDEX
% FAMILY NAME "WORKPACK"
REPLACE POINTER (ARY [4]) BY 48"Ø8", 8"WORKPACK";
RSLT := SETSTATUS (2, 2Ø, 5, A);
```

For the system command *ARCDUPLICATE – WORKPACK (1)*, use the following input:

```
A[Ø] := 3;
A[1] := 1;    %FAMILY INDEX TO DELETE
% FAMILY NAME "WORKPACK"
REPLACE POINTER (ARY [4]) BY 48"Ø8", 8"WORKPACK";
RSLT := SETSTATUS (2, 2Ø, 5 & 1 [1Ø:1], A);
```

ARCREPLACE Call

Use this call to replace the current archive directory for a family with a designated replacement directory.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	20
VAL	6
ARRAYROW	Designates the family name and the replacement directory file title.

Use the following array values:

Array A	Value or Description
A [0]	3
A [1]	1
A [4] through A [6]	The family name (in substandard form).
A [7] through end	Directory file title in standard form.

Call

```
RSLT := SETSTATUS (2, 20, 6, A);
```

Results

SETSTATUS returns errors if the family name or directory name are not properly formed. If there are no errors, SETSTATUS starts the independent runner COPYDIR and returns. COPYDIR proceeds to activate the replacement directory and close the old archive directory for the family.

Example

Use the following input for the system command *ARCREPLACE WORKPACK BACKUP/ARCHIVE/WORKPACK/001*:

```
 A[0] := 3;
 A[1] := 1;
% FAMILY NAME "WORKPACK"
 REPLACE POINTER (ARY [4]) BY 48"08", 8"WORKPACK";
% SUBSTITUTE FILENAME = "BACKUP/ARCHIVE/WORKPACK/001"
 REPLACE POINTER (ARY [7]) BT 48"1F0104",
          48"06" 8"BACKUP, 48"07", 8"ARCHIVE",
          48"08", 8"WORKPACK", 48"03", 8"001";
 RSLT := SETSTATUS (2, 20, 6, A);
```

ASD Call

Use this call to designate the factor that is used to calculate the size of the actual segment descriptor (ASD) table at the next halt/load.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	66
VAL	0
ARRAYROW	User array at least four words long: • A[0] = 4 • A[1] = 0 • A[2] = Value of factor • A[3] = Memory size or 0 & 1 [47:01] if memory size is not to be changed

Call

```
RSLT := SETSTATUS (2, 66, Ø, A);
```

Results

SETSTATUS returns soft error 10 ("INVALID NUMBER") if A [2] or A [3] is out of range.

Example

For the system command *ASD 60*, use the following input:

```
A[Ø] := 4;
A[2] := 6Ø;
A[3] := Ø & 1 [47:Ø1];
RSLT := SETSTATUS (2, 66, Ø, A);
```

AUTORESTORE Call

Use this call to designate or change the value of the system AUTORESTORE archiving option.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	83
VAL	0 = DONTCARE
	1 = NEVER
	2 = YES
ARRAYROW	A[0] = 2
	A[1] = 0

Call

```
RSLT := SETSTATUS (2, 83, VAL, A);
```

Results

SETSTATUS returns soft error 69 if VAL is not in the range 0 to 2.

Examples

For the system command *AUTORESTORE YES*, use the following input:

```
A[Ø] := 2;
A[1] := Ø;
RSLT := SETSTATUS (2, 83, 2, A);
```

BNAVERSION Call

Use this call to designate the version of BNA to be brought up after the next halt/load.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	67
VAL	0
ARRAYROW	A[0] = 2
	A[1] = Next BNA version (1 for BNA Version 1 and 2 for BNA Version 2)

Call

```
RSLT := SETSTATUS (2, 67, Ø, A);
```

Results

SETSTATUS can return the following soft errors if VAL is not equal to 1 or 2:

Soft Error Number	Message
10	INVALID NUMBER
69	INVALID VALUE

Example

For the system command *BNAVERSION = BNAV2*, use the following input:

```
A[Ø] := 2;
A[1] := 2;
RSLT := SETSTATUS (2, 67, Ø, A);
```

CF Call

Use this call to establish or remove a configuration file title from the system tables. The designated file is not used until the system is reconfigured with the RECONFIGURE system command. You must have SECADMIN status to execute this call.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	38
VAL	0 = Removes designation of the configuration file
	1 = Establishes the configuration file
ARRAYROW	A[0] = 2
	A[2] through end = Standard form file title if VAL = 1 or 0 if VAL = 0.

Call

```
RSLT := SETSTATUS (2, 38, VAL, A);
```

Results

If the name of the configuration file is too long, soft error 59 ("NAME TOO LONG") might be returned.

Example

For the system command *CF *SYSTEM/CONFIG/FILE*, use the following input:

```
A[Ø] := 2;
REPLACE POINTER (A[2]) BY 48"160203",
                          48"Ø6" 8"SYSTEM",
                          48"Ø6" 8"CONFIG",
                          48"Ø4" 8"FILE";
RSLT := SETSTATUS (2, 38, 1, A);
```

CHANGE system file STATUS Call

Use this call to change the system file status for the file *SYSTEM/FAULTLOG. (Normally, a disk file marked as a system file cannot be removed or have its name changed.)

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	86
VAL	0 if system file = FALSE
	1 if system file = TRUE
ARRAYROW	A[0] = 2
	A[2] = *SYSTEM/FAULTLOG

Call

```
RSLT := SETSTATUS (2, 86, VAL, A);
```

Result

No unique soft errors are returned for this call.

Example

If you do not want the file *SYSTEM/FAULTLOG ON PACK to be a system file, use following input:

```
A[Ø] := 2;
A[1] := Ø;
REPLACE POINTER (A[2]) BY "*SYSTEM/FAULTLOG ON PACK.";
RSLT := SETSTATUS (2, 86, Ø, A);
```

CM Call

Use this call to designate a new MCP on either the current halt/load family or another family. You can also use this call to remove a pending CM request or to remove a CM designation from the non-halt/load family.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	6
VAL	0 = Cancels a pending CM designation for the current halt/load family
	1 = Establishes the CM designation
	2 = Establishes the temporary CM designation
	4 = Removes the CM designation from a family
ARRAYROW	A[0] = 2
	A[2] through end = Standard form name (not used if VAL = 0)

Call

```
RSLT := SETSTATUS (2, 6, VAL, A);
```

Results

This call can generate the following soft errors:

Soft Error Number	Message
9	REQUEST DENIED (Returned if a CM is already in progress for the current halt/load family when another CM is attempted for that same family.)
82	INVALID STANDARD FORM (An invalid standard form name was supplied.)

Examples

For the system command *CM – ON TESTDISK*, use the following input:

```
A[Ø] := 2;
REPLACE POINTER (A[2]) BY 48"ØCØ1Ø4Ø8",
                          "TESTDISK";
RSLT := SETSTATUS (2, 6, 4, A);
```

For the system command *CM *SYSTEM/MCP ON TESTDISK*, use the following input:

```
A[Ø] := 2;
REPLACE POINTER (A[2]) BY 48"17Ø6Ø3",
                          48"Ø6" 8"SYSTEM",
                          48"Ø3" 8"MCP",
                          48"Ø8" 8"TESTDISK";
RSLT := SETSTATUS (2, 6, 1, A);
```

COMPILERTARGET Call

Use this call to establish the default TARGET for the system.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	58
VAL	0
ARRAYROW	A[0] = 2
	A[1] = 0 & CLASS [4:5]
	A[2] = PRIMARY_ID

CLASS can have one of the following values:

Number	Value
1	LEVEL0
2	B7900
3	LEVEL1
4	LEVEL2

PRIMARY_ID, the internal system designation of a particular system, can take one of the following values:

System	Value
MicroA	20
A 1	16
A 2	1
A 3	5
A 4	19
A 5	10
A 6	2
A 9	6
A 10	9
A 12	15
A 15	7
A 17	17

You can make the call either by designating a machine type such as A 1 or A 9 or by designating a level such as LEVEL0, which is similar to the COMPILERTARGET system command.

Use the following values to designate a LEVEL:

LEVEL	Designations
LEVEL0	Set TARGET_CLASS to 1, PRIMARY_ID to –1.
LEVEL1	Set TARGET_CLASS to 3, PRIMARY_ID to –2.
LEVEL2	Set TARGET_CLASS to 4, PRIMARY_ID to –3.

Use the following values to designate a specific machine type:

Machine Type	Designations
A 9, A 10	Set TARGET_CLASS to 3, PRIMARY_ID to A 9 or A 10 ID (6 or 9).
MicroA and all other A Series systems	Set TARGET_CLASS to 4, PRIMARY_ID to ID from previous table.

Call

```
RSLT := SETSTATUS (2, 58, Ø, A);
```

Results

SETSTATUS does not return any unique soft errors for this call. Exercise caution when you use this command, because no checking is done on the validity of machine type as compared to level.

Examples

For the system command *COMPILERTARGET A9*, use the following input:

```
A[Ø] := 2;
A[1] := Ø & 3 [4:5];   % TARGET CLASS 3
A[2] := 6;             % MACHINE ID OF A9
RSLT := SETSTATUS (2, 58, Ø, A);
```

For the system command *COMPILERTARGET MICROA*, use the following input:

```
A[Ø] := 2;
A[1] := Ø & 4 [4:5];   % TARGET CLASS 4
A[2] := 2Ø;            % MACHINE ID OF MICROA
RSLT := SETSTATUS (2, 58, Ø, A);
```

For the system command *COMPILERTARGET LEVEL1*, use the following input:

```
A[Ø] := 2;
A[1] := Ø & 3 [4:5];   % LEVEL1 HAS TARGET CLASS OF 3
A[2] := -2;            % NO MACHINE ID DESIGNATED, BUT LEVEL = 1
RSLT := SETSTATUS (2, 58, Ø, A);
```

COPYCAT Call

Use the COPYCAT call to create a duplicate copy of the catalog.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	20
VAL	3
ARRAYROW	A[0] = 3
	A[1] = Family index or 0
	A[3] through A [48] = Contains the new file title in standard form with usercode and family name if required

Call

```
RSLT := SETSTATUS (2, 2Ø, 3, A);
```

Results

If SETSTATUS does not find any errors in the request, a value of FALSE is returned to RSLT. Possible soft error values for COPYCAT calls include the following:

Soft Error Number	Message
60	INVALID PARAMETER INDEX
63	PARAMETER LENGTH ERROR
69	INVALID VALUE
81	FIVE CHARS REQ
82	INVALID STANDARD FORM

Example

For the system command *COPYCAT *BACKUP/CAT ON SYSPACK*, use the following input:

```
A[Ø] := 3;
A[1] := Ø;
REPLACE POINTER (A[3]) BY 48"16Ø6Ø3"     % 22 CHAR FILETITLE
    48"Ø6" "BACKUP" 48"Ø3" "CAT" 48"Ø7" "SYSPACK";
RSLT := SETSTATUS (2, 2Ø, 3, A);
```

CP Call

Use the CP call to designate a code file as a control program.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	8
VAL	0 = Changes the status of a control program to that of not being a control program
	1 = Changes a program into a control program
ARRAYROW	A[0] = 2
	A[2] through end = File title in standard form

Call

```
RSLT := SETSTATUS (2, 8, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
4	PREVIOUSLY DONE (File does not need to be changed.)
8	INVALID FILE TYPE (File is not properly structured.)
49	NO FILE (File is not present.)
60	INVALID PARAMETER INDEX (Index name is too large.)
63	PARAMETER LENGTH ERROR (Length of name is too large.)
81	FIVE CHARS REQ (Standard form name is badly formed.)
82	INVALID STANDARD FORM (Standard form name is badly formed.)
255	CODEFILE INCOMPATIBLE WITH THIS MCP VERSION (File is not properly structured.)

Example

For the system command *CP *MYPROGRAM*, use the following input:

```
A[Ø] := 2;
REPLACE POINTER (A[2]) BY 48"ØDØ2Ø1Ø9" 8"MYPROGRAM";
RSLT := SETSTATUS (2, 8, 1, A);
```

CS Call

Use this call to cancel a supervisor program or to designate a code file as a supervisor program.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	9
VAL	0 = Cancels a supervisor program
	1 = Establishes a supervisor program
ARRAYROW	A[0] = 2
	A[2] through end = File title in standard form (not used if VAL = 0)

Call

```
RSLT := SETSTATUS (2, 9, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
8	INVALID FILE TYPE (file is not properly structured)
49	NO FILE (File is not present.)
60	INVALID PARAMETER INDEX (Index name is too large.)
63	PARAMETER LENGTH ERROR (Length of name is too large.)
81	FIVE CHARS REQUIRED (Standard form name is badly formed.)
82	INVALID STANDARD FORM (Standard form name is badly formed.)
255	CODEFILE INCOMPATIBLE WITH THIS MCP VERSION (File is not properly structured.)

Example

For the system command *CS *OUR/SOUP*, use the following input:

```
A[Ø] := 2;
REPLACE POINTER (A[2], 8) BY 48"ØCØ2Ø2Ø3" 8"OUR"
                             48"Ø4" 8"SOUP";
RSLT := SETSTATUS (2, 9, 1, A);
```

DD Call

Use the DD call to delete a duplicate directory or catalog.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	20
VAL	0 = For the command *DD <family> (<number>)*
	0 & 1 [10:1] = For the command *DD – <family> (<number>)*
ARRAYROW	A[0] = 3
	A[1] = Family index
	A[3] = 18
	A[4] through A[6] = Family name in substandard form

Call

```
RSLT := SETSTATUS (2, 2Ø, VAL, A);
```

Results

If SETSTATUS does not find any errors in the request, a value of FALSE is returned to RSLT. This value means that an independent system process was started to process the request. The success or failure of that process is not reported by SETSTATUS. If SETSTATUS does detect an error with the request, bit [0:1] of the RSLT is turned on. If a soft error occurred (REAL (RSLT) IS 1), bit A [1].[47:1] is turned on and bits A [1].[46:8] contain the soft error code. Possible soft error values for the DD call include the following:

Soft Error Number	Message
60	INVALID PARAMETER INDEX
63	PARAMETER LENGTH ERROR
69	INVALID VALUE
81	FIVE CHARS REQ
82	INVALID STANDARD FORM

Example

For the system command *DD DATAPACK (5)*, use the following input:

```
A[Ø] := 3;
A[1] := 5;
A[3] := 18;
REPLACE POINTER (A[4]) BY 48"Ø8" "DATAPACK";
RSLT := SETSTATUS (2, 2Ø, Ø, A);
```

DF Call

Use this call to transfer memory dumps from the dumpdisk file to a dumpfile or to tape.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	56
VAL	0 = If DF with a file name
	1 = IF DF without a file name
ARRAYROW	A[0] = 2
	A[1] = 0
	A[2] through end = Optional file title in standard form

Call

```
RSLT := SETSTATUS (2, 56, Ø, A);
```

Results

This call can generate soft error 61 ("ALREADY RUNNING: A PRIOR EMPTY DUMP DISK FILE OPERATION HAS NOT YET COMPLETED").

Examples

For the system command *DF *SYSTEM/DUMP/FILE*, use the following input:

```
A[Ø] := 2;
REPLACE POINTER (A[2], 8) BY 48"14Ø2Ø3",
                             48"Ø6" 8"SYSTEM",
                             48"Ø4" 8"DUMP",
                             48"Ø4" 8"FILE";
RSLT := SETSTATUS (2, 56, Ø, A);
```

For the system command *DF*, use the following input:

```
A[Ø] := 2;
RSLT := SETSTATUS (2, 56, 1, A);
```

DL Call

Use this call to designate that system files reside on a family other than the halt/load family.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	43
VAL	0
ARRAYROW	A[0] = 2
	A[1].[0:1] = 1 % OVERLAY
	A[1].[1:1] = 1 % LOG
	A[1].[2:1] = 1 % BACKUP
	A[1].[3:1] = 1 % USERDATA
	A[1].[4:1] = 1 % JOBS
	A[1].[5:1] = 1 % CATALOG
	A[1].[6:1] = 1 % DPFILES
	A[1].[7:1] = 1 % SORT
	A[1].[8:1] = 1 % IPFILES
	A[2] through A[5] = Family name in substandard form (for example, 48"04" "DISK").
	A[2] through A[5] = 0, Family name equals a period (.). Zero can be designated only when A[1].[6:1] is turned on.

More than one bit in A [1] can be turned on.

The following descriptions only apply to the MCP/AS systems when A[1].[0:1] is turned on:

Array A	Value or Description
A[2].[3:4]	1 = Sectors per AREASIZE specified
	2 = *DL OVERLAY + ON <family name>*
	3 = *DL OVERLAY- ON <family name>*
A[2].[9:6]	N = Number of the family names
A[2].[35:24]	Sectors per AREASIZE
A[M*3+3]	Family name in substandard form, where M ranges from 0 to N–1.

Call

```
RSLT := SETSTATUS (2, 43, Ø, A);
```

Results

This call can return the following soft errors:

Soft Error Number	Message
55	FAMILY NAME REQUIRED
61	ALREADY RUNNING
221	REQUEST DENIED: SECADMIN USERCODE REQUIRED

Examples

For the system command *DL BACKUP ON MYPACK*, use the following input:

```
A[Ø] := 2;
A[1] := Ø & 1 [2:1];
REPLACE POINTER (A[2], 8) BY 48"Ø6" "MYPACK";
RSLT := SETSTATUS (2, 43, Ø, A);
```

For the system command *DL DPFILES*, use the following input:

```
A[Ø] := 2;
A[1] := Ø & 1 [6:1];
REPLACE POINTER (A[2], 8) BY 48"ØØ" FOR 18;
RSLT := SETSTATUS (2, 43, Ø, A);
```

The following examples are only valid on MCP/AS systems.

For the system command *DL OVERLAY + ON DISK, ON USERPACK*, use the following input:

```
A[Ø] := 2;
A[1] := Ø & 1 [Ø:1];
A[2] := Ø & 2 [3:4]    %  " + "
          & 2 [9:6];   %  two family names specified
REPLACE POINTER (A[3], 8) BY 48"Ø4" "DISK";
REPLACE POINTER (A[6], 8) BY 48"Ø8" "USERPACK";
RSLT := SETSTATUS (2, 43, Ø, A);
```

For the system command *DL OVERLAY 504 ON DISK, ON USERS, ON PACK*, use the following input:

```
A[Ø] := 2;
A[1] := Ø & 1 [Ø:1];
A[2] := Ø & 1 [3:4]        % Sectors per AREASIZE specified
          & 3 [9:6]        % Three family names specified
          & 5Ø4 [35:24];   % Sectors per AREASIZE
REPLACE POINTER (A[3], 8) BY 48"Ø4" "DISK";
REPLACE POINTER (A[6], 8) BY 48"Ø5" "USERS";
REPLACE POINTER (A[9], 8) BY 48"Ø4" "PACK";
RSLT := SETSTATUS (2, 43, Ø, A);
```

DN Call

Use this call to control the creation and assignment of the disk file used for the system memory dumps to disk.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	29
VAL	Number of units in 16K-word (from 16 to 2048)
ARRAYROW	A[0] = 2
	A[1] = 0
	A[2] through end = File title in standard form

Call

```
RSLT := SETSTATUS (2, 29, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
55	FAMILY NAME REQUIRED
59	NAME TOO LONG
92	LOCK IN USE

Example

For the system command *DN DUMPER ON DISK [128]*, use the following input:

```
A[Ø] := 2;
A[1] := Ø;
REPLACE POINTER (B) BY "DUMPER ON DISK.";
DISPLAYTOSTANDARD (POINTER (B), POINTER (A[2]));
RSLT := SETSTATUS (2, 29, 128, A);
```

DR Call

Use this call to change the system date used by the MCP.

Note: *If you are using old programs, you can use this call. For new programs, use the enhanced version of the TR call described later in this section of the manual.*

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	15
VAL	0
ARRAYROW	A[0] = 6
	A[1].[38:6] = 4
	A[2] = Month
	A[3] = Day of month (From 1 to 28, 30, or 31)
	A[4] = Year (must be greater than 1900)
	A[5] = Day of week (Sunday = 0, Monday, = 1, and so on)

Call

```
RSLT := SETSTATUS (2, 15, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
64	INVALID DAY
65	INVALID YEAR
72	INVALID MONTH
76	THREE WORDS REQUIRED
77	INCOMPLETE ENTRY
92	REQUIRED LOCK IN USE (If a date or time change is already running)

Example

For the system command *DR WEDNESDAY JANUARY 11, 1989*, use the following input:

```
A[Ø] := 6;
A[1] := Ø & 4 [38:Ø6];
A[2] := 1;      % January is the first month
A[3] := 11;
A[4] := 1989;
A[5] := 3;      % Wednesday is the third day
RSLT := SETSTATUS (2, 15, Ø, A);
```

DRC Call

Use this call to control disk space usage on a per user basis.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	79
VAL	1 = *DRC +* 2 = *DRC –* 3 = *DRC LB OLDNAME = <family name> NAME = <family name>* 4 = *DRC PG OLDNAME = <family name>*
ARRAYROW	For VAL 3 or 4, use the following: • A[0] = 2 • A[1] = 0 • A[2] = Size of the following family name (including size byte) in bytes • A[3] through A[5] = Family name (for OLDNAME) in substandard form For VAL 3, use the following: • A[6] = Size of following family name (including size byte) in bytes • A[7] through A[9] = Family name (for NAME) in substandard form

Call

```
RSLT := SETSTATUS (2, 79, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
58	NOT ALLOWED (Tried to do DRC + while DRC – was in progress, or vice versa.)
60	INVALID PARAMETER INDEX (Index name is too large.)
63	PARAMETER LENGTH ERROR (Length of name is too large.)
69	INVALID VALUE (VAL is not equal to 1, 2, 3, or 4.)

continued

continued

Soft Error Number	Message
82	INVALID STANDARD FORM (Standard form name is badly formed.)
235	THE DRC SYSTEM IS NOT ACTIVE

Example

For the system command *DRC +*, use the following input:

```
A[Ø] := 2;
A[1] := Ø;
RSLT := SETSTATUS (2, 79, 1, A);
```

DUMP Call

Use this call to dump the entire contents of memory and supply the first 16 characters of text to appear as the reason for the dump.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	1
VAL	2
ARRAYROW	A[0] = 2
	A[1] = 0
	A[2] = Length of the text
	A[3] through A[5] = Text
	A[6] = Code to indicate the extent of the memory dump

Call

```
RSLT := SETSTATUS (2, 1, 2, A);
```

Results

SETSTATUS does not return any unique soft errors for this call.

Example

For the system command *DUMP LOOPING*, use the following input:

```
A[Ø] := 2;
A[1] := Ø;
A[2] := 7;
REPLACE POINTER (A[3], 8) BY "LOOPING";
RSLT := SETSTATUS (2, 1, 2, A);
```

HLUNIT Call

Use this call to designate new halt/load units.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	53
VAL	0 = If no processor mask is designated
	1 = If a processor mask is designated (A 10 only)
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–1] = List of halt/load units (refer to the information that follows)

HDU machines (A 12 and A 15) need the following values for A:

Array A	Description
A[AI].[29:06]	Unit type: • SC = 1 • PK = 17 • MT = 13 • LP = 6
A[AI+1]	Unit number

Resource Management Module (RMM) machines (A 17) need the following values for A:

Array A	Description
A[AI].[38:06]	Entry size (number of extra words in this entry).
A[AI].[32:01]	Unit is the backup unit.
A[AI].[31:08]	Unit type (refer to Appendix C).
A[AI+1]	Unit number.
A[AI+2]	DLP or control number.

EMS machines need the following values for A:

Array A	Description
A[AI].[15:16]	Processor mask (bit mask of processors for this halt/load unit—A 10 only)
A[AI+1]	Unit number
A[AI+2]	DLP or CTL number

Call

```
RSLT := SETSTATUS (2, 53, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
9	REQUEST DENIED
15	UNIT NOT READY
22	NO IO PATH
23	CODE ERROR
28	UNIT CORRESPONDENCE
31	INVALID UNIT NUMBER
41	INVALID UNIT TYPE
52	UNIT NOT AVAILABLE
58	NOT ALLOWED
77	INCOMPLETE ENTRY
151	BAD PROCESSOR ID
183	PROCESSOR ID NOT ALLOWED
185	SCP COULD NOT SET HL UNIT

Examples

For the system command *HLUNIT PK 44* on an EMS machine, use the following input:

```
A[Ø] := 2;
A[2] := 44;
RSLT := SETSTATUS (2, 53, Ø, A);
```

For the system command *HLUNIT PK 44* on an HDU machine, use the following input:

```
A[Ø] := 2;
A[1] := Ø & 17 [29:Ø6];
A[2] := SETSTATUS (2, 53, Ø, A);
```

For the system command *HLUNIT PK 44* on an RMM machine, use the following input:

```
A[Ø] := 2;
A[1] := Ø & 17 [31:Ø6] & 1 [38:Ø6];
A[2] := 44;
RSLT := SETSTATUS (2, 53, Ø, A);
```

HOSTGROUP Call

Use this call to allocate the name of the host group to be used after the next halt/load. No corresponding system command exists for this call.

A host group name can be from 1 to 17 characters long.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	69
VAL	Less than 0 = Makes the next host group name 0
	Greater than or equal to 0 = Sets the next host group to the name in ARRAY A
ARRAYROW	A[0] = 2
	A[1].[47:08] = Index of the first word of the next host group
	A[A[1].[47:08]] = Next host group name in substandard form

Call

```
RSLT := SETSTATUS (2, 69, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
59	NAME TOO LONG
60	INVALID PARAMETER INDEX

Example

To erase any pending change in the host group name, use the following input:

```
A[Ø] := 2;
A[1] := Ø;
RSLT := SETSTATUS (2, 69, -1, A);
```

HOSTNAME Call

Use this call to establish a host name that will take effect at the time of the next halt/load.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	41
VAL	0
ARRAYROW	A[0] = Length of request in words
	A[1].[38:06] = Length of standard form hostname (in words)
	A[2] through A[A[0]–1] = Hostname in standard form

Call

```
RSLT := SETSTATUS (2, 41, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
59	NAME TOO LONG
60	INVALID PARAMETER INDEX
63	PARAMETER LENGTH ERROR
81	FIVE CHARS REQUIRED
82	INVALID STANDARD FORM

Example

For the system command *HOSTNAME = HOSTA*, use the following input:

```
A[Ø] := 4;
A[1] := Ø & 2 [38:Ø6];
REPLACE POINTER (A[2], 8) BY 48"Ø9Ø1Ø1Ø5" 8"HOSTA";
RSLT := SETSTATUS (2, 41, Ø, A);
```

HS Call

Use this call to stop or to resume the initialization of new jobs or tasks.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	2
VAL	1 = Stops the initiation of jobs and tasks
	0 = Resumes the initiation of jobs and tasks
ARRAYROW	A[0] = 2
	A[1] = 0

Call

```
RSLT := SETSTATUS (2, 2, VAL, A);
```

Results

SETSTATUS can return soft error 77 ("INCOMPLETE ENTRY") if A[0] is not equal to 2 or if A[1] is not equal to 0.

SETSTATUS can return soft error 221 ("REQUEST DENIED; SECADMIN USERCODE REQUIRED") if you need security administrator status to execute this call.

Example

For the system command *HS*, use the following input:

```
A[Ø] := 2;
A[1] := Ø;
RSLT := SETSTATUS (2, 2, 1, A);
```

HU Call

Use this call to designate a usercode for certain BNA Host Service requests if they come from an ODT that has no terminal usercode assigned to it.

If security administrator status is authorized for the system, you must have a security administrator usercode to execute this command.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	44
VAL	0 = Establishes a host usercode
	–1 = Deletes a host usercode
ARRAYROW	A[0] = Total length of request
	A[1].[38:06] = Length of standard form usercode in words
	A[2] through A[A[0]–1] = Usercode in standard form

Call

```
RSLT := SETSTATUS (2, 44, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
60	INVALID PARAMETER INDEX
63	PARAMETER LENGTH ERROR
81	FIVE CHARS REQUIRED
82	INVALID STANDARD FORM
221	REQUEST DENIED: SECADMIN USERCODE REQUIRED

Example

For the system command *HU SITE*, use the following input:

```
A[Ø] := 4;
A[1] := Ø & 2 [38:Ø6];
REPLACE POINTER (A[2]) BY 48"ØØØ1Ø1Ø4" 8"SITE";
RSLT := SETSTATUS (2, 44, Ø, A);
```

ID Call

Use this call to initialize data comm, to allocate data comm prefixes, to allocate NSP audit options, and to initiate data comm.

If security administrator status is authorized on your system, this SETSTATUS call can be invoked only by a security administrator usercode. If security administrator status is not authorized on the system, this call can be invoked by any privileged user.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	52
VAL	[34:01] = 1 (AUDITON.)
	[33:01] = 1 (AUDITOFF.)
	[32:01] = 1 (An LSN list is supplied in ARRAYROW.)
	[31:08] = Audit options to be RESET.
	[23:08] = Audit options to be SET.
	[13:01] = 1 (Quit NSP only. Do not remove MCS status of running MCSs.)
	[12:01] = 1 (Display if current DCTRACE option setting is requested.)
	[11:01] = New value of DCTRACE option, if VALUE.[10:01] = 1.
	[10:01] = 1 (Change the setting of the DCTRACE option.)
	[09:01] = 1 (Maximum pseudostations are supplied.)
	[07:01] = 1 (A suffix is supplied in ARRAYROW, to which the NSP audit file can be released.)
	[04:01] = 1 (A circular audit file size is supplied in ARRAYROW.)
	[01:01] = 1 (The data comm prefix has been supplied.)
ARRAYROW	A[0] = Length of entry (including A[0]).
	A[1].[23:08] = NSP number.
	A[1].[38:06] = Length of standard-form data comm prefix (in words) + 1.
	A[2] = NSP number.
	A[3] through A[3 +A[1].[38:6]–2] = Data comm prefix in standard form (only needed if VAL.[01:01] = 1).
	A[A[0]–1] = The maximum number of pseudostations to be allocated (only needed if VAL.[09:01] = 1).

Call

```
RSLT := SETSTATUS (2, 45, VAL, A);
```

Results

SETSTATUS does not generate any soft errors for this call. The call initiates a process separate from SETSTATUS.

Example

For the system command *ID 108 *TESTNDL ON PACK*, use the following input:

```
A[Ø] := 5;
A[1] := Ø & 1Ø8 [23:8] & 3 [38:6];
REPLACE POINTER (A[2], 8) BY 48"1ØØ6Ø2Ø7" 8"TESTNDL",
                             48"Ø4" 8"PACK";
RSLT := SETSTATUS (2, 52, Ø & 1 [Ø:1] & 1 [1:1], A);
```

INSERT IN SUPERVISOR QUEUE Call

No corresponding system command exists for this SETSTATUS call. Use this call to insert messages in the supervisor input queue, if it exists. The supervisor input queue is established when a task executes an ATTACHSPOQ statement. Before the message is inserted in the supervisor input queue, the time of day [TIME(11)] is appended to the front of the message. The supervisor expects the time of day in word 0 of incoming messages and expects the text of the message starting in word 1.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	23
VAL	0
ARRAYROW	A[0] = Index of the word in the array that contains the length of the message in bytes A[2] = Length of the message in bytes A[3] through end = Message

Call

```
RSLT := SETSTATUS (2, 23, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
60	INVALID PARAMETER INDEX
63	PARAMETER LENGTH ERROR

Example

To insert the message "THIS IS A DUMMY MESSAGE" into the supervisor queue, use the following input:

```
A[Ø] := 2;
A[2] := 19;
REPLACE POINTER (A[3], 8) BY "THIS IS A DUMMY MESSAGE" FOR 19;
RSLT := SETSTATUS (2, 23, Ø, A);
```

LC Call

Use this call to enter a comment in the system log (SYSTEM/SUMLOG).

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	26
VAL	0
ARRAYROW	A[0] = Index of the word in the array that contains the length of the comment in bytes
	A[2] = Length of the comment in bytes
	A[3] through end = Comment

Call

```
RSLT := SETSTATUS (2, 26, Ø, A);
```

Results

SETSTATUS does not return any unique soft errors for this call.

Example

For the system command *LC BEGIN SYSTEM TESTING*, use the following input:

```
A[Ø] := 2;
A[2] := 2Ø;
REPLACE POINTER (A[3], 8) BY "BEGIN SYSTEM TESTING" FOR 2Ø;
RSLT := SETSTATUS (2, 26, Ø, A);
```

LOGGING Call

Use this call to change the system logging options.

If security administrator status is authorized on the system, this call can be invoked only by a security administrator usercode. If security administrator status is not authorized on the system, this call can be invoked by any privileged user.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	76
VAL	1 = For *LOGGING **, *LOGGING MINIMAL*, and *LOGGING ALL*
	2 = For *LOGGING <major> <log options>* and for *LOGGING <major>, <minor> <log options>*
ARRAYROW	A[0] = Index of the word of the array where the log setting stream starts
	A[2] through end = Log-setting stream

The log-setting stream is a sequence of 16-bit and 8-bit items. You can build the stream by using EBCDIC pointer manipulations.

If VAL = 1, the log-setting stream is in the following format:

Number of Items	Size of Each	Description
1	16-bit	Number of MAJOR types defined. Number = M. Three exceptions exist. The LOGGING * designation is an exception if M is equal to 4"FF00". LOGGING MINIMAL is the exception if M is equal to 4"FF01". LOGGING ALL is the exception if M is equal to 4"0000".
M	16-bit	A specification whose type is MAJOR. The value for the MAJOR type is S.
		If S is greater than or equal to 4"FF00", this log-setting is a MAJOR type, and S–4"FF00" is the log setting for all MINOR types associated with the MAJOR type.
		If S is less than 4"FF00", this item is an offset to a designation of a MINOR type at POINTER(A[A[0]])+S. This specification whose type is MINOR is shown in the format that follows.

The following information for the MINOR type specification that was previously mentioned:

Number of Items	Size of Each	Description
1	16-bit	Number of MINOR types defined. Number = m.
m	8-bit	A MINOR-type log setting whose value equals s; s is the setting for the combination M,m.

If VAL = 2, the log setting stream is in the following format:

Number of Items	Size of Each	Description
1	48-bit	Number of items to be updated. Number = U.

Each update specification is in the format that follows:

Number of Items	Size of Each	Description
1	16-bit	MAJOR type to be updated. Type = M.
1	16-bit	MINOR type to be updated. Type = m. If m = 0, all MINOR types are updated.
1	8-bit	Mask bit to be applied on the next item. If the applicable bit is turned on, the value of the next item is used; if the applicable bit is turned off, the current value stays in effect.
1	8-bit	Minor-type log setting whose value equals s; s is the updated setting for the combination M,m.

For both cases (VAL = 1 or VAL = 2), the log setting is an 8-bit item that controls logging for the applicable MAJOR type or combination of MAJOR type and MINOR type. The lower four bits control logging to the SUMLOG, and the upper four bits control logging to the job file. Every log record has a result indicator that indicates the type of record. All 8-bit log setting items are controlled independently for each applicable major type or combination of MAJOR type and MINOR type.

The bits and their results are as follows:

SUMLOG	Job File	Log Record Results Indicator
Bit 0	Bit 4	If this bit is turned on, success records are logged.
Bit 1	Bit 5	If this bit is turned on, failure records are logged.
Bit 2	Bit 6	If this bit is turned on, security-relevant records are logged.
Bit 3	Bit 7	If this bit is turned on, security violation records are logged.

Call

```
RSLT := SETSTATUS (2, 76, VAL, A);
```

Results

SETSTATUS does not return any special error messages for this call.

Examples

For the system command *LOGGING* *, use the following input:

```
A[Ø] := 2;
REPLACE POINTER (A[2]) BY 4"FFØØ";
RSLT := SETSTATUS (2, 76, 1, A);
```

For the system command *LOGGING MINIMAL*, use the following input:

```
A[Ø] := 2;
REPLACE POINTER (A[2]) BY 4"FFØ1";
RSLT := SETSTATUS (2, 76, 1, A);
```

For the system command *LOGGING 4,5 NONE*, use the following input:

```
A[Ø] := 2;
REPLACE POINTER (A[2]) BY
          1,  % NUMBER OF UPDATES
    48"ØØØ4",  % MAJOR TYPE
    48"ØØØ5",  % MINOR TYPE
    48"FF",    % MASK ALL BITS
    48"ØØ";    % SETTING OF Ø (NONE)
RSLT := SETSTATUS (2, 76, 2, A);
```

MA Call

Use this call to make a file accessible only to APL by making its attribute APL equal to TRUE. You can also make the file attribute APL equal to FALSE to remove APL access privileges.

This SETSTATUS call with VAL = 0 corresponds to the following system command:

```
MA - APL <file title>
```

This SETSTATUS call with VAL = 1 corresponds to the following system command:

```
MA APL <file title>
```

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	40
VAL	0 = Removes APL access privileges from the file
	1 = Makes the file accessible only to APL
ARRAYROW	A[0] = Index of the word in the array where the file title starts
	A[2] through end = File title in standard form

Call

```
RSLT := SETSTATUS (2, 40, VAL, A);
```

Results

If the specified file does not exist or cannot be found, soft soft error 49 ("NO FILE") or error 7 ("FILE NOT ON DISK") is returned.

Example

For the system command *MA APL (FRED)A/B ON ANYPACK*, use the following input:

```
A[0] := 2;
REPLACE POINTER (A[2]) BY
    48"14060404" 8"FRED" 48"01" 8"A" 48"01" 8"B"
    48"07" 8"ANYPACK";
RSLT := SETSTATUS (2, 40, 1, A);
```

MB Call

Use this call to make a designated BOOTCODE file usable for initialization or to remove the eligible status of a BOOTCODE file.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	61
VAL	0 = For *MB + <filename> ON <family name> PRIORITY*
	1 = For *MB – ON <family name>*
ARRAYROW	A[0] = Index of the word in the array where the file title starts
	A[1].[47:01] = Family index specification is valid
	A[1].[46:08] = Family index specification
	A[1].[11:12] = Priority number
	A[2] through end = File title in standard form

Call

```
RSLT := SETSTATUS (2, 61, VAL, A);
```

Results

SETSTATUS does not return any unique soft errors for this call.

Example

For the system command *MB + BOOT ON DISK (2)*, use the following input:

```
A[Ø] := 2;
A[1] := Ø & 1 [47:Ø1] & 2 [46:Ø8];
REPLACE POINTER (A[2]) BY
48"ØDØ6Ø2" 48"Ø4" 8"BOOT" 48"Ø4" 8"DISK";
RSLT := SETSTATUS (2, 61, Ø, A);
```

MC Call

Use this call to designate a code file whose title is of the form SYSTEM/<file title> as a compiler code file. The call can also remove compiler status from a code file.

If security administrator status is authorized on the system, this SETSTATUS call can be invoked only by a security administrator usercode. If security administrator status is not authorized on the system, this call can be invoked by any privileged user.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	7
VAL	0 = Marks a code file as not being a compiler
	1 = Marks a code file as a compiler
ARRAYROW	A[0] = Index of the word in the array where the file title starts
	A[2] through end = File title in standard form

Call

```
RSLT := SETSTATUS (2, 7, VAL, A);
```

Results

Soft Error Number	Message
49	NO FILE (A file with the designated name was not found.)
221	SECADMIN USERCODE REQUIRED (An InfoGuard security administrator is authorized for the system, and the caller does not have a usercode that is designated for a security administrator.)

A successful SETSTATUS call with a nonzero A [1].[15:16] indicates a warning. The value in A [1].[15:16] contains the warning number.

Warning	Description
941	CODEFILE IS TADS-CAPABLE

Example

For the system command *MC *ALGOL*, use the following input:

```
A[Ø] := 2;
A[1] := Ø;
REPLACE POINTER (A[2], 8) BY 48"1ØØ2Ø2Ø6" 8"SYSTEM",
                             48"Ø5" 8"ALGOL";
RSLT := SETSTATUS (2, 7, 1, A);
```

MP Call (Mark Program)

Use this call to associate an identity with a program.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	84
VAL	0
ARRAYROW	A[0] = 2
	A[2].[32:17] = 5
	A[5].[32:17] = Index to the file title in standard form
	A[5].[46:08] = Number of attribute words
	A[6] through A[n] = The attribute words

Each attribute word has the following layout:

Field	Value or Description
[38:06]	Attribute number (1 = Identity).
[32:17]	Index to the attribute value.
[15:16]	If 0, delete the attribute. If any other value, turn on the attribute.

Call

```
RSLT := SETSTATUS (2, 84, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
18	NOT A CODEFILE.
49	NO FILE (The file is not present.)
255	CODEFILE INCOMPATIBLE WITH THIS MCP VERSION (The file is not properly structured.)

Example

To give the program *OBJECT/MM an identity of ID1, use following input:

```
A[Ø] := 2;
A[2] := Ø & 5 [32:17];
A[5] := Ø & 15 [32:17] & 1 [46:Ø8];
A[6] := Ø & 1 [38:Ø6] & 18 [32:17] & 3 [15:16];
REPLACE POINTER (A[15]) BY 48"ØDØ2Ø2Ø6" 8"OBJECT", 48"Ø2" 8"MM";
REPLACE POINTER (A[18]) BY 48"Ø7Ø1Ø1Ø3 8"ID1";
RSLT := SETSTATUS (2, 84, Ø, A);
```

MU Call

Use this call to enter a designated usercode in the USERDATAFILE as a valid usercode. If a password is designated, it becomes the password for the usercode.

This SETSTATUS call with VAL = 0 corresponds to the following system command:

```
MU <usercode>/<password>
```

This SETSTATUS call with VAL = 1 corresponds to the following system command:

```
MU <usercode>/<password> PRIVILEGED
```

If security administrator status is authorized for the system, you must have security administrator status to execute this function.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	10
VAL	0 = Removes privileged status from a usercode
	1 = Makes the usercode privileged
ARRAYROW	A[0] = Pointer to usercode/password in standard form
	A[2] through end = Usercode/password in standard form

Call

```
RSLT := SETSTATUS (2, 10, VAL, A);
```

Results

If the designated password does not meet existing requirements for minimum password length, SETSTATUS returns soft error 152 ("MINIMUM PASSWORD LENGTH ENFORCED, REQUEST DENIED").

If a valid USERDATAFILE is not present, SETSTATUS returns soft error 9 ("REQUEST DENIED").

Example

For the system command *MU FRED/SECRET PRIVILEGED*, use the following input:

```
A[Ø] := 2;
REPLACE POINTER (A[2]) BY
   48"ØFØ1Ø2Ø4" 8"FRED" 48"Ø6" 8"SECRET";
RSLT := SETSTATUS (2, 1Ø, 1, A);
```

NETEX Call

Use this call to initiate and to terminate NETEX (NETwork EXecutive) software. The call can also designate the NETEX initialization file.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	74
VAL	1 = Terminate NETEX. 2 = Start NETEX. 3 = Start NETEX and use the default initialization file. 4 = Set NETEX initialization file to the default file title. 5 = Start NETEX, but do not use an initialization file. 6 = Set NETEX initialization file to a null value. 7 = Start NETEX and use the specified file title as the initialization file. 8 = Set the NETEX initialization file to the specified title.
ARRAYROW	A[0] = 2

If VAL = 7 or 8, array A has the following values:

Array	Description
A[1]	Length of the standard form file title in 8-bit bytes
A[2] through A[n]	NETEX initialization file title in standard form

Call

```
RSLT := SETSTATUS (2, 74, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
211	NETEX TERMINATE IN PROGRESS
212	NETEX INITIALIZE IN PROGRESS

Example

For the system command *NETEX –*, use the following input:

```
A[Ø] := 2;
RSLT := SETSTATUS (2, 74, 1, A);
```

NETWORK INITIALIZATION AND TERMINATION Call

Use this call to initiate or terminate operation of BNA Version 1, BNA Version 2, and X25. You can also designate the network initialization file to use.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	80
VAL	Request for network changes. .[8:1] = 1 (Going to isolated mode—BNA Version 2.) .[8:1] = 0 (Not going to isolated mode—BNA Version 2.) .[7:4] = Network initialization/termination action: • 0 = No action • 1 = Initiate network • 2 = Slow network shutdown • 3 = Fast network shutdown • 4 = Terminate network library .[3:4] = Network initialization file action: • 0 = No action • 1 = Set NETINIT file to *NULL • 2 = Set NETINIT file to *DEFAULT • 3 = Set NETINIT file to file name
ARRAYROW	A[0] = 4 A[2].[47:1] = 1 if the network is selected by the network name indicator. The following value is also true if A[2].[47:1] = 1: • A[2].[3:4] = Contains the index of the network name in the array. The network name should be in substandard form and can have the following values: – BNAV1 – BNAV2 – X25

continued

continued

Parameter	Value or Description
	A[2].[47:1] = 0 if the network is selected by the network number. The following value is also true if A[2].[47:1] = 0: A[2].[3:4] contains the network index, which can have the following values: • 1 = BNAV1 • 2 = BNAV2 • 3 = X25 It is safer to use the network name, because the numerical values for the network index are subject to change. A[2].[46:1] = 0 if there is no NETINIT file. A[2].[46:1] = 1 if a NETINIT file is supplied. A[2].[7:4] contains the NETINIT offset to the file title. Index of A[2].[7:4] = Index of NETINIT file title in standard form if A[2].[46:1] = 1 Index of A[2].[3:4] = Index of network name in substandard form if A[2].[47.1] = 1

Call

```
RSLT := SETSTATUS (2, 8Ø, VAL, A);
```

Results

SETSTATUS does not return any messages for this call.

Examples for BNAV2

For the system command *BNA +*, use the following input:

```
A[Ø] := 4;
A[1] := Ø;
A[2] := Ø & 1[47:1] & 3[3:4];
REPLACE POINTER (A[3]) BY 48"Ø5" 8"BNAV2";
VAL.[8:1] = 1;  % Isolated mode
VAL.[7:4] = 1;  % Initiate
VAL.[3:4] = Ø;  % Do not use NETINIT file for BNA +
RSLT := SETSTATUS (2, 8Ø, VAL, A);
```

For the system command *BNAV2 +*, use the following input to bring BNAV2 to network mode:

```
A[Ø] := 4;
A[2] := Ø & 1[47:1] & 3[3:4];
REPLACE POINTER (A[3]) BY 48"Ø5" 8"BNAV2";
V.[8:1] := Ø;   % Going to network mode, not isolated mode
V.[7:4] := 1;   % Initiate network
V.[3:4] := Ø;   % No netinit file given, use existing netinit file
RSLT := SETSTATUS (2, 8Ø, V, A);
```

To completely shut down the network (which you can do anytime) use the following input to perform the equivalent of the *BNA* – command:

```
A[Ø] := 4;
A[1] := Ø;
A[2] := Ø & 1[47:1] & 3[3:4];
REPLACE POINTER (A[3]) BY 48"Ø5" 8"BNAV2";
VAL.[7:4] = 4;   % Terminate
RSLT := SETSTATUS (2, 8Ø, VAL, A);
```

Use the following input to perform the equivalent of the *NET + *HOST/NETINIT/V2* command. BNAV2 is the network name.

```
A[Ø] := 4;
A[1] := Ø;
A[2] := Ø & 1[47:1] & 1[46:1] & 4[7:4] & 3[3:4];
REPLACE POINTER (A[3]) BY 48"Ø5" 8"BNAV2";
REPLACE POINTER (A[4]) BY 48"13Ø2Ø3Ø4", 8"HOST", 48"Ø7",
                                        8"NETINIT", 48"Ø2, 8"V2";
VAL.[8:1] := Ø;
VAL.[7:4] = 1;  % Initiate
VAL.[3:4] = 3;  % NETINIT file
RSLT := SETSTATUS (2, 8Ø, VAL, A);
```

Examples for BNAV1

Use the following input to perform the equivalent of the *NET + *HOST/NETINIT/V1* command. BNAV1 is the network name.

```
A[Ø] := 4;
A[1] := Ø;
A[2] := Ø & 1[47:1] & 1[46:1] & 4[7:4] & 3[3:4];
REPLACE POINTER (A[3]) BY 48"Ø5" 8"BNAV1";
REPLACE POINTER (A[4]) BY 48"13Ø2Ø3Ø4", 8"HOST", 48"Ø7",
                                        8"NETINIT", 48"Ø2, 8"V1";
VAL.[8:1] := Ø;
VAL.[7:4] = 1;  % Initiate
VAL.[3:4] = 3;  % NETINIT file
RSLT := SETSTATUS (2, 8Ø, VAL, A);
```

Use the following input to perform the equivalent of the *NET – NOW* command with the network name. BNAV1 is the network name.

```
A[Ø] := 4;
A[1] := Ø;
A[2] := Ø & 1 [47:1] & 3[3:4];
REPLACE POINTER (A[3]) BY
   48"Ø5" 8"BNAV1";
VAL.[7:4] = 4;  % Terminate
RSLT := SETSTATUS (2, 8Ø, VAL, A);
```

Examples for X.25

Use the following input to perform the equivalent of the *X25 +* command. X25 is the network name.

```
A[Ø] := 4;
A[1] := Ø;
A[2] := Ø & 1[47:] & 3[3:4];
REPLACE POINTER (A[3]) BY 48"Ø3" 8"X25";
VAL.[7:4] = 1;  % Initiate
VAL.[3:4] = Ø;  % If no NETINIT file
RSLT := SETSTATUS (2, 8Ø, VAL, A);
```

Use the following input to perform the equivalent of the *X25 –* command. X25 is the network name.

```
A[Ø] := 4;
A[1] := Ø;
A[2] := Ø & 1[47;1] & 3[3:4];
REPLACE POINTER (A[3]) BY 48"Ø3" 8"X25";
VAL.[7:4] = 4;  % Terminate
RSLT := SETSTATUS (2, 8Ø, VAL, A);
```

OP Call

Use this call to turn on or off an MCP run-time option. For a list of available options, refer to the information on the OP (Options) system command in the *System Commands Reference Manual*.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	0
VAL	0 = Turn off option
	1 = Turn on option
ARRAYROW	A[0] = 2
	A[1] = Option number

Call

```
RSLT := SETSTATUS (2, Ø, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
5	IT CANT BE DONE
10	INVALID NUMBER
92	REQUIRED LOCK IN USE (Try again)
194	DISK MIRRORING ALREADY SET
195	DISK MIRRORING NOT YET SET
196	MIRRORED SET(S) STILL EXIST

Example

For the system command *OP + 4*, use the following input:

```
A[Ø] := 2;
A[1] := 4;                    % LPBDONLY
RSLT := SETSTATUS (2, Ø, 1, A);
```

PB Call

Use this call to print or to punch backup disk or tape files.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	50
VAL	0
ARRAYROW	A[0] = Length of request
	A[1] through A[A[0]–1] = List of requests

The layout of each request entry is as follows:

Array A	Value or Description
A[n].[38:06]	0 = Print or punch backup disk files.
A[n].[38:06]	1 = Print backup tape files.
A[n].[31:08]	6 = Line printer.
A[n].[31:08]	11 = Card punch.
A[n].[31:08]	13 = Magnetic tape. Must be designated if [33:01] is 1.
A[n].[15:16]	Mix number. Valid if [38:06] is 0.
A[n+1]	Unit number. Used if [38:06] is 1.

Call

```
RSLT := SETSTATUS (2, 50, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
31	INVALID UNIT NUMBER
71	NO BACKUP ON DK OR PK

Examples

For the system command *PB MT 29*, use the following input:

```
A[Ø] := 3;                          %% total length
A[1] := Ø & 13 [31:Ø8] & 1 [38:6];  %% MT
A[2] := 29;                         %% unit number
RSLT := SETSTATUS (2, 5Ø, Ø, A);
```

For the following system command sequence, use the indicated input:

```
PB 2345 CP
PB 5678 LP
PB 569Ø LP

A[Ø] := 4;
A[1] := Ø & 11 [31:Ø8] & 2345 [15:16]; %% CP, mix number
A[2] := Ø &  6 [31:Ø8] & 5678 [15:16]; %% LP, mix number
A[3] := Ø &  6 [31:Ø8] & 569Ø [15:16];
RSLT := SETSTATUS (2, 5Ø, Ø, A);
```

PP Call

Use the PP call to change the privileged status of a codefile. You can assign privileged (PU) and security administrator (SECADMIN) status.

If security administrator status is authorized on your system, you must have a security administrator usercode to invoke this call. If security administrator status is not authorized on your system, you must have privileged status to invoke this call.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	8
VAL	–1 = Makes a privileged program transparent
	2 = Changes the status of a privileged program to nonprivileged
	3 = Makes a program into a privileged program
	14 = Applies the privileged list to the program
	15 = Adds the privileged list to the program
	16 = Removes the privileged list from the program
ARRAYROW	A[0] = 6
	A[2] = Mask of privileged
	A[3] = Mask of transparent privileges
	A[6] through end = File title in standard form

The following privileges are defined and can be used when VAL is equal to 14, 15, or 16:

Privilege	MASK Bit
PU	MASK bit 0
SECADMIN	MASK bit 1

Call

```
RSLT := SETSTATUS (2, 8, VAL, A);
```

Results

If VAL is 14, 15, or 16 and the call is successful, SETSTATUS returns the new privilege status in word 4 and the new transparent privilege status in Word 5.

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
4	PREVIOUSLY DONE (File does not need to be changed.)
8	INVALID FILE TYPE (File is not properly structured.)
49	NO FILE (File is not present.)
60	INVALID PARAMETER INDEX (Index name is too large.)
63	PARAMETER LENGTH ERROR (Length of name is too large.)
81	FIVE CHARS REQ (Standard form name is badly formed.)
82	INVALID STANDARD FORM (Standard form name is badly formed.)
99	PU AND PU TRANSPARENT CANNOT BOTH BE SET (Would conflict with existing privileges for the file.)
100	SECADMIN AND SECADMIN TRANSPARENT CANNOT BOTH BE SET (Would conflict with existing privileges for the files.)
255	CODEFILE INCOMPATIBLE WITH THIS MCP VERSION (File is not properly structured.)

A successful SETSTATUS call with a nonzero A [1].[15:16] indicates a warning. The value in A [1].[15:16] contains the warning number:

Warning Number	Message
941	CODEFILE IS TADS-CAPABLE

Example

For the system command *PP *MYPROGRAM:PU, SECADMIN TR*, use the following input:

```
A[Ø] := 6;
A[1] := Ø;
A[2] := Ø & 1 [Ø:Ø1];
A[3] := Ø & 1 [Ø1:Ø1];
REPLACE POINTER (A[6]) BY 48"ØDØ2Ø1Ø9" 8"MYPROGRAM";
RSLT := SETSTATUS (2, 8, 14, A);
```

RB Call

Use this call to cause the MCP to build a new access structure for the system directory if the name starting at word 3 of the array A is a family name, or for the catalog if the name is TAPE. If you designate TAPE, the catalog level of the system must not be less than 1. If the name is VOLUMEDIRECTORY, a new key structure for the volume directory is rebuilt.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	4
VAL	0
ARRAYROW	A[0] = 2
	A[1] = 0
	A[2] = Character length of name beginning at A[3]. The name can be either a family name, a volume directory, or a tape.
	A[3] = Name in substandard form.

Call

```
RSLT := SETSTATUS (2, 4, Ø, A);
```

Results

SETSTATUS does not return any unique soft errors for this call. If the catalog level of the system is less than 1, the MCP issues an error message.

Example

For the command *RB ON USERPACK*, use the following input:

```
A[Ø] := 2;
A[1] := Ø;
A[2] := 9;
REPLACE A[3] BY 48"Ø8" 8"USERPACK";
RSLT := SETSTATUS (2, 4, Ø, A);
```

RECONFIGURE Call

Use this call to regroup the hardware resources according to either the designated groups or the default specification. The call causes a halt/load of existing groups that are being reconfigured to the new groups.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	51
VAL	0
ARRAYROW	A[0] = Length of request
	A[1].[38:6] = Number of additional words (A[0]–2).
	A[1].[32:1] = 1 (Reconfigure NOW.)
	A[1].[32:1] = 0 (Wait for a null mix.)
	A[1].[00:1] = 0 (All other systems.)
	A[2] = 0 (All other systems.)
	A[3] = Word length of the source group list.
	A[4] through A[A[3]+3] = Source group list.
	A[A[3]+4] = Word length of the target group list.
	A[A[3]+5] through A[A[0]–1] = Target group list.

The layout of each entry in the source group list is as follows:

Array A	Value or Description
A[n]	Character length of the name.
A[n+1] through end of entry	GROUP (A 12, A 15, or A 17). Resource ID, GROUP, or INSTALLATION on all other systems.

The layout of each entry in the target group list is as follows:

Array A	Value or Description
A[n]	Character length of the name.
A[n+1] through end of entry	Group ID, DEFAULT, or MINIMAL for A 12, A 15, or A 17 systems. Resource ID, DEFAULT, or installation ID for all other systems.

Call

```
RSLT := SETSTATUS (2, 51, Ø, A);
```

Results

SETSTATUS does not return any unique soft errors for this call.

Examples

For the system command *RECONFIGURE GROUP AS TWOBY NOW*, use the following input:

```
A[Ø] := 9;
A[1] := Ø & (A[Ø]-2) [38:Ø6]        %% additional words
          & 1       [32:Ø1];        %% NOW
A[2] := Ø;                          %% not B7ØØ
A[3] := 2;
REPLACE POINTER (A[4], 8) BY        %% A[4]-A[5]
     5 FOR 1 WORDS, 8"GROUP";
A[6] := 2;
REPLACE POINTER (A[7], 8) BY        %% A[7]-A[8]
     5 FOR 1 WORDS, 8"TWOBY";
RSLT := SETSTATUS (2, 51, Ø, A);
```

For the system command *RECONFIGURE INSTALLATION AS SYSA, SYSB, SYSC*, use the following input:

```
A[Ø] := 14;
A[1] := Ø & (A[Ø]-2) [38:Ø6];       %% additional words
A[2] := Ø;                          %% not B7ØØ
A[3] := 3;
REPLACE POINTER (A[4], 8) BY        %% A[4]-A[6]
     12 FOR 1 WORDS, 8"INSTALLATION";
A[7] := 6;
REPLACE POINTER (A[8], 8) BY        %% A[8]-A[13]
     4 FOR 1 WORDS, 8"SYSA  ",
     4 FOR 1 WORDS, 8"SYSB  ",
     4 FOR 1 WORDS, 8"SYSC  ";
RSLT := SETSTATUS (2, 51, Ø, A);
```

RES Call

The Reserve Disk specification consists of the RES (Reserve) system command without the RES or RES– prefix.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	12
VAL	0 = Reserves the disk
	1 = Returns the disk
ARRAYROW	A[0] = Index of the word in the array that contains the length of the message in characters
	A[2] = Length of message in characters
	A[3] through end = Reserve Disk specification

Call

```
RSLT := SETSTATUS (2, 12, VAL, A);
```

Results

Because the Reserve Disk specification is not parsed by SETSTATUS, the caller cannot determine whether the Reserve Disk call is successful. SETSTATUS can return the following soft errors for the RES– call:

Soft Error Number	Message
60	INVALID PARAMETER INDEX
63	PARAMETER LENGTH ERROR

Example

For the system command *RES PK 46 SECTOR 3000 FOR 100*, use the following input:

```
A [Ø] := 2;
A [2] := 25;
REPLACE POINTER (A[3]) BY "PK 46 SECTOR 3ØØØ FOR 1ØØ";
RSLT := SETSTATUS (2, 12, Ø, A);
```

RESTRICT FILE Call

Use this call to turn on or off security-related restrictions on files.

If security administrator status is authorized on the system, this SETSTATUS call can be invoked only by a security administrator usercode. If security administrator status is not authorized, this call can be invoked by any privileged user.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	77
VAL	4 = Unrestricts access to a file
	5 = Restricts access to a file
ARRAYROW	A[0] = Index of the word in the array where the file title starts
	A[2] through end = File title in standard form

Call

```
RSLT := SETSTATUS (2, 77, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
8	INVALID FILE NAME
9	REQUEST DENIED
49	NO FILE

Example

For the system command *RESTRICT FILE *MYPROGRAM*, use the following input:

```
A[Ø] := 2;
REPLACE POINTER (A[2], 8) BY 48"ØDØ2Ø1Ø9" 8"MYPROGRAM";
RSLT := SETSTATUS (2, 77, 5, A);
```

RESTRICT VOLUME Call

Use this call to turn on or off security-related restrictions on volumes.

If security administrator status is authorized on the system, this SETSTATUS call can be invoked only by a security administrator usercode. If security administrator status is not authorized, this call can be invoked by any privileged user.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	77
VAL	2 = Unrestricts access to volume
	3 = Restricts access to volume
ARRAYROW	A[0] = Length of request
	A[1] through A[(A[0]–1)] = List of requests

The layout of each request entry is as follows:

Array A	Value or Description
A[n].[38:6]	1 (Entry size)
A[n+1]	6-character serial number in EBCDIC, with leading EBCDIC zeroes for a numeric string, or trailing blanks for an alphanumeric string if the serial number is less than 6 characters long

Call

```
RSLT := SETSTATUS (2, 77, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
10	INVALID NUMBER
224	VOLUME DIRECTORY NOT READY YET
226	SECOPT TAPE CHECK IS NOT AUTOMATIC

Example

For the system command *RESTRICT – VOLUME ABC, 1234*, use the following input:

```
A[Ø] := 5;
A[1] := Ø & 1 [38:6];
REPLACE POINTER (A[2], 8) BY "ABC    ";
A[3] := Ø & 1 [38:6];
REPLACE POINTER (A[4], 8) BY "ØØ1234";
RSLT := SETSTATUS (2, 77, 2, A);
```

RO Call

This call corresponds to the system command OP (Options). Use this call to turn various options off.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	0
VAL	0
ARRAYROW	A[0] = Number of options + 1 A[1] through A[(A[0]–1)] = Option list; one option member per word in [15:16] of that word

Call

```
RSLT := SETSTATUS (2, Ø, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
5	IT CANT BE DONE
10	INVALID NUMBER
195	DISK MIRRORING NOT SET YET
196	MIRRORED SET(S) STILL EXIST

Example

For the system command *OP– 1,7,9,15*, use the following input:

```
A[Ø] := 5;
A[1] := 1;
A[2] := 7;
A[3] := 9;
A[4] := 15;
RSLT := SETSTATUS (2, Ø, Ø, A);
```

RP Call

Use this call to make a code file resident.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	63
VAL	0 = Cancels resident status of a code file
	1 = Marks a code file as a resident program
ARRAYROW	A[0] = Index of the word in the array where the file title starts
	A[2] through end = File title in standard form

Call

```
RSLT := SETSTATUS (2, 63, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
4	PREVIOUSLY DONE
8	INVALID FILE TYPE
49	NO FILE
60	INVALID PARAMETER INDEX
63	PARAMETER LENGTH ERROR
82	INVALID STANDARD FORM
255	CODEFILE INCOMPATIBLE WITH THIS MCP VERSION

Example

For the system command *RP *SYSTEM/DUMPANALYZER*, use the following input:

```
A[Ø] := 2;
REPLACE POINTER A (A[2]) BY 48"17Ø2Ø2Ø6""SYSTEM
                            48"OC""DUMPANALYZER";
RSLT := SETSTATUS (2, 63, 1, A);
```

SB Call

Use this call to substitute one backup medium for another.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	25
VAL	0
ARRAYROW	A[0] = 8
	A[1] = 0 & 6 [38:06]
	A[2] through A[7] = Substitute backup information

The substitute backup information consists of six words. Each word represents the backup kind list for a specific backup medium. The backup media are as follows:

Word	Backup Medium
A[2]	PACK
A[3]	TAPE7
A[4]	TAPE9
A[5]	PETAPE
A[6]	TAPE
A[7]	DISK

The backup kind list consists of up to 6 bytes, each byte representing a backup kind. The backup kind list should be right-justified. The backup kinds are as follows:

Byte	Backup Kind
4"C1"	DLBACKUP
4"51"	PACK
4"4D"	TAPE7
4"4E"	TAPE9
4"4F"	PETAPE
4"6D"	TAPE
4"41"	DISK

Call

```
RSLT := SETSTATUS (2, 25, Ø, A);
```

Results

The backup kind lists are not checked for duplicates. When you designate an invalid kind or the length is not 6, SETSTATUS returns soft error 9 ("REQUEST DENIED").

Example

For the system command *SB PACK = PACK TAPE9, TAPE7 = TAPE7, TAPE9 = TAPE9, PETAPE = PETAPE, TAPE = TAPE, DISK = DLBACKUP*, use the following input:

```
A[Ø] := 8;
A[1] := Ø & 6 [38:Ø6];
REPLACE POINTER (A[2]) BY 48"ØØØØØØØØ4E51"
                          48"ØØØØØØØØØØ4D"
                          48"ØØØØØØØØØØ4E"
                          48"ØØØØØØØØØØ4F"
                          48"ØØØØØØØØØØ6D"
                          48"ØØØØØØØØØØC1";
RSLT := SETSTATUS (2, 25, Ø, A);
```

SBP Call

Use this call to display and to set the time interval used for computing information on system utilization. On HDU systems, this call can also change the IOINTERRUPT strategy that the MCP uses.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	47
VAL	0
ARRAYROW	A[0] = Number of words in entry
	A[1] through end = System balancing parameters

The SBP parameters consist of sets of 2-word entries. The first word defines the type of parameter, and the second defines the new value.

Two types are defined: INTERVAL (0) and IOINTERRUPT (1).

The INTERVAL type expects a number representing the number of seconds. The IOINTERRUPT type expects a mask representing the required strategy. The mask-bit assignments are as follows:

Mask Bit	Description
0:1	Interrupt word is valid.
1:1	IOFINISH strategy.
2:1	QEMPTY strategy.
3:1	WAITING strategy.

Call

```
RSLT := SETSTATUS (2, 47, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
10	INVALID NUMBER (This message is returned if the parameter type is not INTERVAL or IOINTERRUPT, or if a you designated a negative counterinterval.)
11	ADDITIONAL INFO REQUIRED.

Example

For the system command *SBP INTERVAL = 10 IOINTERRUPT = QEMPTY WAITING*, use the following input:

```
A[Ø] := 5;
A[1] := Ø & 1 [38:Ø6];          % Ø = Interval
A[2] := 1Ø;                     % 1Ø = Seconds
A[3] := 1 & 1 [38:Ø6];          % 1 = IOinterrupt
A[4] := Ø & 1 [Ø:1]             % Valid Interrupt
          & Ø [1:1]             % IOfinish
          & 1 [2:1]             % Qempty
          & 1 [3:1];            % Waiting
RSLT := SETSTATUS (2, 47, Ø, A);
```

SECOPT Call

Use this call to allocate the security options word for the system. The values to be assigned to security options are placed in the appropriate fields of VAL. A bit mask must also be supplied in A[1]. This mask indicates the fields of VAL that are being assigned values.

If security administrator status is authorized on the system, this call can be invoked only by a security administrator usercode. If security administrator status is not authorized on the system, this call can be invoked by any privileged user.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	28
VAL	Values for security options: .[00:01] = Ignored. .[01:01] = 1 (Turns on program dump filtering option.) .[02:01] = 1 (Turns on mandatory disk scrubbing option.) .[03:01] = Turns on nonuser file security: • 0 = PUBLIC • 1 = PRIVATE .[04:01] = 1 (Restricts all hosts.) .[05:01] = 1 (Backup files have usercodes of owners.) .[07:01] = Reserved for future use. .[11:04] = Sets security class: • 0 = Class U (unspecified) • 1 = Class S0 • 2 = Class S1 • 3 = Class S2 .[13:02] = Ignored. .[15:02] = Designates tape security: • 0 = NONE • 1 = AUTOMATIC .[18:03] = Designates password management, where • 0 = MINIMAL • 1 = GENERATED .[27:09] = Reserved for future use.

continued

continued

Parameter	Value or Description
	.[30:03] = An integer in the range 0 to 6, inclusive, sets the CANDE *LAISSEZFILE* option.
	.[31:01] = Reserved for future use.
	.[32:01] = 1 (Sets the CANDE *DIALLOGIN* option.)
	.[33:01] = 1 (Sets the CANDE *ALLLOGIN* option.)
	.[34:01] = 1 (Sets the CANDE *SECDIALIN* option.)
	.[35:01] = 1 (Sets the CANDE *SECPSEUDO* option.)
	.[36:01] = 1 (Sets the CANDE *SECALL* option.)
	.[37:01] = 1 (Sets the CANDE *USECOMSPRIV* option.)
	.[46:09] = Reserved for future use.
	.[47:01] = Ignored.
ARRAYROW	A[0] = 2 (Number of valid words in this array.)
	A[1] = Mask for assigning security options values. In this bit mask, 1s indicate the values in the VAL parameter that are to be applied to the security options word. For example, if bits [18:3] of A[1] are equal to 1, the corresponding field in the VAL parameter ([18:3]) is used to establish the new system password management selection. But if bits 18, 17, and 16 of A[1] are 0, the system password management selection is left unchanged.

On systems without InfoGuard security enhancements software, you can use this SETSTATUS call only to turn on or off file security based on the declarer [06:01]. No other options of the call are allowed.

On systems with InfoGuard security enhancements software, all the options of this call are allowed.

Call

```
RSLT := SETSTATUS (2, 28, VAL, A);
```

Results

If an attempt is made to assign new security options values that are incompatible with the resultant new security class, an error is returned by SETSTATUS, and the system security options are not changed.

If a new security class is assigned that is more restrictive than the current security class, any preexisting security options values that are not explicitly modified to match this new class are automatically modified so that they are compatible with the new class.

To enable the caller to determine how the security options word actually changed, the old value of the security options word is returned in A[2], and the new value of the security options word is returned in A[3]. If no soft error occurs, the original contents of A[1] (the assignment mask) are also preserved by SETSTATUS. If a soft error does occur (for example, if the new value to be assigned is incompatible with the new security class), then A[1] returns the soft error code (for example, 220) in the following manner:

```
A[1] := Ø & SOFTERRORCODE [46:8] & 1 [47:1];
```

Example

For the system command *SECOPT + DISKSCRUB, CLASS = S0*, use the following input:

```
A[Ø] := 2;
A[1] := Ø & 1 [1:1] & 15 [11:4];
VAL  := Ø & 1 [1:1] & 1 [11:4];
RSLT := SETSTATUS (2, 28, V, A);
```

SEGARRAYSTART Call

Use this call to allocate the array size (in words) beyond which the MCP segments an array by default.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	60
VAL	0
ARRAYROW	A[0] = 4
	A[1] = 0
	A[3] = New SEGARRAYSTART value

Call

```
RSLT := SETSTATUS (2, 60, 0, A);
```

Results

This call can return the following soft errors:

Soft Error Number	Message
178	SEGARRAYSTART VALUE TOO SMALL
179	SEGARRAYSTART VALUE TOO LARGE

Example

For the system command *SEGARRAYSTART 8192*, use the following input:

```
A[0] := 4;
A[1] := 0;
A[2] := 1;
A[3] := 8192;
RSLT := SETSTATUS (2, 60, 0, A);
```

SF Call

Use this call to change the memory-management parameters for the entire system.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	55
VAL	0
ARRAYROW	A[0] = 7
	A[1] = OLAYGOAL
	A[2] = AVAILMIN
	A[3] = FACTOR
	A[4] = MEM PRIORITY FACTOR
	A[5] = BUFFERGOAL

The parameter words A[1] through A[5] should be formatted as follows:

Array A	Value or Description
A[n].[47:01]	0 = If the corresponding factor is not to be changed
	1 = If the corresponding factor is to be changed
A[n].[46:47]	New value for factor

Call

```
RSLT := SETSTATUS (2, 55, VAL, A);
```

Results

This call can return the following soft errors:

Soft Error Number	Message
77	INCOMPLETE ENTRY (If A[0] is incorrect)

Example

For the system command *SF FACTOR 100*, use the following input:

```
A[Ø] := 6;
A[1] := Ø;
A[2] := Ø;
A[3] := Ø & 1 [47:Ø1] & 1ØØ [46:47];
A[4] := Ø;          % MEM PRIORITY FACTOR NOT TO BE CHANGED
A[5] := Ø;
RSLT := SETSTATUS (2, 55, Ø, A);
```

SI Call

Use this call to change the code file used for the system intrinsics.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	5
VAL	0
ARRAYROW	A[0] = 2
	A[1] = 0
	A[2] through end = File title in standard form

Call

```
RSLT := SETSTATUS (2, 5, Ø, A);
```

Results

This call can return the following soft errors:

Soft Error Number	Message
59	NAME TOO LONG
63	PARAMETER LENGTH ERROR
81	FIVE CHARACTERS REQ
82	INVALID STANDARD FORM

Example

For the system command *SI *SYSTEM/INTRIN*, use the following input:

```
A[Ø] := 2;
REPLACE POINTER (A[2], 8) BY 48"11Ø2Ø2Ø6" 8"SYSTEM",
                             48"Ø6" 8"INTRIN";
RSLT := SETSTATUS (2, 5, Ø, A);
```

SL Call

Use this call to map function names to library code files and modify the library attributes of library functions.

If security administrator status is authorized on the system, this SETSTATUS call can be invoked only by a security administrator usercode. If security administrator status is not authorized on the system, this call can be invoked by any privileged user.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	75
VAL	2 = To delete a function.
	3 = To add a function (with default attributes).
	9 = To add a function with attributes or to change the attributes of a function.
ARRAYROW	A[0] = 2
	A[1] = 0
	A[2] through A[4] = Function name in substandard form.
	A[5] through A[n] = File title in standard form. Not used if VAL = 2. Required if VAL = 3. Optional if VAL = 9.
	A[n+1] = Number of attributes if VAL = 9. If no file title is specified, n is 4.
	A[n+2] through A[A[n+1] + n+1] = Attribute assignments.

Each attribute assignment occupies one word of the ARRAYROW parameter. The attribute number is placed in bits [31:08], and the attribute value is placed in bits [23:24]. The following attributes can be used:

Number	Name	Type
1	LINKCLASS	Integer (range = 0 to 15)
2	ONEONLY	Boolean
3	TRUSTED	Boolean
4	MCPINIT	Boolean
6	SYSTEMFILE	Boolean

Call

```
RSLT := SETSTATUS (2, 75, VAL, A);
```

Results

The following soft error codes can be returned for this call:

Soft Error Number	Description
10	INVALID NUMBER (The attribute number is not valid.)
18	NOT A CODE FILE
49	NO FILE (Library code file is not resident.)
56	INVALID SYNTAX (No file title is specified for VAL = 3.)
59	NAME TOO LONG
89	UNIT(S) TO BE SAVED (File title is not in standard form.)
119	SIMPLE NAME ERROR (A function name is required.)
214	CODEFILE NOT LIBRARY CAPABLE
215	CANNOT CHANGE A SYSTEM LIBRARY (Cannot use the SL call on a system library.)
216	FUNCTION WAS NOT ESTABLISHED (Function is not present.)
217	CAN'T BE MODIFIED WHILE IN USE
218	SL'ED FILE NOT PUBLIC

A successful SETSTATUS call and nonzero A [1].[46:08] indicates that either the pending title is removed (VAL=2), or the new title is pending (VAL=3). A successful SETSTATUS call with nonzero A [1].15:16] indicates a warning. The value in A [1].[15:16] contains the warning number:

Warning Number	Message
941	CODEFILE IS TADS-CAPABLE

Examples

For the system command *SL USERFUNCTION = SYSTEM/USERLIBRARY*, use the following input:

```
A[Ø] := 2;
A[1] := Ø;
REPLACE POINTER (A[2]) BY 48"ØC" 8"USERFUNCTION";
REPLACE POINTER (B) BY "SYSTEM/USERLIBRARY";
DISPLAYTOSTANDARD (POINTER (B), POINTER (A[5]));
RSLT := SETSTATUS (2, 75, 3, A);
```

For the system command *SL– USERFUNCTION*, use the following input:

```
A[Ø] := 2;
A[1] := Ø;
REPLACE POINTER (A[2]) BY 48"ØC" 8"USERFUNCTION";
RSLT := SETSTATUS (2, 75, 2, A);
```

For the system command *SL USERFUNCTION = *SYSTEM/LIBRARY : LINKCLASS = 2, TRUSTED,* use the following input:

```
A[Ø] := 2;
A[1] := Ø;
REPLACE POINTER (A[2]) BY 48"ØC" 8"USERFUNCTION";
REPLACE POINTER (A[5]Ø BY 48"16Ø2Ø2Ø6"
                          8"SYSTEM" 48"ØB"
                          8"USERLIBRARY";
A[9] := 2;
A[1Ø] := 2 & 1 [31:Ø8];          % Linkclass = 2
A[11] := 1 & 3 [31:Ø8];          % Trusted
```

SO Call

This call corresponds to the system command OP (Options). Use this call to turn various options on.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	0
VAL	1
ARRAYROW	A[0] = Number of options + 1 A[1] through A[(A[0]–1)] = Option list; one option member per word in [15:16] of that word

Call

```
RSLT := SETSTATUS (2, Ø, 1, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
5	IT CANT BE DONE
10	INVALID NUMBER
194	DISK MIRRORING ALREADY SET

Example

For the system command *OP + 1,7,9,15*, use the following input:

```
A[Ø] := 5;
A[1] := 1;
A[2] := 7;
A[3] := 9;
A[4] := 15;
RSLT := SETSTATUS (2, Ø, 1, A);
```

SQUASH Call

Use this call to move in-use areas on disk so that fragmented disk allocation is reduced and larger available areas are created.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	12
VAL	4
ARRAYROW	A[0] = Length of the request in words
	A[1] = Family index
	A[3] = Length of family name in characters + 1
	A[4] = Family name in substandard form

Call

```
RSLT := SETSTATUS (2, 12, 4, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
60	INVALID PARAMETER INDEX
63	PARAMETER LENGTH ERROR

Example

For the system command *SQUASH PRODUCTION (1)*, use the following input:

```
A[Ø] := 7;
A[1] := 1;
A[2] := Ø;
A[3] := 11;
REPLACE POINTER (A[4], 8) BY 48"ØA" 8"PRODUCTION";
RSLT := SETSTATUS (2, 12, 4, A);
```

SS Call

Use this call to send a message to data comm stations identified either by station name or by LSN.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	19
VAL	0
ARRAYROW	A[0] = Index of the word in the array that contains the length of the message in characters. A[1] through A[(A[0]–1)] = Station list. The layout of each entry in the list is as follows: • A[n].[38:06] = Number of words that follow as part of the current entry. • A[n].[15:16] = LSN if field [15.16] equals 0. A[n+1] through end of entry = Station name in standard form if A[n].[38.06] is greater than 0. A[A[0]+1] through end = Message.

Call

```
RSLT := SETSTATUS (2, 19, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
10	INVALID NUMBER
73	UNKNOWN STATION
74	DATACOM INACTIVE
75	INACTIVE STATION

Example

For the system command *SS FRED 14, : HI THERE*, use the following input:

```
A[Ø] := 5;
A[1] := Ø & 2 [38:6];
REPLACE POINTER (A[2], 8) BY 48"Ø8Ø1Ø1Ø4" 8"FRED";
A[4] := Ø & 14 [15:16];
A[5] := 8;
REPLACE POINTER (A[6], 8) BY "HI THERE" FOR 8;
RSLT := SETSTATUS (2, 19, Ø, A);
```

SUPPRESS Call

Use the SUPPRESS call to prevent jobs or tasks from appearing in the mix while they are active.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	8
VAL	4 = Unsuppresses a program
	5 = Suppresses a program display in mix
ARRAYROW	A[0] = 2
	A[2] through end = File title in standard form

Call

```
RSLT := SETSTATUS (2, 8, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
4	PREVIOUSLY DONE (File does not need to be changed.)
8	INVALID FILE TYPE (File is not properly structured.)
49	NO FILE (File is not present.)
60	INVALID PARAMETER INDEX (Index name is too large.)
63	PARAMETER LENGTH ERROR (Length of name is too large.)
81	FIVE CHARS REQ (Standard form name is badly formed.)
82	INVALID STANDARD FORM (Standard form name is badly formed.)
255	CODEFILE INCOMPATIBLE WITH THIS MCP VERSION (File is not properly structured.)

Example

For the system command *SUPPRESS *MYPROGRAM*, use the following input:

```
A[Ø] := 2;
REPLACE POINTER (A[2]) BY 48"ØDØ2Ø1Ø9" 8"MYPROGRAM";
RSLT := SETSTATUS (2, 8, 5, A);
```

SUPPRESSWARNING Call

Use this call to modify the list of warning messages that the system can display.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	78
VAL	0
ARRAYROW	A[0] = 2
	A[1] = N/A.
	A[2] = Number of words following that contain warning numbers. If this number is positive, these warning numbers are to be added to the list of suppressed warnings. If this number is negative, these suppressed warnings are to be subtracted. If A[2] = –1 then A[3] has to be given a text string value of either ALL (to suppress all warnings) or NONE (to suppress no warnings).
	A[3] through end = List of warnings to be suppressed or unsuppressed. A range of warnings can be specified by making one of these values negative. For example, to suppress warnings 5–8, make A[3] equal –5 and make A[4] equal 8.

Call

```
RSLT := SETSTATUS (2, 78, Ø, A);
```

Results

This call does not generate any unique soft errors.

Example

For the system command *SUPPRESSWARNING 3,5–8*, use the following input:

```
A[Ø] := 2;
A[1] := Ø;
A[2] := 3;
A[3] := 3;
A[4] := -5;
A[5] := 8;
RSLT := SETSTATUS (2, 78, Ø, A);
```

SUSPEND/RESUME Call

The SUSPEND/RESUME call allows DCALGOL programs running under privileged usercodes to suspend and resume processes. The SUSPEND call also halts scheduling of new processes until the suspended task process is reactivated.

The program can designate the affected task by means of a mix number, or SETSTATUS can select an appropriate task. In the latter case, SETSTATUS uses the same algorithm as WSSHERRIFF. Among tasks with lowest priority, the most recently started is suspended; among suspended tasks of highest priority, the least recently started is resumed.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	37
VAL.[00:01]	1 = Suspend.
	0 = Resume.
ARRAYROW	A[0] = 2 (Number of words in entry.)
	A[1] = 0 (SETSTATUS selects the task.)
	A[1] greater than 0 = A[1] is the mix number of the process to be resumed or suspended.

Call

```
RSLT := SETSTATUS (2, 37, VAL, A);
```

Results

SETSTATUS returns the mix number of the affected task in A[1].

If SETSTATUS cannot perform the requested action, soft error 5 ("IT CANT BE DONE") is returned. If a mix number is specified, the given task was not in an acceptable state; otherwise, an error indicates that no tasks on the system were in an acceptable state.

Any suspended task is resumable. However, to be suspendable, a task must meet the following criteria:

- The task must be visible (not an invisible independent runner).
- The task must be normal (not discontinued, DUMPING SUSPENDED, and so forth).
- The task must not be a SORT job.
- The task must not be a control program.

Example

The following is an example of SUSPEND/RESUME code:

```
REAL ARRAY A[Ø:4];
REAL MIXNUM;
% THE FOLLOWING WILL SUSPEND TASK WITH MIX #1234
A[Ø] := 2;
A[1] := 1234;
IF SETSTATUS(2,37,1,A) THEN
   BEGIN
     % TASK 1234 WAS NOT SUSPENDABLE
   END;
% THE FOLLOWING WILL RESUME A SYSTEM-SELECTED TASK
A[1] := Ø;
IF SETSTATUS(2,37,Ø,A) THEN
   BEGIN
     % NO SUSPENDED TASKS TO RESUME
   END
ELSE
   BEGIN
     MIXNUM: = A[1];
     % MIXNUM CONTAINS MIX NUMBER OF RESUMED TASK
   END;
```

SYSTEMACCOUNTING Call

Use the SYSTEMACCOUNTING call to set systemwide attributes for dependent task accounting and file accounting.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	85
VAL	0
A [0]	3
A [1].[38:06]	1
A [1].[15:15]	0 = Set dependent task accounting
	1 = Set file accounting
A [2]	0 = UNSPECIFIED accounting
	1 = ANONYMOUS accounting
	2 = IDENTIFIED accounting

Call

```
RSLT := SETSTATUS (2, 85, Ø, A);
```

Results

Possible soft error values for the SYSTEMACCOUNTING call include the following:

Soft Error Number	Message
60	INVALID PARAMETER INDEX
63	PARAMETER LENGTH ERROR
69	INVALID VALUE
81	FIVE CHARS REQ
82	INVALID STANDARD FORM

Example

For the system command *SYSTEMACCOUNTING DEPENDENTTASK IDENTIFIED*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:6];
A[2] := 2;
RSLT := SETSTATUS (2, 85, Ø, A);
```

SYSTEMLANGUAGE Call

Use this call to designate the default language to be used for system messages.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	65
VAL	0
ARRAYROW	A[0] = Number of words of data. Must be greater than 2.
	A[1] = Length of language string in bytes.
	A[2] = Start of language string.

Call

```
RSLT := SETSTATUS (2, 65, Ø, A);
```

Result

No unique soft errors are returned for this call.

Example

For the system command *SYSTEMLANGUAGE ENGLISH*, use the following input:

```
A[Ø] := 5;
A[1] := 7;
REPLACE POINTER (A[2]) BY "ENGLISH";
RSLT := SETSTATUS (2, 65, Ø, A);
```

TL Call

This call corresponds to the TL (Transfer Log) system command. Use this call to release the current system log file and to start a new one.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	22
VAL	0
ARRAYROW	A[0] = 2

Call

```
RSLT := SETSTATUS (2, 22, Ø, A);
```

Results

SETSTATUS returns no unique soft errors for this call.

Example

For the system command *TL*, use the following input:

```
A[Ø] := 2;
RSLT := SETSTATUS (2, 22, Ø, A);
```

TR Call (Old Version)

This is the version of the call used by already existing programs that change the system time of day. This call can also change the date. The time is designated to be on the 24-hour clock. For new programs, use the extended version of the call that immediately follows in this manual.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	14
VAL	0
ARRAYROW	Refer to the values that follow.

If you are changing the time only, array A should contain the following:

Array A	Value or Description
A[0]	5
A[1]	0 & 3 [38:06]
A[2]	Hours
A[3]	Minutes
A[4]	Seconds

If you are changing both time and date, array A should contain the following:

Array A	Value or Description
A[0]	9
A[1]	0 & 7 [38:06]
A[2]	Hours
A[3]	Minutes
A[4]	Seconds
A[5]	Month
A[6]	Day of month
A[7]	Year
A[8]	Day of week: 0 = Sunday, 1 = Monday, and so on

Call

```
RSLT := SETSTATUS (2, 14, Ø, A);
```

Results

This call can return the following soft errors:

Soft Error Number	Message
10	INVALID NUMBER (In time field)
11	ADDITIONAL INFO REQUIRED
64	INVALID DAY
65	INVALID YEAR
72	INVALID MONTH
92	REQUIRED LOCK IS IN USE – TRY LATER

Examples

For the system command *TR 11*, which changes the time only, use following input:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:Ø6];
A[2] := 11;
RSLT := SETSTATUS (2, 14, Ø, A);
```

For the system command *TR 11:05:47*, which changes the time only, use the following input:

```
A[Ø] := 5;
A[1] := Ø & 3 [38:Ø6];
A[2] := 11;
A[3] := 5;
A[4] := 47;
RSLT := SETSTATUS (2, 14, Ø, A);
```

For the system command *TR WEDNESDAY JANUARY 12, 1977*, which changes the date, use the following input:

```
A[Ø] := 7;
A[1] := Ø & 5 [38:Ø6];
A[2] := 11;
A[3] := 1;
A[4] := 12;
A[5] := 1977;
A[6] := 3;                          % Ø is Sunday, 6 is Saturday
RSLT := SETSTATUS (2, 14, Ø, A);
```

For the system command *TR 11:05:47 WEDNESDAY JANUARY 12, 1977*, which changes the time and date, use the following input:

```
A[Ø] := 9;
A[1] := Ø & 7 [38:Ø6];
A[2] := 11;
A[3] := 5;
A[4] := 47;
A[5] := 1;
A[6] := 12;
A[7] := 1977;
A[8] := 3;
RSLT := SETSTATUS (2, 14, Ø, A);
```

TR Call (Extended Version)

Use this version of the call for new programs that change the system time of day, the time zone, the date, or all three. The time is designated to be on the 24-hour clock. For new programs,

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	14
VAL.[4:01]	DATECHANGEF, as follows: • 0 = Date is not being changed. • 1 = Date is being changed.
VAL.[3:02]	TIMEZONECHANGEF, as follows: • 0 = Time zone is not being changed. • 1 = Time zone is being cleared. • 2 = Time zone is being set to a predefined value. • 3 = Time zone is being set to a custom value.
VAL.[1:02]	TIMECHANGEF, as follows: • 0 = Time is not being changed. • 1 = Hours are being designated. • 2 = Hours and minutes are being designated. • 3 = Hours, minutes, and seconds are being designated.
ARRAYROW	A [0] = 21. A [1] = 0 & 19 [38:06]. A [2] = If setting time, the new value for hours (0 through 23). A [3] = If setting minutes, the new value for minutes (0 through 59). A [4] = If setting seconds, the new value for seconds (0 through 59). A [5] = If setting the date, the new value for month (1 through 12). A [6] = If setting the date, the new value for day (1 through 31). A [7] = If setting the date, the new value for year (1900 through 1999). A [8] = If setting the date, the new value for the day of the week (0 through 6). 0 is Sunday and 6 is Saturday. A [9] = If setting the time zone to a predefined value, the new predefined time zone number (1 through 31).

continued

continued

Parameter	Value or Description
	A [10] = If setting the time zone to a custom value, the new offset value: • 1 = Universal time (UT) plus offset produces local time. • 2 = UT minus offset produces local time. A [11] = If setting the time zone to a custom value, the new hours offset (0 through 23). A [12] = If setting the time zone to a custom value, the new minutes offset (0 through 59). A [13] through A [20] = If setting the time zone to a custom value, the new time zone in substandard form (with a maximum of 35 characters) followed by the abbreviation in substandard form (with a maximum of 6 characters).

The following table lists the predefined time zones:

Number	Name	Abbreviation	Offset
1	Alaska Standard Time	(AKST)	– 9 : 00
2	Alaska Daylight Time	(AKDT)	– 8 : 00
3	Pacific Standard Time	(PST)	– 8 : 00
4	Pacific Daylight Time	(PDT)	– 7 : 00
5	Mountain Standard Time	(MOST)	– 7 : 00
6	Mountain Daylight Time	(MODT)	– 6 : 00
7	Central Standard Time	(CST)	– 6 : 00
8	Central Daylight Time	(CDT)	– 5 : 00
9	Eastern Standard Time	(EST)	– 5 : 00
10	Eastern Daylight Time	(EDT)	– 4 : 00
11	Atlantic Standard Time	(AST)	– 4 : 00
12	Atlantic Daylight Time	(ADT)	– 3 : 00
13	Greenland Standard Time	(GST)	– 3 : 00
14	Greenland Daylight Time	(GDT)	– 2 : 00
15	Newfoundland Standard Time	(NST)	– 3 : 30
16	Newfoundland Daylight Time	(NDT)	– 2 : 30
17	Hawaii-Aleutian Standard Time	(HST)	– 10 : 00
18	Hawaii-Aleutian Daylight Time	(HDT)	– 9 : 00
19	Greenwich Mean Time	(GMT) or (GCT)	+ 0 : 00
20	British Summer Time	(BST)	+ 1 : 00
21	Central European Standard Time	(CEST)	+ 1 : 00

continued

continued

Number	Name	Abbreviation	Offset
22	Central European Daylight Time	(CEDT)	+ 2 : 00
23	Eastern European Standard Time	(EEST)	+ 2 : 00
24	Eastern European Daylight Time	(EEDT)	+ 3 : 00
25	Indian Standard Time	(IST)	+ 5 : 30
26	Hong Kong Standard Time	(HKST)	+ 8 : 00
27	Chinese Standard Time	(CHST)	+ 8 : 00
28	Japanese Standard Time	(JST)	+ 9 : 00
29	Eastern Australian Standard Time	(EAST)	+ 10 : 00
30	Eastern Australian Daylight Time	(EADT)	+ 11 : 00
31	Central Australian Standard Time	(CAST)	+ 9 : 30
32	Central Australian Daylight Time	(CADT)	+ 10 : 30
33	Western Australian Standard Time	(WAST)	+ 8 : 00
34	Western Australian Daylight Time	(WADT)	+ 9 : 00

Call

```
RSLT := SETSTATUS (2, 14, VAL, A);
```

Results

This call can return the following soft errors:

Soft Error Number	Message
10	INVALID NUMBER (In the time field.)
11	ADDITIONAL INFO REQUIRED
64	INVALID DAY
65	INVALID YEAR
72	INVALID MONTH
92	REQUIRED LOCK IS IN USE – TRY LATER
139	INVALID TIME (The hours, minutes, seconds, or both minutes and seconds of the new designated time are not within the valid range of 0 through 23 for the hours and 0 through 59 for the minutes and seconds.)

continued

continued

Soft Error Number	Message
140	INVALID DAY OF WEEK (The designated day of the week either does not match the designated day and month or is not in the range of 0 through 6.)
141	INVALID TIME ZONE (The predefined time zone number that you designated is less than or equal to zero or larger than the maximum. You can call SYSTEMSTATUS to find out the maximum value.)
142	INVALID OFFSET DIRECTION (The designated offset direction for a custom time zone is neither 1 nor 2.)
143	INVALID OFFSET VALUE (The hours, minutes, or hours and minutes that you designated for a custom time zone are not within the valid ranges of 0 through 23 for the hours and 0 through 59 for the minutes.)

Examples

For the system command *TR 11:05:47*, which changes the time only, use the following input:

```
A[Ø] := 21;
A[1] := Ø & 19 [38:Ø6];
A[2] := 11;
A[3] := 5;
A[4] := 47;
RSLT := SETSTATUS (2, 14, 3, A);
```

For the system command *TR 11:05:47 PM WEDNESDAY JANUARY 11, 1991*, which changes the time and date, use the following input:

```
A[Ø] := 21;
A[1] := Ø & 19 [38:Ø6];
A[2] := 23;
A[3] := 5;
A[4] := 47;
A[5] := 1;
A[6] := 11;
A[7] := 1991;
A[8] := 3;                          % Ø is Sunday, 6 is Saturday
RSLT := SETSTATUS(2, 14, 3 & (1) [4:Ø1], A);
```

For the system command *TR 15:30 WEDNESDAY JANUARY 11, 1991*, which changes the time and date with a 24-hour clock, use the following input:

```
A[Ø] := 21;
A[1] := Ø & 19 [38:Ø6];
A[2] := 15;
A[3] := 3Ø;
A[5] := 1;
A[6] := 11;
A[7] := 1991;
A[8] := 3;                          % Ø is Sunday, 6 is Saturday
RSLT := SETSTATUS (2, 14, 1 & (1) [4:Ø1], A);
```

For the system command *TR 11:05 TIMEZONE PST*, use the following input:

```
A[Ø] := 21;
A[1] := Ø & 19 [38:Ø6];
A[2] := 11;
A[3] := 5;
A[9] := 3;
RSLT := SETSTATUS (2, 14, 3 & (2) [3:Ø2], A);
```

For the system command *TR 1300 SUNDAY JUNE 4, 1989 –6 TIMEZONE "Kansas City Summer Time" (KCST)*, use the following input:

```
A[Ø] := 21;
A[1] := Ø & 19 [38:Ø6];
A[2] := 13;
A[5] := 6;
A[6] := 4;
A[7] := 1989;
A[9] := Ø;
A[1Ø] := 2;
A[11] := 6;
A[12] := Ø;
REPLACE POINTER (A[12]) BY
        LENGTH ("Kansas City Summer Time").[7:48] FOR 1,
                "Kansas City Summer Time",
        LENGTH ("KCST").[7:48] FOR 1,
                "KCST";
RSLT := SETSTATUS (2, 14, 3 & (3) [3:Ø2], A);
```

For the system command *TR 11:05:47 TUESDAY FEBRUARY 14, 1989 TIMEZONE INDIAN STANDARD TIME*, use the following input:

```
A[Ø] := 21;
A[1] := Ø & 19 [38:Ø6];
A[2] := 11;
A[3] := 5;
A[4] := 47;
A[5] := 2;
A[6] := 14;
A[7] := 1989;
A[8] := 2;                          % Ø is Sunday, 2 is Tuesday
A[9] := 25;
RSLT := SETSTATUS (2, 14, Ø & (1) [4:Ø1]
                          & (2) [3:Ø2]
                          & (3) [1:Ø2], A);
```

For the system command *TR 01:12:00 Saturday November 14, 1959 +0:0 TIMEZONE "REEL TIME" (REEL)*, use the following input:

```
A[Ø] := 21;
A[1] := Ø & 19 [38:Ø6];
A[2] := 1;
A[3] := 12;
A[4] := Ø;
A[5] := 11;
A[6] := 14;
A[7] := 1959;
A[8] := 6;                          % Ø is Sunday, 6 is Saturday
A[1Ø] := 1;
A[11] := Ø;
A[12] := Ø;
REPLACE POINTER (A[13]) BY
        LENGTH ("Reel Time").[7:48] FOR 1,
               "Reel Time",
        LENGTH ("REEL").[7:48] FOR 1,
               "REEL";
RSLT := SETSTATUS (2, 14, Ø & (1) [4:Ø1]
                          & (3) [3:Ø2]
                          & (3) [1:Ø2], A);
```

The examples that follow show system commands that use a fragmentary syntax for setting time zones and the corresponding SETSTATUS calls.

For the system command *TR TIMEZONE + 5:40 "Nepal Time" (NT)*, use the following input:

```
A[Ø] := 21;
A[1] := Ø & 19 [38:Ø6];
A[1Ø] := 1;
A[11] := 5;
A[12] := 4Ø;
REPLACE POINTER (A[13]) BY
        LENGTH ("Nepal Time").[7:48] FOR 1,
               "Nepal Time",
        LENGTH ("NT").[7:48] FOR 1,
               "NT";
RSLT := SETSTATUS (2, 14, Ø & (3) [3:Ø2], A);
```

For the system command *TZ Eastern Daylight Time*, use the following input:

```
A[Ø] := 21;
A[1] := Ø & 19 [38:Ø6];
A[9] := 1Ø;
RSLT := SETSTATUS (2, 14, Ø & (3) [3:Ø2], A);
```

For the system command *TR TIMEZONE NONE*, use the following input:

```
A[Ø] := 21;
A[1] := Ø & 19 [38:Ø6];
RSLT := SETSTATUS (2, 14, Ø & (1) [3:Ø2], A);
```

XD Call

Use this call to eliminate from the available disk table the defective segments on disk. If the disk is mirrored, any changes are made to all the members of the mirrored set.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	12
VAL	2
ARRAYROW	A[0] = Index of the word in the array A that contains the length of the command. A[2] = Length in characters of command. A[3] = Text of command. The command text should be identical to the command except that the letters XD for the command name should be removed.

Call

```
RSLT := SETSTATUS (2, 12, 2, A);
```

Results

No unique soft errors are returned for this call.

Example

For the system command *XD PK 63 SEGMENT 1750000 FOR 30*, use the following input:

```
A[Ø] := 2;
A[2] := 28;
REPLACE POINTER (A[3]) BY "PK 63 SEGMENT 175ØØØØ FOR 3Ø";
RSLT := SETSTATUS (2, 12, 2, A);
```

XP Call

Use the XP call to mark a nonexecutable unsafe code file as an executable unsafe code file and vice versa.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	8
VAL	6 = Marks an executable unsafe code file as a nonexecutable unsafe code file
	7 = Marks a nonexecutable unsafe code file as an executable unsafe code file
ARRAYROW	A[0] = 2
	A[2] through end = File title in standard form

If security administrator status is authorized on your system, you must have a security administrator usercode to invoke this call. If security administrator status is not authorized on your system, you must have privileged status to invoke this call.

Call

```
RSLT := SETSTATUS (2, 8, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
4	PREVIOUSLY DONE (File does not need to be changed.)
8	INVALID FILE TYPE (File is not properly structured.)
49	NO FILE (File is not present.)
60	INVALID PARAMETER INDEX (Index name is too large.)
63	PARAMETER LENGTH ERROR (Length of name is too large.)
81	FIVE CHARS REQ (Standard form name is badly formed.)
82	INVALID STANDARD FORM (Standard form name is badly formed.)
255	CODEFILE INCOMPATIBLE WITH THIS MCP VERSION (File is not properly structured.)

Example

For the system command *XP *MYPROGRAM*, use the following input:

```
A[Ø] := 2;
A[1] := Ø;
REPLACE POINTER (A[2]) BY 48"ØDØ2Ø1Ø9" 8"MYPROGRAM";
RSLT := SETSTATUS (2, 8, 7, A);
```

Section 10
SETSTATUS Request Type 3 (Directory Requests)

This section contains information on SETSTATUS calls that change information in directories. The calls in this section are arranged in alphabetical order.

ARCHIVE RECORD ADD Call

This SETSTATUS call allows DCALGOL programs running under privileged usercodes to add a record to or to replace a record in the archive directory for a disk family.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	3
SUBTYPE	2
VAL	0
ARRAYROW	A[0] = 6
	A[1].[38:6] = 3
	A[2] through A[4] = Disk family name in substandard form. If the family name matches the target name of an active family substitution statement for the calling process, the primary family name is substituted for the designated family name.
	A[6] through end = Contents of the new or replacement archive directory record.
	Bits [31:11] of the first word of the archive record (Word 6 in array A) must contain the length of the record in words (including the first word). The file name to which the archive record applies must be embedded in the record itself. The embedded file name must be in standard form without a family name, and the security byte must have either the value 2 (for a * file) or 3 (for a file name under a usercode). Refer to the description of the format of archive directory records in the *Disk Subsystem Guide*.

Call

```
RSLT := SETSTATUS (3, 2, Ø, A);
```

Results

If the family name or the file name is not properly formed, SETSTATUS returns hard error 44.

If the record (as indicated by its length in bits [31:11] of its first word) extends beyond the end of the array, SETSTATUS returns hard error 57.

If the named family is not online or does not have an active archive directory, or if an error occurs in the archive directory handling routine, SETSTATUS returns hard error 130.

Example

Suppose that you have a properly formed archive record in an array named ARREC. You can enter that record into the archive directory for the disk family named WORKPACK as follows:

```
A[Ø] := 6;
A[1] := Ø & 3 [38:6];
REPLACE POINTER (A[2]) BY 48"Ø8", 8"WORKPACK";
REPLACE POINTER (A[6]) BY POINTER (ARREC)
                          FOR ARREC [Ø].[31:11] WORDS;
B := SETSTATUS (3, 2, Ø, A);
```

ARCHIVE RECORD PURGE Call

This SETSTATUS call allows DCALGOL programs running under privileged usercodes to purge a record from the archive directory for a disk family. This call is equivalent to the WFL statement *ARCHIVE PURGE*.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	3
SUBTYPE	1
VAL	0
ARRAYROW	A[0] = 6
	A[1].[38:6] = 3
	A[2] through A[4] = Disk family name in substandard form. If the family name matches the target name of an active family substitution statement for the calling process, the primary family name is substituted for the designated family name.
	A[6] through end = File name in standard form.

Call

```
RSLT := SETSTATUS (3, 1, Ø, A);
```

Results

If the family name or the file name is not properly formed, SETSTATUS returns hard error 44.

SETSTATUS returns hard error 130 for one of the following reasons:

- The named file does not have an archive record for the named family in the archive directory.
- The named family is not online.
- The named family does not have an active archive directory.
- An error occurs in the archive directory handling routine,

Example

For the WFL statement *ARCHIVE PURGE MY/FILE(FAMILYNAME = WORKPACK)*, use the following input:

```
A[Ø] := 6;
A[1] := Ø & 3 [38:6];
REPLACE POINTER (A[2]) BY 48"Ø8", 8"WORKPACK";
REPLACE POINTER (A[6]) BY
                          48"ØBØ1Ø2", 48"Ø2", 8"MY";
                          48"Ø4", 8"FILE";
B := SETSTATUS (3, 1, Ø, A);
```

Section 11
SETSTATUS Request Type 4 (Disk and Tape Volumes)

This section contains information on SETSTATUS calls that change information in the volume directory. The calls in this section are arranged in alphabetical order.

VOLUME ADD Call

This SETSTATUS call allows DCALGOL programs running under privileged usercodes to add a record to the volume directory. This call can be used only if the TAPECHECK = AUTOMATIC option of the system command SECOPT (Security Options) has been designated on the system. This call corresponds to the WFL *VOLUME ADD* statement. Although the volume directory is changed, the volume library is not changed.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	4
SUBTYPE	1
VAL	0
ARRAYROW	A[0] = Number of words in the record to be added to the volume directory. A[1] through A[A[0]] = Contents of the record to be added. The format of volume directory records is documented in the *Security Administration Guide*.

Call

```
RSLT := SETSTATUS (4, 1, Ø, A);
```

Results

Errors are returned in the normal manner.

Error number 42 is a hard error. The value specified in A[0] is greater than the maximum allowed for a record in the volume directory.

For soft errors, the following codes apply:

Soft Error Number	Message
227	ERROR PROCESSING VOLUME DIRECTORY REQUEST (The request was incorrect, usually because a listed serial number is already in the volume directory or the specified record was not correctly formed.)
231	VOLUME DIRECTORY NOT ACTIVE

Example

For the WFL statement *VOLUME ADD SCRATCH (TAPE, SERIALNO=5646)*, use the following input:

```
ARRAY  REC [0:200];    % ARRAY IN WHICH TO CONSTRUCT VOL DIR RECORD

% DEFINITION OF THE STRUCTURE OF A VOLUME DIRECTORY RECORD
 DEFINE
    MARKERF = [47:16] #,
    HDRBLOCKLENGTHF = [31:11] #,
  VSTYPEX = 1 #,                % DETERMINES STRUCTURE OF RECORD
    VSRECF = [47:8] #,          % LETTER "D" MEANS DATA BLOCK
    VSLABELTYPEF = [39:8] #,    % TAPE LABELTYPEL IN [3:4]
                                %      & LABELTYPES IN [4:1]
                                %   (0 FOR SCRATCH TAPES)
    VSLEVELF = [31:8] #,        % MCP RELEASE LEVEL FOR THIS VOLUME
                                %    DIRECTORY DATA BLOCK STRUCTURE
    VSTYPEF  = [15:4] #,        %  KIND OF RECORD
      VSTPGMTV = 1 #,           %    SCRATCH TAPE
      VSTMTV   = 2 #,           %    NON-SCRATCH TAPE
      VSTPKV   = 3 #,           %    DISK OR PACK
    VSWILDF = [10:1] #,         % DON'T NEED TO MATCH FAMILY NAME,
                                %  CREATION DATES, ETC.
    VSVARF = [ 9:10] #,         % INDEX TO START OF VARIABLE LENGTH
                                %    STUFF;  RECORDS PRODUCED BY
VSTIMESTAMPX = 2 #,            % DATE & TIME WHEN THIS RECORD
                                %   CHANGED OR UPDATED.
  VSDATEX = 3 #,                % DATE   FIRST VOLUME IN FAMILY WAS
                                % ON FIRST TAPE IN VOLUME SET
  VSCRTNSITEX = 4 #,            % CREATION SITE SYSTEM SERIAL
                                %   (BINARY)
```

```
  VSSAVEX = 5 #,                % SAVEFACTOR OF FIRST FILE ON FIRST
                                %   TAPE VOLUME.
  VSUSERX = 6 #,                % OWNER OF TAPE (3 WORDS):
                                %  NORMALLY LENGTH BYTE FOLLOWED BY
                                %    USERCODE.  BUT THERE ARE
                                %    SPECIAL CASES:
                                %  FIRST BYTE = Ø MEANS UNOWNED
    VSASTERISK                  %  THIS PATTERN MEANS OWNED BY *
        = 48"Ø2ØØ" & "*" #,
  VSFAMILYX = 9 #,              % VOLUME NAME (3 WORDS)
 VSSECURITYX = 12 #,           % SECURITY ATTRIBUTES FOR TAPE FILE
                                %  THE VOLUME SET.
    VSSECTYPEF   = [47:2] #,    % SECURITYTYPE
    VSSECUSEF    = [45:2] #,    % SECURITYUSE

  % EVERY LINK WORD IS LAID OUT AS FOLLOWS
  VSLINKEDF = [47:1] #,        % 1 MEANS WORD HAS VSSIZEF & VSLINKF
  VSSIZEF  = [23:12] #,        %   LENGTH OF EACH ENTRY (WORDS)
  VSLINKF  = [11:12] #,        %   INDEX TO STUFF IN VARIABLE PART

VSMEMSNSX  = 16 #,             % LINK TO FAMILY SERIALNO LIST
                               %   SERIAL NUMBERS ARE STORED IN EBCDIC
VSMEMTYPSX = 17 #,             % LINK TO FAMILY STATUS LIST
  % FORMAT OF EACH FAMILY STATUS WORD IS:
     VSPERMF = [47:1] #,  %  1 IF USERCODE SHOULD RETAIN OWNERSHIP
     VSTRUSTF = [46:1] #, %  1 IF OK TO ALLOW DANGEROUS FILES TO
                          %    TO BE COPIED FROM TAPE.
     VSDSTRYF = [45:1] #, %  VOLUME DESTROYED
     VSKINDF  = [5:6] #,  %  TAPE, 7TRK, 9TRK, PE;  DISK, PACK
VSGUARDX = 18 #,               % LINK TO GUARDFILE TITLE OR Ø.
VSVARSTX = 19 #;

 REC [VSTYPEX ]  := VSVARSTX & 1 VSTYPEF
                    & 37 VSLEVELF & Ø VSLABELTYPEF
                    & 8"D" VSRECF;
 % GET DATE (YYDDD) AND CHANGE IT TO BINARY
 REC [VSDATEX ] := TIME (1Ø);
 REC [VSDATEX ] := INTEGER (POINTER (REC [VSDATEX])+1, 5);
 REC [VSCRTNSITEX ] := TIME (23).[23:16];   % SYSTEM SERIAL NUMBER
 REC [VSUSERX ] := Ø;                       % NO OWNER

 REPLACE POINTER (REC [VSFAMILYX ]) BY 48"Ø7" 8"SCRATCH";

 REC [VSSECURITYX ] := Ø & 3 VSSECTYPEF        % PRIVATE
                       & Ø VSSECUSEF;          % I/O

 REC [VSMEMSNSX ] := VSVARSTX & 1 VSSIZEF & 1 VSLINKEDF;
 REPLACE POINTER (REC [VSVARSTX ]) BY 5646 FOR 6 DIGITS;  % SERIALNO
 REC [VSMEMTYPSX ] := (VSVARSTX + 1) & 1 VSSIZEF & 1 VSLINKEDF;
 REC [VSVARSTX + 1] := 15 & 1 VSTRUSTF;
 REC [Ø] := Ø & 4"3F3F" MARKERF & (VSVARSTX+3) HDRBLOCKLENGTHF;
```

```
A[Ø] := REC [Ø].HDRBLOCKLENGTHF + 1;
REPLACE POINTER (A[1]) BY POINTER (REC)
          FOR REC [Ø].HDRBLOCKLENGTHF WORDS;

RSLT := SETSTATUS (4, 1, Ø, A);
```

VOLUME DELETE Call

The VOLUME DELETE call allows DCALGOL programs running under privileged usercodes to delete one or more volume serial numbers from a volume directory, but not from a volume library. This call can only be used if the TAPECHECK = AUTOMATIC option of the SECOPT (Security Options) system command has been designated on the system. This call is similar to the WFL *VOLUME DELETE* statement, except that the SETSTATUS call applies only to the volume directory and not to the volume library.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	4
SUBTYPE	0
VAL	0
ARRAYROW	A[0] = Length of the request. A[1] = 12 (tape). A[2] through A[4] = Volume family name in substandard form, such as 48"04" "TAPE". A[5] = Number of serial numbers specified, in the range 1 to 255, inclusive. A[6] through A[A[0]] = The serial numbers to be removed. All the serial numbers should belong to one volume family. Each serial number consists of 6 EBCDIC characters. Serial numbers that contain characters other than digits and are less than 6 characters long should be left-justified and filled with trailing blanks; serial numbers that are completely numeric should be right-justified and filled with leading EBCDIC zeros.

Call

```
RSLT := SETSTATUS (4, 1, Ø, A);
```

Results

Errors are returned in the normal manner.

For hard error number 42, either the value designated in A[0] or A[5] is incorrect or the number of characters designated for the name of the volume family is greater than 17.

The following soft errors can be returned:

Soft Error Number	Message
227	ERROR PROCESSING VOLUME DIRECTORY REQUEST (The delete request was incorrect, usually because a serial number you listed is not present in the volume directory or the volume name did not match the name in the directory.)
231	VOLUME DIRECTORY NOT ACTIVE

Example

For the WFL statement *VOLUME DELETE LMTAPE (SERIALNO = 123456, TAPE)*, use the following input:

```
A[Ø] := 7;
A[1] := 12;
REPLACE POINTER (A[2]) BY 48"Ø6" 8"LMTAPE";
A[5] := 1;
REPLACE POINTER (A[6]) BY 123456 FOR 6 DIGITS;
RSLT := SETSTATUS (4, Ø, Ø, A);
```

Section 12
SETSTATUS Request Type 5 (Unit Requests)

Request Type 5 calls are used to change the status of peripheral units or other hardware components. Most of these calls correspond to a system command; they are arranged alphabetically by name.

The following is a summary list of Request Type 5 calls:

Call Name	SUBTYPE	Description
AB	9	Turns on and off the automatic backup mark for card punch and printer units.
ACQUIRE	15	Allows the system to use selected hardware resources.
ADM	28	Starts or stops the automatic display of current system status information.
FORM	10	Assigns, modifies, or cancels a device according to the special forms designated by FORMID.
FREE	15	Prohibits the system from using certain hardware resources.
LB	11	Changes the family name, volume serial number, or name of the owner of a pack.
LH	21	Loads the pack controlware file to the designated disk pack controller for EMS systems.
MIRROR CREATE	25	Creates a mirrored disk set.
MIRROR OPTION	27	Designates recovery options for a mirrored set.
MIRROR RELEASE	26	Releases a pack from a mirrored set.
MODE	8	Informs the MCP that the write-enabled status of the unit has been changed.
MOVE	17	Moves a native-mode disk pack to another drive, even if the pack is in use.
PA	12	Creates or terminates associations between input and output devices.
PG and PGL	2	PG purges tape, disk pack, or host control units. PGL locks the tape units after they have been purged.
POWER	6	Powers disk packs up or down.

continued

continued

Call Name	SUBTYPE	Description
RC	6	Groups hardware resources to either the designated group or to the default designation.
REPLACE	22	Initiates a pack volume replacement operation.
RESTRICT DK, MT, or PK	33	Turns on or off security related restrictions for any unit.
RESTRICT ODT	5	Turns on or off security-related restrictions for the ODT.
RW	0	Rewinds, unloads, and locks magnetic tape units.
RY	1	Readies units, storage interface modules (SIMs), and memory subsystem modules (MSMs).
SCAN	23	Reads a pack or disk volume; analyzes and records read errors.
SEND	24	Sends a message that communicates with the image printer host support library, or network processor support library.
SN and SNL	2	Purges and assigns serial numbers to tape volumes and designates their recording density.
SR	5	Causes the system to reject all card decks without a USER statement.
SV	1	Saves units, SIMs, and MSMs.
TERM	29	Controls whether or not the designated ODT should behave like a data comm terminal and whether or not the end-of-text (ETX) character is to be used. If the ODT should behave like a data comm terminal, ETX should not be used.
UR	4 or 20	Reserves a unit for maintenance or releases a unit that was previously reserved for maintenance.

AB Call

SETSTATUS can still be called to turn on or off the auto backup mark for card punch and printer units. However, the form of the AB command used to establish the maximum count of printer and punch tasks can no longer be reached with a SETSTATUS call. That function has been taken over by the Print Router subsystem.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	9
VAL	0 = For *AB <unit>*
	1 = For *AB – <unit>*
ARRAYROW	A [0] = 3
	A [1] = 0 & 1 [38:6] & UTYPE [31:8]
	A [2] = Unit number

UTYPE is the numeric code for the card punch or printer device. For printers, UTYPE should be 6. For card punches, UTYPE should be 11. For ASID units, UTYPE should be 26. Refer to the table of hardware resource codes and unit type codes in Appendix C.

Call

```
RSLT := SETSTATUS (5, 9, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
4	PREVIOUSLY DONE (The unit was already designated for automatic backup and VAL = 1 in the call, or the unit was not automatically designated for backup VAL = 1 in the call.)
28	UNIT CORRESPONDENCE (UTYPE was not for a unit that can be used for automatic backup output.)
31	INVALID UNIT NUMBER (The unit number supplied in A [2] was out of range, or UTYPE did not match the kind of unit addressed.)

Example

For the system command *AB LP 7*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:6] & 6 [31:8];        % PRINTER CODE = 6
A[2] := 7;
RSLT := SETSTATUS (5, 9, Ø, A);         % AB CALL
```

ACQUIRE Call

Use the ACQUIRE call to let an active group acquire other resources. This call can be used to acquire peripheral units, controls, DLPs, HDUs, memory modules, central processors, DTUs, I/O processors (IOPs), ports, and RMMs.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	15
VAL	Bits that direct the action to be taken: • VAL.[0:1] = If bit 0 is turned off, the call is for ACQUIRE. If bit 0 contains a 1, the call is for FREE. • VAL.[1:1] = If bit 1 is turned on for ACQUIRE, the unit is to be acquired and reserved. • VAL.[2:1] = If bit 2 is turned on for ACQUIRE, the unit is to be acquired and saved. • VAL.[3:1] = If bit 3 is turned on for ACQUIRE, the resource is to be acquired with temporary status (it will be freed after the next halt/load). • VAL.[4:1] = If bit 4 is turned on for an ACQUIRE call for a peripheral unit, the unit is to be acquired with the OVERRIDE option.
ARRAYROW	Designates the unit or resource to be acquired or freed.

The encoding of the A parameter can differ because the required parameter information depends on the kind of resource to be acquired and on the kind of system. The following text summarizes the various formats of the parameter A.

Use the following values to ACQUIRE a peripheral unit such as a disk or a printer:

Array A	Value or Description
A [0]	3
A [1]	0 & 1 [38:6] & UTYPE [31:8]
A [2]	Unit number

UTYPE is a numeric code indicating the kind of unit to be acquired. Refer to Appendix C for a list of unit types.

Use the following values to acquire a base, a control, a data transfer unit (DTU), an MLI, or a port on an A 12 or an A 15 system:

Array A	Value or Description
A [0]	3
A [1]	0 & RESTYPE [31:8] & OVERRIDE [23:1]
A [2]	Contains the number of the base, control, DTU, MLI, or port

RESTYPE is a numeric code indicating the kind of hardware component to be acquired (refer to Appendix C for a list of hardware resource codes). OVERRIDE is 1 if the ACQUIRE of a DLP is to be done with OVERRIDE.

Use the following values to acquire a DLP on a system other than an A 12 or an A 15 system:

Array A	Value or Description
A [0]	2
A [1]	0 & 77 [31:8] & OVERRIDE [23:1] & PROCNO [22:3] & DLPNO [19:4] & BASENUM [15:16]

The number 77 is the DLP hardware code.

OVERRIDE is 1 if an ACQUIRE is to be done with OVERRIDE. PROCNO is the number of the processor to which the base is connected. DLPNO is the address of the DLP within the base. BASENUM is the base number through which the DLP is connected. BASENUM is formatted as follows:

```
BASENUM = Ø & "base bus address"      [13:6]
          & "base bus number"         [7:4]
          & "base extension number"   [3:4]
```

Use the following values to acquire a processor:

Array A	Value or Description
A [0]	2
A [1]	0 & 64 [31:8] & PROCID [15:16]

PROCID is the processor number.

Use the following values to acquire an RMM:

Array A	Value or Description
A [0]	2
A [1]	0 & 66 [31:8] & RMMID [23:8]

RMMID is the number of the RMM.

Call

```
RSLT := SETSTATUS (5, 15, VAL, A);
```

Results

SETSTATUS can return the following soft errors for ACQUIRE:

Soft Error Number	Description
28	UNIT CORRESPONDENCE (The hardware type code supplied in A [1].[31:8] does not match the hardware type code of the unit specified.)
31	INVALID UNIT NUMBER (The unit number supplied in A [2] is out of range, or the unit type code supplied in A [1].[31:8] is missing or or invalid or does not match the code stored in the MCP unit table. Soft error 31 is also produced if other numbers in the array parameter fields are invalid.)
42	INVALID MAIN FRAME TYPE (The hardware type code supplied in A [1].[31:8] is not valid.)
137	UNIT NOT ONLINE TO GROUP
180	REQUESTOR TESTING

Example

On an A 10 system, for the system command *ACQUIRE DLP dd BASE ba/bn/be PROC p*, use the following input:

```
A[Ø] := 2;
A[1] := Ø & 77 [31:8]
          & p [22:3] & dd [19:4]              % PROC NUM & DLP NUM
          & ba [13:6] & bn [7:4] & be [3:4]; % BASE NUMBER
RSLT := SETSTATUS (5, 15, Ø, A);              % ACQUIRE CALL
```

ADM Call

Use this call to start or stop the automatic display of current system status information. You must designate the unit number of the target ODT in the call.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	28
VAL	0 = ADM OK
	1 = ADM ST
ARRAYROW	A[0] = 3
	A[1] = 0 & 1 [38:06] & 2 [31:08]
	A[2] = Unit number of the ODT

Call

```
RSLT := SETSTATUS (5, 28, VAL, A);
```

Results

SETSTATUS does not return any unique soft errors for this call.

Example

For the system command *ADM OK* (and an ODT with a unit number of 129), use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:Ø6] & 2 [31:Ø8];
A[2] := 129;
RSLT := SETSTATUS (5, 28, Ø, A);
```

FORM Call

The FORM call corresponds to the PS CONFIGURE call. Use this call to assign, modify, or cancel the FORMID associated with an output peripheral unit. The quoted string is a string contained within quotation marks and can be up to 100 characters long.

This call corresponds to the following system commands:

```
PS CONFIGURE <device ID> FORMID = <quoted string>

PS CONFIGURE <device ID> - FORMID
```

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	10
VAL	0 = Assign FORMID configuration to a device
	–1 = Remove FORMID configuration from a device
ARRAYROW	A[0] = Pointer to the standard form message
	A[1].[38:06] = 1
	A[1].[31:08] = Unit type (refer to Appendix C)
	A[2] = Unit number
	A[3] through end = Message in standard form

Call

```
RSLT := SETSTATUS (5, 1Ø, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
24	UNIT IN USE
41	INVALID UNIT TYPE

Example

For the system command *FORM LP11 "MOUNT 3-PART"*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:Ø6] & 7 [31:Ø8];
A[2] := 11;
REPLACE POINTER (A[3], 8) BY 48"1ØØ1Ø1ØC" 8"MOUNT 3-PART";
RSLT := SETSTATUS (5, 1Ø, Ø, A);
```

FREE Call

Use the FREE call to detach resources from an active group. This call can be used to detach peripheral units, controls, DLPs, HDUs, memory modules, central processors, DTUs, IOPs, ports, and RMMs.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	15
VAL	[0:1] = Bit 0 must be turned on.
ARRAYROW	Designates the unit or resource to be freed.

The encoding of the A parameter can differ because it depends on the kind of resource to be freed and on the kind of system. The following text summarizes the various formats of the parameter A.

Use the following values to FREE a peripheral unit such as a disk or a printer.

Array A	Value or Description
A [0]	3
A [1]	0 & 1 [38:6] & UTYPE [31:8]
A [2]	Unit number

UTYPE is a numeric code indicating the kind of unit to be freed. Refer to Appendix C for a table of unit types.

Use the following values to free a base, a control, a DTU, an IOP, or a port on an A 12 or an A 15 system:

Array A	Value or Description
A [0]	3
A [1]	0 & RESTYPE [31:8] & OVERRIDE [23:1]
A [2]	Contains the number of the base, control, DTU, IOP, or port

RESTYPE is a numeric code indicating the kind of hardware component to be acquired or freed (refer to Appendix C for a list of hardware resource codes).

Use the following values to free a DLP on a system other than an A 12 or an A 15 system:

Array A	Value or Description
A [0]	2
A [1]	0 & 77 [31:8]

continued

continued

Array A	Value or Description
	& OVERRIDE [23:1] & PROCNO [22:3]
	& DLPNO [19:4] & BASENUM [15:16]

The number 77 is the DLP hardware code.

PROCNO is the number of the processor to which the base is connected. DLPNO is the address of the DLP within the base. BASENUM is the base number through which the DLP is connected. BASENUM is formatted as follows:

```
BASENUM = Ø & "base bus address"      [13:6]
          & "base bus number"         [7:4]
          & "base extension number"   [3:4]
```

Use the following values to free a processor:

Array A	Value or Description
A [0]	2
A [1]	0 & 64 [31:8] & PROCID [15:16]

PROCID is the processor number.

Use the following values to free an RMM:

Array A	Value or Description
A [0]	2
A [1]	0 & 66 [31:8] & RMMID [23:8]

RMMID is the number of the RMM.

Call

```
RSLT := SETSTATUS (5, 15, VAL, A);
```

Results

SETSTATUS can return the following soft errors for FREE:

Soft Error Number	Description
28	UNIT CORRESPONDENCE (The hardware type code supplied in A [1].[31:8] is not valid. For FREE commands, the unit number supplied in A [2] does not address an available unit.)
31	INVALID UNIT NUMBER (The unit number supplied in A [2] is out of range, or the unit type code supplied in A [1].[31:8] is missing or or invalid or does not match the code stored in the MCP unit table. Soft error 31 is also produced if other numbers in the array parameter fields are invalid.)

continued

continued

Soft Error Number	Description
39	ONLY ONE PROCESSOR
42	INVALID MAIN FRAME TYPE (The hardware type code supplied in A [1].[31:8] is not valid.)
150	UNIT NOT CLOSED
169	BAD EIO RESPONSE
172	UNIT HUNG
180	REQUESTOR TESTING

Example

For the system command *FREE PK 66*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:6] & 17 [31:8];    % DISK PACK CODE = 17
A[2] := 66;
RSLT := SETSTATUS (5, 15, 1, A);     % FREE CALL
```

LB Call

Use this call to change the family name, volume serial number, or name of the owner of a pack. You can also change a host control unit from SCRATCH to LABELED. You cannot change the volume serial number of a member of a multipack family. You cannot relabel a unit if the disk pack is not ready or has any files open when the command is issued.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	11
VAL	0
ARRAYROW	A[0] = Total length of input in words
	A[1].[38:06] = Length of the following entry in words, that is, A[0] – 2
	A[1].[31:08] = Unit type (refer to Appendix C)
	A[2] = Unit number
	A[3] through end = A list of attributes

The following characters relate to the data that starts in A [3]:

Character	Description or Value
1	Length of attribute list in characters (including this one)
2	0
3	Number of attributes
4 through end	Attribute assignments

The following is the format of attribute assignments:

Character	Description or Value
1	Length of attribute assignment in characters (not including the length character)
2	Attribute number: • NAME = 1 • SERIAL = 3 • OWNER = 7 • OLDNAME = 11
3 forward	Attribute value

Call

```
RSLT := SETSTATUS (5, 11, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
41	INVALID UNIT TYPE
167	INVALID FOR EXTERNAL IO UNITS
192	CANNOT LB MIRRORED UNIT

Example

For the system command *LB PK 46 NAME = ABCDEF SERIAL = 123456*, use the following input:

```
A[Ø] := 7;
A[1] := Ø &  5 [38:Ø6]
         & 17 [31:Ø8];
A[2] := 46;
REPLACE POINTER (A[3]) BY
     48"13" FOR 1,          % Total length in characters
     48"ØØ" FOR 1,
     48"Ø2" FOR 1,          % Two attributes
        48"Ø7Ø1" 8"ABCDEF"  % Len 7 Att 1 (Name) = ABCDEF
        48"Ø7Ø3" 8"123456"; % Len 7 Att 3 (Serial) = 123456

RSLT := SETSTATUS (5, 11, Ø, A);
```

LH Call (A 1, A 2, A 3, A 4, A 5, A 6, A 9, or A 10)

Use this call to load the pack controlware file to the designated disk pack controller.

If the specified path is not already reserved, it is automatically reserved before the controlware is loaded. If a file name is not designated, the default controlware file title is determined by the type of the disk pack controller.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	21
VAL	0
ARRAYROW	A[0] = 3
	A[1].[46:01] = 1
	A[1].[44:06] = Path ID
	A[1].[38:06] = 1
	A[1].[31:08] = Unit type (refer to Appendix C)
	A[2] = Unit number
	A[3] to end = Optional file name in standard form

Call

```
RSLT := SETSTATUS (5, 21, Ø, A);
```

Results

SETSTATUS can return soft error 167 ("INVALID FOR EXTERNAL IO UNITS") for this call.

Example

For the system command *LH PK 46 PATHID 1*, use the following input:

```
A[Ø] := 3;
A[1] := Ø &  1 [46:Ø1]
          &  1 [44:Ø6]
          &  1 [38:Ø6]
          & 17 [31:Ø8];
A[2] := 46;
RSLT := SETSTATUS (5, 21, Ø, A);
```

LH Call (A 12 or A 15)

Use this call to load a controlware file to a disk pack drive controller by way of the designated data link processor (DLP). The DLP must have been previously reserved.

If a file name is not designated, the default controlware file title is determined by the type of the disk pack controller.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	16
VAL	0
ARRAYROW	A[0] = 3
	A[1].[38:06] = 1
	A[1].[31:08] = Controller type
	A[2] = Controller number
	A[3] to end = Optional file name in standard form

Call

```
RSLT := SETSTATUS (5, 16, Ø, A);
```

Results

SETSTATUS can return soft error 167 ("INVALID FOR EXTERNAL IO UNITS") for this call.

Example

For the system command *LH DLP 4007*, use the following input:

```
A[Ø] := 3;
A[1] := Ø &  1 [38:Ø6]
        & 77 [31:Ø8];
A[2] := 4ØØ7;
RSLT := SETSTATUS (5, 16, Ø, A);
```

MIRROR CREATE Call

Use this call to create a mirrored disk set. The OP + MIRRORING option must be turned on before you can use this call.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	25
VAL	0
ARRAYROW	A[0] = 5
	A[1].[38:6] = 1
	A[1].[31:8] = 17
	A[2] = The unit number of the destination unit
	A[3].[38:6] = 1
	A[3].[31:8] = 17
	A[4] = The unit number of the source unit

Call

```
RSLT := SETSTATUS (5, 25, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
41	INVALID UNIT TYPE
58	NOT ALLOWED
195	DISK MIRRORING NOT YET SET
199	DESTINATION IS A MIRRORED UNIT

Example

For the system command *MIRROR CREATE PK44 FROM PK47*, use the following input:

```
A[Ø] := 5;
A[1] := Ø & 1 [38:6]
          & 1 [38:6];  % PK
A[2] := 44;
A[3] := Ø & 1 [38:6]
          & 1 [38:6];
A[4] := 47;
RSLT := SETSTATUS (5, 25, Ø, A);
```

MIRROR OPTION Call

Use this call to designate recovery options for any mirrored set. This option determines the action to be taken after a halt/load if certain critical MCP information has been lost.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	27
VAL	0 = Make mirror recovery option equal to DISCARD.
	1 = Make mirror recovery option equal to DMS.
ARRAYROW	A[0] = 3
	A[1].[38:06] = 1
	A[1].[31:08] = 17
	A[2] = Unit number

The number 17 is the unit type—refer to Appendix C.

Call

```
RSLT := SETSTATUS (5, 27, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
41	INVALID UNIT TYPE
113	INVALID ATTRIBUTE LIST
195	DISK MIRRORING NOT YET SET
198	DESTINATION IS A MIRRORED UNIT

Example

For the system command *MIRROR OPTION PK 47 RECOVERY=DMS*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1[38:6]
          & 17[31:8];  % PK
A[2] := 47;
RSLT := SETSTATUS (5, 27, 1, A);
```

MIRROR RELEASE Call

Use this call to release a pack from a mirrored set.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	26
VAL	0 = Release the specified pack.
	1 = Audit nonpresent copies of the specified pack.
	2 = Audit nonpresent copies of all partial mirrored sets.
	3 = Release offline copies of the specified pack (copies being audited).
	4 = Release all copies of the specified pack.
	5 = Ignore all copies of the specified pack (for partial mirrored sets).
ARRAYROW	A[0] = 3
	A[1].[38:6] = 1
	A[1].[31:8] = 17
	A[2] = The unit number

Call

```
RSLT := SETSTATUS (5, 26, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
41	INVALID UNIT TYPE
195	DISK MIRRORING NOT YET SET

Example

For the system command *MIRROR RELEASE PK 47*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:6]
          & 17 [31:8];
A[2] := 47;
RSLT := SETSTATUS (5, 26, Ø, A);
```

MODE Call

Use this call to tell the MCP that the write-enabled status of a unit has been changed.

For an HC unit, if a field for mode value or the hub index of a partner contains 0, no value is specified for that field. Therefore, that part of the specification for the unit is changed.

For an HC unit, if a field for the hub index of a partner contains a value greater than 16, there is no partner restriction (any existing restriction is removed).

This SETSTATUS call corresponds to the following system command:

```
MODE <unit type> <unit number> <mode value> <partner specs>
```

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	8
VAL	0
ARRAYROW	A[0] = 4
	A[1].[38:6] = 2
	A[1].[31:8] = The unit type (refer to Appendix C)
	A[2] = The unit number
	A[3].[46:1] = 0 (If unit type is not HC.)
	A[3].[46:1] = 1 (If unit type is HC.)
	A[3].[3:4] = 1 plus the mode value

If A[3].[46:1] = 1, the following applies:

```
A[3].[19:8]  = 1 + read partner's hub index
A[3].[11:8]  = 1 + write partner's hub index
```

Call

```
RSLT := SETSTATUS (5, 8, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
24	UNIT IN USE (An attempt was made to change the specifications of a designated hub controller.)
41	INVALID UNIT TYPE (The unit type is not disk, pack, magnetic tape, hub controller, or HYPERchannel.)
52	UNIT NOT AVAILABLE (The unit type is hub controller, and the specified unit is in an unacceptable state—for example, it is not labeled and not scratched, saved or assigned.)
173	MUST BE LABELED (An attempt was made to change the partner specification for a controller that is not labeled.)

Examples

For the system command *MODE PK 65 OUT*, use the following input:

```
A[Ø] := 4;
A[1] := Ø & 2 [38:6] & 17 [31:8];     % 17 = PK
A[2] := 65;
A[3] := VALUE(OUT) + 1;
RSLT := SETSTATUS (5, 8, Ø, A);
```

For the system command *MODE HC 123 IO WRITEPARTNER = 4 READPARTNER = 3*, use the following input:

```
A[Ø] := 4;
A[1] := Ø & 2 [38:6] & 2Ø [31:8];     % 2Ø = HC
A[2] := 123;
A[3] := (VALUE(IO) + 1) & 1 [46:1]    % hub controller
                        & 4 [19:8]    % 1 + read partner
                        & 5 [11:8];   % 1 + write partner
RSLT := SETSTATUS (5, 8, Ø, A);
```

MOVE Call

Use this call to move a native-mode disk pack to another drive, even if the pack is in use.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	17
VAL	0
ARRAYROW	A[0] = 5
	A[1].[38:6] = 1
	A[1].[31:8] = The unit type of the unit that contains the disk volume to be moved
	A[2] = The unit number of the unit that contains the disk volume to be moved
	A[3].[38:6] = 1
	A[3].[31:8] = The unit type of the destination unit
	A[4] = The unit number of the destination unit

Call

```
RSLT := SETSTATUS (5, 17, Ø, A);
```

Results

SETSTATUS can return soft error 167 ("INVALID FOR EXTERNAL IO UNITS") for this call.

Example

For the system command *MOVE PK 65 TO PK 73*, use the following input:

```
A[Ø] := 5;
A[1] := Ø & 1 [38:6] & 17 [31:8];      % 17 = PK
A[2] := 65;
A[3] := Ø & 1 [38:6] & 17 [31:8];      % 17 = PK
A[4] := 73;
RSLT := SETSTATUS (5, 17, Ø, A);
```

PA Call

Use this call to create or to terminate associations between input and output devices.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	12
VAL	0 = Delete all previous associations of a unit (example: PA– SC1). 1= Modify an existing association (example: PA SC1=+LP4, PA SC1=–LP4). 2 = Create a new association (example: PA SC1=LP4).
ARRAYROW	The information in the array A depends on VAL.

If VAL = 0, Array A is as follows:

Array A	Value or Description
A[0]	3
A[1].[46:8]	0
A[1].[38:6]	1
A[1].[31:8]	Unit type: • Input devices: – 2 for SC – 9 for CR • Output devices: – 7 for LP (train printer) – 11 for CP – 38 for LP (EBCDIC buffer printer)
A[2]	Unit number

If VAL = 1 or VAL = 2, use the following input for Array A:

Array A	Value or Description
A[0]	3
A[1].[46:8]	0
A[1].[38:6]	1
A[1].[31:8]	Unit type of input device (refer to Appendix C).
A[2]	Unit number of input device.
A[3]	3

continued

continued

Array A	Value or Description
A[4].[46:8]	2 = Modify an association by removing an associated output device.
	1 = Modify an association by adding an output device.
	0 = To create a new association.
A[4].[38:6]	1
A[4].[31:8]	Unit type of the output device (refer to Appendix C).
A[5]	Unit number of the output device.
A[6]	0

Call

```
RSLT := SETSTATUS (5, 12, VAL, A);
```

Results

Hard error 41 is returned for VAL 1 and 2 calls if A [0] + A [A[0]] is greater than or equal to SIZE (A).

To modify an association (VAL = 1), the association has to have been previously created. Otherwise, a SETSTATUS call with VAL = 1 will cause soft error 93 ("UNIT NOT ASSOCIATED").

If a soft error occurs, bit [47:1] of A[1] or A[4] will be turned on and [46:8] of A[1] or A[4] will contain the soft error number. Possible error numbers include the following:

Soft Error Number	Message
31	INVALID UNIT NUMBER
41	INVALID UNIT TYPE
93	NO PREVIOUS ASSOCIATION WAS MADE
137	UNIT NOT AVAILABLE

Examples

For the system command *PA – SC 13*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 2 [31:8] & 1 [38:6] & Ø [46:8];
A[2] := 13;
RSLT := SETSTATUS (5, 12, Ø, A);
```

For the system command *PA SC 12 = + LP 67*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 2 [31:8] & 1 [38:6] & Ø [46:8];
A[2] := 12;
A[3] := 3;
A[4] := Ø & 38 [31:8] & 1 [38:6] & 1 [46:8];
A[5] := 67;
A[6] := Ø;
RSLT := SETSTATUS (5, 12, 2, A);
```

PG and PGL Calls

Use the PG call to purge tape, disk pack, or host control units. Use the PGL call to lock the tape units after they have been purged (the unit type must be MT). PG is not valid for use with online mirrored packs. You must release the mirrored pack before the PG call can be issued.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	2
VAL	[0:1] = 1 if lock (PGL) [1:1] = 0 [15:8] = Density code: • 0 = Not specified (Density will not be changed.) • 1 = 800 bits per inch • 2 = 556 bits per inch • 3 = 200 bits per inch • 4 = 1600 bits per inch • 5 = 6250 bits per inch
ARRAYROW	A[0] = Length of request A[1] through A[A[0]–1] = Unit list

Each entry in the unit number list that requires from two to six words is as follows:

Array A	Message
A[n].[31:08]	Unit type: • 13 : MT • 17 : PK • 20 : HC
A[n].[38:06]	Number of additional words; must be 1 for MT or HC
A[n+1]	Unit number

For a PG PK request, the next few words of each entry in the unit number list are used to designate the OLDNAME. The first three bytes in the n + 2 word must be specified for PG PK requests even if there is no OLDNAME (see the example).

Array A	Message
A[n+2].[47:08]	Byte length from this byte to the end of entry
A[n+2].[39:08]	0
A[n+2].[31:08]	Number of specifications (0 or 1)
A[n+2].[23:08]	Byte length from the next byte to the end of the entry
A[n+2].[15:08]	11 (OLDNAME code)
A[n+2].[07:08]	Beginning of the family name
A[n+3] through end of entry	Continuation of the family name

Call

```
RSLT := SETSTATUS (5, 2, VAL, A);
```

Results

SETSTATUS might return one of the following soft errors for PG and PGL:

Soft Error Number	Message
14	THE REQUESTED UNIT IS LOCKED
41	INVALID UNIT TYPE
92	THE REQUIRED MCP LOCK IS IN USE
167	REQUEST IS NOT VALID FOR EIO
193	MIRRORED PACKS MUST BE RELEASED FIRST

Examples

For the system command *PG MT 30*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 13 [31:Ø8]          %% MT unit type
          &  1 [38:Ø6];         %% an additional word
A[2] := 3Ø;                     %% unit number
RSLT := SETSTATUS(5,2,Ø,A);
```

For the system command *PGL MT 160 (1600)*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 13 [31:Ø8]          %% MT unit type
          &  1 [38:Ø6];         %% an additional word
A[2] := 16Ø;                    %% unit number
RSLT := SETSTATUS (5, 2,
                   1 &            %% PGL
                   VALUE (BPI16ØØ) + 1 [15:8],  %% 16ØØ BPI
                   A);
```

For the system command *PG PK 55*, use the following input:

```
A[Ø] := 4;                      %% total words
A[1] := Ø & 17 [31:Ø8]          %% PK unit type
          &  2 [38:Ø6];         %% 2 additional words
A[2] := 55;                     %% unit number
REPLACE POINTER (A[3], 8) BY
    48"Ø3"                      %% total length
    48"ØØ"                      %% Ø
    48"ØØ";                     %% Ø (no specs)
RSLT := SETSTATUS (5, 2, Ø, A);
```

For the system command *PG PK 45 OLDNAME = USERPACK*, use the following input:

```
A[Ø] := 6;                      %% total words
A[1] := Ø & 17 [31:Ø8]          %% PK unit type
          &  4 [38:Ø6];         %% 4 additional words
A[2] := 45;                     %% unit number
REPLACE POINTER (A[3], 8) BY
    48"ØD"                      %% total length
    48"ØØ"                      %% Ø
    48"Ø1"                      %% one spec
    48"Ø9"                      %% length of spec
    48"ØB"                      %% OLDNAME
    8"USERPACK";                %% value
RSLT := SETSTATUS (5, 2, Ø, A);
```

POWER Call

Use this call to power disk packs either up or down.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	6
VAL	0 = For POWER OFF or POWER DOWN 6 = For POWER UP 7 = For POWER UP: OVERRIDE
ARRAYROW	A[0] = Length of request A[1] through A[(A[0]–1)] = Request list in the following format: • A[n].[38:6] = 1 • A[n].[31:8] = 17 • A[n+1] = Unit number

Call

```
RSLT := SETSTATUS (5, 6, VAL, A);
```

Results

SETSTATUS can return the following soft errors that refer to units:

Soft Error Number	Message
24	UNIT IN USE
165	UNIT/PATCH NOT POWERUP CAPABLE

SETSTATUS can return the following soft errors that refer to the system:

Soft Error Number	Message
11	ADDITIONAL INFO REQUIRED
186	SCP REPLIED: REQUEST NOT DONE

Example

For the system command *POWER DOWN PK44*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 17 [31:8]
          &  1 [38:6];
A[2] := 44;
RSLT := SETSTATUS (1, 6, Ø, A);
```

For the system command *POWER UP PK44*, use the following format:

```
A[Ø] := 3;
A[1] := Ø & 17 [31:8]
          & 1 [38:6];
A[2] := 44;
RSLT := SETSTATUS (1, 6, 6, A);
```

RC Call

Use this call to purge all files and to create a new set of volume labels on a disk pack.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	6
VAL	1
ARRAYROW	A[0] = Total length of input in words
	A[1].[38:06] = Length of the following entry in words, that is, A[0] – 2
	A[1].[31:08] = Unit type (refer to Appendix C)
	A[2] = Unit number
	A[3] through end = A list of attributes

The following characters relate to the data that starts in A [3]:

Character	Description or Value
1	Length of attribute list in binary characters (including this character)
2	0
3	Number of attributes
4 through end	Attribute assignments

The following is the format of attribute assignments:

Character	Description or Value
1	Length of attribute assignment in characters (not including the length character)
2	Attribute number: • NAME = 1 • INTERCHANGE = 2 • SERIAL = 3 (The serial number should be six EBCDIC digits.) • BP = 4 • OWNER = 7 • FAMILYINDEX = 9 • KEEP = 10 • OLDNAME = 11
3 forward	Attribute value in EBCDIC characters

Call

```
RSLT := SETSTATUS (5, 6, 1, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
41	INVALID UNIT TYPE
167	INVALID FOR EXTERNAL IO UNITS

Example

For the system command *RC PK 48*, use the following input:

```
A[Ø] := 4;
A[1] := Ø &  2 [38:Ø6]
          & 17 [31:Ø6];
A[2] := 48;
REPLACE POINTER (A[3]) BY
     48"Ø3ØØØØ";

RSLT := SETSTATUS (5, 6, 1, A);
```

REPLACE Call

Use this call to initiate a pack volume replacement operation.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	22
VAL	0
ARRAYROW	A[0] = 5
	A[1].[38:06] = 1
	A[1].[31:08] = Source unit type (refer to Appendix C)
	A[2] = Source unit number
	A[3].[38:06] = 1
	A[3].[31:08] = Destination unit type (refer to Appendix C)
	A[4] = Destination unit number
	A[5] = 0 for REPLACE
	A[5] = 1 for REPLACE and COMPARE

Call

```
RSLT := SETSTATUS (5, 22, Ø A);
```

Results

SETSTATUS checks to verify that the unit type and unit number designated are correct. If the parameters are correct, a waiting entry will appear in the mix. SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
31	INVALID UNIT NUMBER (This message is returned if the parameter type is neither INTERVAL nor IOINTERRUPT, or if a negative counterinterval is designated.)
41	INVALID UNIT TYPE
167	INVALID FOR EXTERNAL IO UNITS

Example

For the system command *REPLACE PK 46 ONTO PK 47*, use the following input:

```
A[Ø] := 5;
A[1] := Ø & 1 [38:Ø6] & 17 [31:Ø8];
A[2] := 46;
A[3] := Ø & 1 [38:Ø6] & 17 [31:Ø8];
A[4] := 47;
A[5] := Ø;                    % Replace only
RSLT := SETSTATUS (5, 22, Ø, A);
```

RESTRICT DK, MT, or PK Call

Use this call to turn on or off security-related restrictions for disks, packs, or tape units.

If security administrator status is authorized on the system, this call can be invoked only by a security administrator usercode. If security administrator status is not authorized, this call can be invoked by any privileged user.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	33
VAL	0 = Unrestrict access to disk, pack, or tape unit. 1 = Restrict access to disk, pack, or tape unit.
ARRAYROW	A[0] = Two times the number of entries, plus 1. A[1] through A[(A[0]–1)] = Request list in the following format: • i = 1, 2, 3, . . .(A([0]–1)/2 • A[2*i–1].[38:6] = 1 • A[2*i–1].[31:8] = 1 For DK unit type • A[2*i–1].[31:8] = 13 For MT unit type • A[2*i–1].[31:8] = 17 For PK unit type • A[2*i] = Unit number

Call

```
RSLT := SETSTATUS (5, 33, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
9	REQUEST DENIED
10	INVALID NUMBER
28	UNIT CORRESPONDENCE (The hardware type code supplied in A [1].[31:8] does not match the hardware type code of the unit specified.)
31	INVALID UNIT NUMBER (The unit number supplied in A [2] is out of range, or the unit type code supplied in A [1].[31:8] is missing or or invalid or does not match the code stored in the MCP unit table. Soft error 31 is also produced if other numbers in the array parameter fields are invalid.)

continued

continued

Soft Error Number	Message
137	UNIT NOT ONLINE TO GROUP
167	INVALID FOR EXTERNAL IO UNITS

Example

For the system command *RESTRICT MT 14, 15, PK 63*, use the following input:

```
A[Ø] := 7;
A[1] := Ø & 1 [38:6] & 13 [31:8];
A[2] := 14;
A[3] := Ø & 1 [38:6] & 13 [31:8];
A[4] := 15;
A[5] := Ø & 1 [38:6] & 17 [31:8];
A[6] := 63;
RSLT := SETSTATUS (5, 33, 1, A);
```

RESTRICT ODT Call

Use this call to turn on or off security-related restrictions for the ODT.

If security administrator status is authorized on the system, this call can be invoked only by a security administrator usercode. If security administrator status is not authorized, this call can be invoked by any privileged user.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	33
VAL	0 = Unrestrict access to ODT.
	1 = Restrict access to ODT.
ARRAYROW	A[0] = 6
	A[1].[38:6] = 1 plus the length of the security key in words.
	A[1].[31:8] = Unit type code (refer to Appendix C).
	A[2] = Unit number.
	A[3] = A maximum of three words, where A[3].[47:8] equals the length of the security key and the remainder contains the security key itself. The security key can be 1 to 17 alphanumeric characters.

Call

```
RSLT := SETSTATUS (5, 33, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
9	REQUEST DENIED
10	INVALID NUMBER
28	UNIT CORRESPONDENCE (The hardware type code supplied in A [1].[31:8] does not match the hardware type code of the unit specified.)
31	INVALID UNIT NUMBER (The unit number supplied in A [2] is out of range, or the unit type code supplied in A [1].[31:8] is missing or or invalid or does not match the code stored in the MCP unit table. Soft error 31 is also produced if other numbers in the array parameter fields are invalid.)

continued

continued

Soft Error Number	Message
137	UNIT NOT ONLINE TO GROUP
167	INVALID FOR EXTERNAL IO UNITS

Example

For the system command *RESTRICT SC 14 (SECURITYWORD)*, use the following input:

```
A[Ø] := 6;
A[1] := Ø & 4 [38:6] & 2 [31:8];
A[2] := 14;
REPLACE POINTER (A[3], 8) BY 48"ØC","SECURITYWORD";
RSLT := SETSTATUS (5, 33, 1, A);
```

RW Call

Use this call to rewind, unload, and lock magnetic tape units.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	0
VAL	0
ARRAYROW	A[0] = 3
	A[1].[38:06] = 1
	A[1].[31:08] = 13 % Unit type MT
	A[2] = The unit number

Call

```
RSLT := SETSTATUS (5, Ø, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
28	UNIT CORRESPONDENCE (The hardware type code supplied in A [1].[31:8] does not match the hardware type code of the unit specified.)
31	INVALID UNIT NUMBER (The unit number supplied in A [2] is out of range, or the unit type code supplied in A [1].[31:8] is missing or or invalid or does not match the code stored in the MCP unit table. Soft error 31 is also produced if other numbers in the array parameter fields are invalid.)
41	INVALID UNIT TYPE
137	UNIT NOT ONLINE TO GROUP
167	INVALID FOR EXTERNAL IO UNITS

Example

For the system command *RW MT 49*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:Ø6] & 13 [31:Ø8];
A[2] := 49;
RSLT := SETSTATUS (5, Ø, Ø, A);
```

RY Call

Use this call to ready peripheral units, central processors, MSMs, and storage interface modules (SIMs—A 12, A 15, and A 17). This call can contain more than one request.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	1
VAL	1
ARRAYROW	A[0] = Length of request A[1] to end = Unit number list. Each unit in the list needs a pair of words: • The first word of the pair is formatted as follows: 0 & 1 [38:06] & unit type code [31:08] (See Appendix C for a list of unit type codes.) • The second word of the pair is the external unit or identifying number of the unit, processor, or module.

Use the following fields to ready a SIM:

Array A	Value or Description
A [1].[38:06]	1
A [1].[31:08]	84 (SIM unit type code.)
A [2].[11:04]	Number of SIM.
A [2].[07:08]	Bit mask of MSUs.

Use the following fields to ready an MSM:

Array A	Value or Description
A [1].[38:06]	2
A [1].[31:08]	80 (MSM unit type code.)
A [2].[23:16]	Bit mask of pages.
A [2].[07:08]	Number of MSM.
A [3].[03:04]	Bit Mask of MSUs.

Call

```
RSLT := SETSTATUS (5, 1, 1, A);
```

Results

If the SIM or MSM variations are used on a system that does not support the request, the soft error 5 ("IT CANT BE DONE") is returned. Other possible soft errors are the following:

Soft Error Number	Message
4	PREVIOUSLY DONE
9	REQUEST DENIED
10	INVALID NUMBER
11	ADDITIONAL INFO NEEDED
28	UNIT CORRESPONDENCE
42	INVALID MAIN FRAME TYPE
82	INVALID STANDARD FORM
97	MODULE ALREADY IN USE
112	UNIT(S) TO BE READIED
167	INVALID FOR EXTERNAL IO UNITS
169	BAD EIO RESPONSE
172	UNIT HUNG

Examples

The following examples show RY requests equivalent to the following system commands:

```
RY MT 49
RY SIM 3 MSUS Ø-7
RY MSM 1 MSUS Ø-1
```

For the RY MT 49 system command, use the following:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:Ø6] & 13 [31:Ø8];
A[2] := 49;
RSLT := SETSTATUS (5, 1, 1, A);
```

For the RY SIM 3 MSUS 0–7 system command, use the following:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:Ø6] & 84 [31:Ø8];
A[2] := Ø & 3 [11:Ø4] & 4"FF" [Ø7:Ø8];
RSLT := SETSTATUS (5, 1, 1, A);
```

For the RY MSM 1 MSUS 0–1 system command, use the following:

```
A[Ø] := 4;
A[1] := Ø & 2 [38:Ø6] & 8Ø [31:Ø8];
A[2] := Ø & 4 "FFFF" [23:16] & 1 [Ø7:Ø8];
A[3] := Ø & 3 [31:32];
RSLT := SETSTATUS (5, 1, 1, A);
```

SCAN Call

Use this call to read a pack or disk volume and to analyze and record and read errors. It lets you determine, in advance, the results of issuing a REPLACE command that designates the given device as a source.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	23
VAL	0
ARRAY ROW	A[0] = 3
	A[1].[38:06] = 1
	A[1].[31:08] = Unit type
	(Refer to Appendix C for a list of unit type codes.)
	A[2] = Unit number

Call

```
RSLT := SETSTATUS (5, 23, Ø, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
41	INVALID UNIT TYPE
167	INVALID FOR EXTERNAL IO UNITS

Example

For the system command *SCAN PK 46*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:Ø6] & 17 [31:Ø8];
A[2] := 46;
RSLT := SETSTATUS (5, 23, Ø, A);
```

SEND Call

Use this call to send a message that communicates with the image printer (IP) support library, host support library, or the network processor support library.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	24
VAL	0
ARRAY ROW	A[0] = 3
	A[1].[38:06] = 0 & 1 [38:06] & unit type [31:08]
	(Refer to Appendix C for a list of unit type codes.)
	A[2] = Unit number
	A[3] = Count of characters in message
	A[4] through end = Message text

Call

```
RSLT := SETSTATUS (5, 24, Ø, A);
```

Results

The program sending the message receives no response.

SETSTATUS can return the following messages for this call:

Soft Error Number	Message
15	UNIT NOT READY
41	INVALID UNIT TYPE

Example

For the system command *SEND NP 231 LOAD*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:Ø6] & 27 [31:Ø8];
A[2] := 231;
A[3] := 4;
A[4] := "LOAD  ";
RSLT := SETSTATUS (5, 24, Ø, A);
```

SN and SNL Calls

Use these calls to purge and to assign serial numbers to tape volumes and to designate their recording density.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	2
VAL	[0:1] = 1 If lock (SNL) [1:1] = 1 [15:8] = Density code: • 0 = Not specified (Density will not be changed.) • 1 = 800 bits per inch • 2 = 556 bits per inch • 3 = 200 bits per inch • 4 = 1600 bits per inch • 5 = 6250 bits per inch
ARRAYROW	A[0] = Length of request A[1] = 0 & 2 [38:06] & unit type [31:08] (Refer to Appendix C for a list of unit type codes.) A[2] = Unit number A[3] = Serial number as 6 EBCDIC characters

Call

```
RSLT := SETSTATUS (5, 2, VAL, A);
```

Results

SETSTATUS does not return any unique errors for this call. The SN call shares most of the errors that any Request Type 5 call can return.

Example

For the system command *SN MT 49 UNISYS (1600)*, use the following input:

```
A[Ø] := 4;
A[1] := Ø & 2 [38:Ø6] & 13 [31:Ø8] & 4 [15:Ø8];
A[2] := 49;
A[3] := REPLACE POINTER (A[5]) BY 8"UNISYS";
RSLT := SETSTATUS (5, 2, 2 & (VALUE (BPI16ØØ)+1), A);
```

SR Call

Use this call to cause the system to reject all card decks entered in a card reader if those decks do not contain a USER statement.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	5
VAL	0 = SR–
	1 = SR
ARRAYROW	A[0] = Length of request
	A[1] = 0 & 1 [38:06] & unit type [31:08]
	(Refer to Appendix C for a list of unit type codes.)
	A[2] = Unit number

Call

```
RSLT := SETSTATUS (5, 5, VAL, A);
```

Results

The SR call shares most of the errors that any Request Type 5 call can return. In addition, it also returns the following unique calls:

Soft Error Number	Message
4	PREVIOUSLY DONE (If the unit is already secured by an SR call or is already unsecured by an SR– call.)
41	INVALID UNIT TYPE (If the unit address is not a card reader.)

Example

For the system command *SR CR 10*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:Ø6] & 9 [31:Ø8];
A[2] := 1Ø;
RSLT := SETSTATUS (5, 5, 1, A);
```

SV Call

Use this call to save peripheral units, central processors, MSMs and SIMs (A 12, A 15, and A 17). This call can contain more than one request.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	1
VAL	0
ARRAYROW	A[0] = Length of request A[1] to end = Unit number list. Each unit in the list needs a pair of words: • The first word of a pair is formatted as follows: 0 & 1 [38:06] & unit type code [31:08] (Refer to Appendix C for a list of unit type codes.) • The second word of a pair is the external unit or identifying number of the unit, processor, or module.

Use the following to save a SIM:

Array A	Value or Description
A [1].[38:06]	1
A [1].[31:08]	84
	(SIM unit type code)
A [2].[11:04]	Number of SIM
A [2].[07:08]	Bit mask of MSUs

Use the following to save an MSU:

Array A	Value or Description
A [1].[38:06]	2
A [1].[31:08]	80
	(MSU unit type code)
A [2].[23:16]	Bit mask of pages
A [2].[07:08]	Number of MSM
A [3].[03:04]	Bit mask of MSUs

Call

```
RSLT := SETSTATUS (5, 1, Ø, A);
```

Results

If the SIM or MSM variations are used on a system that does not support the request, the soft error 5 ("IT CANT BE DONE") is returned. Other possible soft errors are the following:

Soft Error Number	Message
4	PREVIOUSLY DONE
9	REQUEST DENIED
39	ONLY ONE PROCESSOR
42	INVALID MAIN FRAME TYPE
169	BAD EIO RESPONSE
172	UNIT IS HUNG

Example

For the system command *SV MT 49*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 1 [38:Ø6] & 13 [31:Ø8];
A[2] := 49;                              % SV MT 49
RSLT := SETSTATUS (5, 1, Ø, A);
```

TERM Call

This SETSTATUS call corresponds to the DCSTATION FALSE and DCSTATION TRUE options of the TERM system command, except that you must designate the unit number of the target ODT in the call.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	29
VAL	0 = For TERM DCSTATION FALSE
	1 = For TERM DCSTATION TRUE
ARRAYROW	A[0] = 3
	A[1] = 0 & 1 [38:06] & 2 [31:08]
	A[2] = Unit number of the ODT

Call

```
RSLT := SETSTATUS (5, 29, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
28	UNIT CORRESPONDENCE (The hardware type code supplied in A [1].[31:8] does not match the hardware type code of the unit specified.)
31	INVALID UNIT NUMBER (The unit number supplied in A [2] is out of range, or the unit type code supplied in A [1].[31:8] is missing or invalid or does not match the code stored in the MCP unit table. Soft error 31 is also produced if other numbers in the array parameter fields are invalid.)
41	INVALID UNIT TYPE
137	UNIT NOT ONLINE TO GROUP

Example

The following example shows the SETSTATUS call that is equivalent to the system command *TERM DCSTATION TRUE*. The target ODT has the unit number 129.

```
A[Ø] := 3;
A[1] := Ø & 1 [38:Ø6] & 2 [31:Ø8];
A[2] := 129;
RSLT := SETSTATUS (5,29,1,A);
```

UR Call

Use this call to reserve a unit so that maintenance can be performed or to make a previously reserved unit available for normal use. You cannot use this call for online mirrored packs.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	4 for a unit.
	20 for a path.
VAL	0 = UR–
	1 = UR
ARRAYROW	A[0] = Number of words in entry.
	A[1].[46:08] = If designated, is the unit path to be reserved. In this case, the SUBTYPE parameter should be 20.
	A[1].[38:06] = 1
	A[1].[38:06] = 0 for UR.
	A[1].[32:01] = 1 for UR with MAINT mode selected.
	A[1].[31:08] = Unit type.
	(Refer to Appendix C for a list of unit type codes.)
	A[2] = Unit number.

Call

```
RSLT := SETSTATUS (5, S, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
29	UNIT WAS NOT RESERVED (for UR–)
62	CANNOT RESERVE HALT/LOAD DISK
150	DISK MUST BE CLOSED TO UR IT
189	CANNOT RESERVE A MIRRORED DISK

Examples

For the system command *UR MT 30*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 13 [31:Ø8] & 1 [38:Ø6];
A[2] := 3Ø;
V := 1;
RSLT := SETSTATUS (5, 4, V, A);
```

For the system command *UR– LP 5*, use the following input:

```
A[Ø] := 3;
A[1] := Ø & 7 [31:Ø8] & 1 [38:Ø6];
A[2] := 5;
V := Ø;
RSLT := SETSTATUS (5, 4, V, A);
```

Appendix A
Hard Errors

Table A–1 lists the hard errors returned for SETSTATUS and GETSTATUS. In some cases the same error number can indicate different errors, depending on context. In these cases, all of the errors associated with the number are listed.

If bit [0:01] of the Boolean result from GETSTATUS or SETSTATUS is turned on and bits [11:08] of the result are not all 0, a hard error has occurred. A numerical error code for the error will appear in bits [11:08] of the Boolean result.

Table A–1. Hard Errors

Number	Explanation
10	The numbers or format of parameters in the array ARY is incorrect.
35	ARY[0].[19:20] is less than 2.
36	ARY[0].[19:20] is greater than or equal to the length of the ARY array.
37	Invalid value in TYPE.TYPEF.
38	For a GETSTATUS call, the SUBTYPE value in the TYPE.[15:08] parameter is invalid. For SETSTATUS calls, the SUBTYPE parameter is invalid.
39	This call is valid only on cataloging systems.
40	For a GETSTATUS call, SIZE (ARY) is greater than 21845 words or ARY[0].[19:20] is greater than 2047. For a SETSTATUS PA call (Request Type 5 SUBTYPE 12), adding the new information to the PA list would make the list more than 300 words long.
41	For a GETSTATUS call, complete information for first request could not be provided (insufficient space in array; A[0].[19:20] is too small). For a SETSTATUS call, the index to a name or string points beyond the end of the array ARY.
42	For a GETSTATUS call, ARY[0].[39:20] is greater than or equal to ARY[0].[19:20]. For SETSTATUS volume directory calls, the volume directory record provided is too large, the list of serial numbers is too long or too short, or the tape name provided is too long. For other SETSTATUS calls, the length of a string or name is designated as less than 0, or it extends beyond the end of the array ARY.
43	Invalid user of GETSTATUS intrinsic. For a SETSTATUS call, a standard form name passed in the array ARY is less than 5 bytes long.

continued

Table A–1. Hard Errors (cont.)

Number	Explanation
44	For a GETSTATUS call, SUBCLASS.ORGLEVELF is greater than the length of the first name found in the directory. For a SETSTATUS call, a standard form name passed in the array ARY is not correctly formatted.
45	SETSTATUS/GETSTATUS fault (and memory dump) occurred.
46	Last name of simple name request list has a standard form name format error, or there is not enough room in the array to return the fixed information for the file.
47	ARY[0].[19:20] is not large enough.
49	Insufficient space in array to insert <on part> identifier.
50	Not used.
51	TYPE DISPLAYFORMNAMEF is turned on, but TYPE.RETURNFULLNAMEF is not turned on.
52	Not used.
53	Not used.
54	For GETSTATUS call Request Type 0 SUBTYPE 5, both the MASK parameter and ARY [0].[39:20] are 0.
55	A[1] is less than 0 or A[1] is greater than or equal to the size of the MCP warning list (in words).
56	For a GETSTATUS call, the value of the SUBCLASS parameter is invalid.
57	The size of the array parameter is too small to contain all the information required for the call. Or the length of a structure or record in the array parameter extends beyond the end of the array parameter.
58	Word 0 of the array parameter does not have a valid value for the given call.
130	An error occurred while SETSTATUS was executing an archive directory request. The family might be offline, there might be no archive record for the requested file name, or the archive record might be incorrectly formatted.
131	Did not compile the MCP with $STATISTICS = TRUE.
179	The array in the MCP that contains the list of warnings to suppress cannot be expanded because it would become too large.
227	Error in processing a volume directory request.
231	Volume directory not active (SECOPT TAPECHECK value is NONE).

Appendix B
Soft Errors

Table B–1 lists the soft errors returned for SETSTATUS and GETSTATUS. In some cases the same error number can indicate different errors, depending on context. In these cases, all of the errors associated with the number are listed.

If bit [0:01] of the Boolean result from GETSTATUS or SETSTATUS is turned on and the field [11:08] is also 0, a soft error has occurred. When a soft error occurs, GETSTATUS or SETSTATUS stores one or more soft error words in the array ARY. A soft error word has bit [47:01] turned on, and the numerical soft error code appears in bits [46:08] of the word.

Table B–1. Soft Errors

Number	Message
0	INVALID ERROR (The requested SETSTATUS action will be done.)
1	IMPROPER STACK STATE
2	INVALID REPLY
3	NO REPLY NEEDED
4	PREVIOUSLY DONE
5	IT CANT BE DONE
6	Not used.
7	Not used.
8	INVALID FILE TYPE
9	REQUEST DENIED
10	INVALID NUMBER
11	ADDITIONAL INFO REQUIRED (The word count in field [38:06] of a word in the array parameter A is either 0 or is too small.)
12	IS A CODE FILE
13	HUB TITLE CONFLICT
14	UNIT LOCKED
15	UNIT NOT READY
16	WRITE PARITY
17	WRITE ERROR

continued

Table B–1. Soft Errors (cont.)

Number	Message
18	NOT A CODE FILE
20	REWIND ERROR
21	TEST FUNCTION
22	NO IO PATH
23	CODE ERROR (A mistake in the MCP code caused an unknown error indication in GETSTATUS or SETSTATUS.)
24	UNIT IN USE
25	UNIT RESERVED
26	UNIT SAVED
27	UNIT IN REWIND
28	UNIT CORRESPONDENCE
29	IS NOT RESERVED
31	INVALID UNIT NUMBER
32	INVALID SUB CLASS
35	MOD BEING SAVED
36	MODULE IN USE
37	CANNOT SAVE STACK VECTOR MOD
38	MODULE NOT ON LINE
39	ONLY ONE PROCESSOR
40	CANNOT SV TOP MOD
41	INVALID UNIT TYPE
42	INVALID MAIN FRAME TYPE
43	DISK READ ERROR
44	INVALID FACTOR TYPE
45	NAME REQUIRED
46	(This soft error number is reserved.)
47	(This soft error number is reserved.)
48	(This soft error number is reserved.)
49	NO FILE
50	(This soft error number is reserved.)
51	INVALID PACK NAME

continued

Table B–1. Soft Errors (cont.)

Number	Message
52	UNIT NOT AVAILABLE
53	(This soft error number is reserved.)
54	(This soft error number is reserved.)
55	FAMILY NAME REQUIRED
56	INVALID SYNTAX
57	CANNOT BE SNED
58	NOT ALLOWED (Either the call requested action or information that is not valid for this type of machine or the configuration in use, or the action or information is not valid for the unit specified in the call.)
59	NAME TOO LONG
60	INVALID PARAMETER INDEX
61	ALREADY RUNNING
62	CAN'T RESERVE H/L UNIT
63	PARAMETER LENGTH ERROR
64	INVALID DAY (Must be in the range of 1 to 28, 29, 30, or 31, depending on the month.)
65	INVALID YEAR (Must be at least 1900.)
66	(This soft error number is reserved.)
67	(This soft error number is reserved.)
68	(This soft error number is reserved.)
69	INVALID VALUE (The third parameter of the call is out of range for the given request type.)
70	(This soft error number is reserved).
71	NO BACKUP ON DK OR PK (SETSTATUS)
72	INVALID MONTH (Must be in the range of 1 to 12.)
73	UNKNOWN STATION
74	DATACOM INACTIVE
75	INACTIVE STATION
76	THREE ADDLWORDS REQUIRED (The value in the [38:06] field of word one of the array parameter is less than three.)
77	INCOMPLETE ENTRY (The value in word 0 of the array parameter is wrong, or a value in field [38:06] of a word in the array parameter is wrong.)

continued

Table B–1. Soft Errors (cont.)

Number	Message
78	INVALID STACK TYPE
79	INVISIBLE STACK
80	IS A COMPILER
81	FIVE CHARS REQ (The first byte of a standard form name is less than five.)
82	INVALID STANDARD FORM
84	MISSING CONTINUATION
85	INSUFFICIENT SPACE (There is not enough space in the array parameter. A name or string supplied in the array has a length indicating that it would extend beyond the size of the array, or the array is not large enough to receive all of the requested information.)
87	INVALID QUEUE HEAD
88	INV MCS
89	UNIT(S) TO BE SAVED
90	UNIT FAULT ERROR
91	MIX FAULT ERROR
92	REQUIRED LOCK IN USE, TRY AGAIN LATER
93	UNIT NOT ASSOCIATED
94	AUDIT IS NOT RUNNING
96	Not used.
97	MODULE ALREADY IN USE
98	NOT TEST JOB
99	'PU' AND 'PU TRANSPARENT' CANNOT BOTH BE SET
100	'SECADMIN' AND 'SECADMIN TRANSPARENT' CANNOT BOTH BE SET
101	WFL NOT INITIATED
102	NOT DSED, LIBRARY NOT RESUMABLE
103	(This soft error number is reserved.)
104	(This soft error number is reserved.)
105	NOT DSED, NETWORK SERVICES LIBRARY
106	NOT DSED, HOST SERVICES LIBRARY
107	MOD ALREADY ON LINE
108	MCM ALREADY ON LINE

continued

Table B–1. Soft Errors (cont.)

Number	Message
109	MCM IN USE
110	MCM NOT ON LINE
111	(This soft error number is reserved).
112	UNIT(S) TO BE READIED
113	INVALID ATTRIBUTE LIST
114	DYNAMIC MASK LINK ERROR (The value in A [0].[39:20] of the array parameter is greater than 1 but [38:06] in one of the list words in the array is greater than A [0].[39:20] – 2.)
115	NOT ENOUGH AVAILABLE DISK SPACE TO COPY
116	COPY ERRORS
117	NOT LOCKED
118	BAD LINK FIELD
119	SIMPLE NAME ERROR
120	FAMILY NOT ONLINE
123	SFN FORMAT
124	NO FILES FOUND UNDER DIRECTORY
125	NO VOLUME LIBRARY
126	SERIAL NUMBER NOT FOUND
127	BAD SECURITY BYTE (The second byte of a standard form name is invalid.)
128	(This soft error number is reserved.)
129	DISPLAY FORMAT (INVALID FILENAME)
129	PATH ALREADY RESERVED
130	CANT START STATISTICS (SETSTATUS)
130	STATISTICS ARRAY NOT PRESENT (GETSTATUS)
131	NOT A STATISTICS MCP
132	NOT WHILE NETWORKING
133	HOSTNAME IS REQUIRED
134	DUP FAMILY
135	PROGRAM IS LOCKED
137	UNIT NOT ONLINE TO GROUP
139	INVALID TIME

continued

Table B–1. Soft Errors (cont.)

Number	Message
140	INVALID DAY OF WEEK
141	INVALID TIME ZONE
142	INVALID OFFSET DIRECTION
143	INVALID OFFSET VALUE
150	UNIT MUST BE CLOSED
151	BAD PROCESSOR ID
152	MINIMUM PASSWORD LENGTH ENFORCED, REQUEST DENIED
160	HOST NAME NOT SET
167	INVALID FOR EXTERNAL IO UNITS
168	MESSAGE ARRAY OVERFLOW
169	INVALID EXTERNALIO RESPONSE
170	UNIT MUST BE SAVED
171	EXTERNAL IO HANDLER INACTIVE
172	UNIT IS HUNG
173	REQUIRES A LABELED UNIT
174	(This soft error number is reserved.)
175	INVALID PATH STATISTICS
176	INVALID FAMILY INDEX
177	Not used.
178	SEGARRAYSTART VALUE TOO SMALL
179	SEGARRAYSTART VALUE TOO LARGE
180	REQUESTOR INUSE FOR TESTING
181	HUB NUMBER TOO LARGE
182	DEFAULT HUBMAP FULL
183	PROCESSOR ID NOT ALLOWED
184	ERROR GETTING HL UNIT FROM SCP
185	SCP COULD NOT SET HL UNIT
186	SCP REPLIED: REQUEST NOT DONE
187	THAT FAMILY IS NOT CM'ED
188	UNIT MUST BE A SMD DISK
189	CANNOT UR A MIRRORED UNIT (SETSTATUS)

continued

Table B–1. Soft Errors (cont.)

Number	Message
189	INVALID HY ADAPTER (GETSTATUS)
190	CANNOT SV A MIRRORED UNIT (SETSTATUS)
190	HY READ STATISTICS ERR (GETSTATUS)
191	CANNOT RC A MIRRORED UNIT
192	CANNOT LB A MIRRORED UNIT
193	CANNOT PG A MIRRORED UNIT
194	DISK MIRRORING ALREADY SET
195	DISK MIRRORING NOT YET SET
196	MIRRORED SET(S) STILL EXIST
197	Not on this EMODE level. (NOT ON MCP/AS)
198	ERR DURING CONVERSION (SETSTATUS)
198	NOT MIRRORED UNIT (GETSTATUS)
199	DESTINATION IS A MIRRORED UNIT (SETSTATUS)
201	UNABLE PERFORM REQUEST
202	ADAPTER DOES NOT EXIST
203	ADAPTER IN USE BY BNA
204	ADAPTER IN USE BY HY FILE
205	DISK CACHE IS NOT RUNNING
206	UNIT IS ALREADY CACHEING
207	UNIT IS NOT CACHEING
208	NO CACHEING UNITS
209	ADAPTER MAY NOT BE CLEARED
210	ERROR CLEARING ADAPTER
211	NETEX TERMINATE IN PROGRESS
212	NETEX INITIALIZE IN PROGRESS
213	HY READ STATISTICS ERROR
214	CODEFILE NOT LIBRARY CAPABLE
215	CANNOT CHANGE A SYSTEM LIBRARY
216	FUNCTION WAS NOT ESTABLISHED
217	CAN'T BE MODIFIED WHILE IN USE
218	SL'ED FILE NOT PUBLIC

continued

Table B–1. Soft Errors (cont.)

Number	Message
219	MCP NOT COMPILE WITH MEMTRACE
220	VALUE(S) INCOMPATIBLE WITH SECURITY CLASS
221	REQUEST DENIED: SECADMIN USERCODE REQUIRED
222	XP FAILED: CODE FILE IS NOT A PROGRAM
223	XP FAILED: CODE FILE IS A SAFE PROGRAM
224	VOLUME DIRECTORY NOT READY YET
225	LIBRARY MARK LEVEL IS NOT EQUAL TO MCP MARKLEVEL
226	SECOPT TAPECHECK IS NOT AUTOMATIC
227	ERROR PROCESSING VOLUME DIRECTORY REQUEST
228	FILE NOT RESTRICTED: INCOMPATIBLE HEADER VERSION
229	SYSTEM HAS NO INFOGUARD AUTHORIZATION – HALT/LOAD TO INSTALL INFOGUARDSUPPORT
230	SYSTEM HAS NO INFOGUARD AUTHORIZATION – INFOGUARDSUPPORT LIBRARY NOT SL-ED
231	VOLUME DIRECTORY NOT ACTIVE
232	ARRAY TOO SMALL FOR VOLUME DIRECTORY RECORD
233	CANNOT CACHE SHARED DISK
234	UNIT IS IN ANOTHER PARTITION
235	THE DRC SYSTEM IS NOT ACTIVE
236	ATTEMPT TO EXCEED TEMPORARY FILE LIMIT
237	ATTEMPT TO EXCEED FAMILY LIMIT
238	FAMILY INTEGRAL LIMIT EXCEEDED
247	PREDICTIVE CACHE ALREADY RUNNING
248	UNIT IS ALREADY PREDICTIVE CACHEING
249	UNIT IS NOT PREDICTIVE CACHEING
250	NO PREDICTIVE CACHEING UNITS

continued

Table B–1. Soft Errors (cont.)

Number	Message
251	CODEFILE ALREADY ASSOCIATED WITH FUNCTION
252	LENGTH FIELD EXCEEDS SIZE OF INPUT ARRAY
254	SYSTEM HAS NO INFOGUARD AUTHORIZATION – INFOGUARDSUPPORT LIBRARY INITIALIZING
255	CODEFILE INCOMPATIBLE WITH THIS MCP VERSION

Appendix C
Hardware Resource Codes and Unit Type Codes

Table C–1 shows hardware resource codes and unit type codes (UNITTYPE). Code numbers 1 through 63 are I/O unit type codes. The remainder are codes for other kinds of hardware objects such as processors, modules, and interfaces.

Table C–1. Hardware Resource Codes and Unit Type Codes

Code Number	Description
1	DK, memory disk
2	SC or ODT
3	(Internal MCP code for REMOTE units)
4	CD, CD-ROM
5	Not used (formerly for paper tape punch)
6	LP, buffer printer
7	LP, train printer
8	HY, HYPERchannel adapter
9	CR, card reader
10	(Internal MCP code value)
11	CP, card punch
12	Not used
13	MT, 7-track tape
14	MT, 9-track tape
15	MT, phase-encoded (PE) tape
16	DC, data comm network support processors (NSPs) and frame recognition data link processors (FR DLPs)
17	PK, disk pack
18	UD, small computer system interface (SCSI) unit
19	PO, (internal MCP code value)
20	HC, host control unit—intersystem control (ISC)

continued

Table C–1. Hardware Resource Codes and Unit Type Codes (cont.)

Code Number	Description
21	FP, floating-point array processor
22	LP, printer
23	IP, image printer used as a line printer
24	SPC, special peripheral control
25	Not used (formerly for diskette)
26	IP, virtual static image device (VSID) DLP, also called ASID
27	NP, communications processor local area network DLP
34	IP, image printer used as a VSID
38	LP, printer
39	LP, printer
45	MT, magnetic tape
64	CPU or CPM, processor
65	MOD, memory module
66	RMM or IOM, resource management module (A 16 = IOM)
67	DTU, data transfer unit
68	Domain
69	System (A 16 = domain)
73	Exchange
75	Maintenance
76	Path ID
77	DLP, data link processor
78	MSM, memory storage module
79	Base
80	HDU, host data unit
81	MLI, message level interface
84	SIM, storage interface module
85	CTL, control
86	Port hardware on a large system
87	IOP, I/O processor

Appendix D
Example of a GETSTATUS Directory Call

The $SET INSTALLATION 1 card must appear before the first BEGIN statement in any program in which you include this procedure. This card in needed because this procedure uses the DISPLAYTOSTANDARD intrinsic.

```
$SET INSTALLATION 1


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% THIS PROCEDURE IS A SIMPLE BUT COMPLETE EXAMPLE %
% OF THE GETSTATUS DIRECTORY INTERFACE INCLUDING  %
% CONTINUATION CALLS AND ATTRIBUTE GATHERING.     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


REAL PROCEDURE DIRREQUEST(FILETITLE,ALLOC,SEGS,LASREC,CDATE,ADATE,
%              ----------
                          TSTAMP);
    ARRAY FILETITLE[Ø];
                   % THE ARRAY IS GIVEN THE INITIAL NODE
                   % FOR THE DIRECTORY AND RETURNS A FILE
                   % TITLE FOR EACH INVOCATION.
    REAL
         LASREC,   % THE LASTRECORD IN USE BY THIS FILE.
         ALLOC,    % THE SECTOR ALLOCATION FOR THIS FILE.
         SEGS,     % THE LAST SECTOR IN USE FOR THIS FILE.
         CDATE,    % THE CREATION DATE OF THE FILE.
         ADATE,    % THE ACCESS DATE OF THE FILE.
         TSTAMP;   % THE TIMESTAMP WORD OF THE FILE.
% PRAGMATICS:
%
%   THIS PROCEDURE MAY BE THE BASIS FOR UTILITIES
%   WHICH NEED DIRECTORY LISTS AND ATTRIBUTES OR
%   PROGRAMS WHICH READ DATA FROM AN ENTIRE NODE
%   OF SEPARATE FILES.
%
% USE:
%
%   CALL THE PROCEDURE PASSING AN ARRAY AND 5 REALS.
%   THE ARRAY SHOULD CONTAIN THE FULLY QUALIFIED NAME
%   OF A FILE INCLUDING THE "ON PART" AND TERMINATED
%   WITH A PERIOD.  THE ARRAY SHOULD BE LARGE ENOUGH
%   FOR THE LARGEST FILENAME POSSIBLE AND THIS ARRAY
%   MUST NOT BE MODIFIED BETWEEN CALLS.
```

Example of a GETSTATUS Directory Call

```
% VALUE:
%
%   THE PROCEDURE RETURNS THE FILE TITLE WITH "ON"
%   PART AND THE ATTRIBUTES OF THE FILE IN THE
%   REALS.  THE PROCEDURE IS SET TO A NONZERO
%   VALUE WHEN COMPLETE.  THE VALUE IS LESS THAN Ø IF
%   THE DIRECTORY IS EXHAUSTED AND MORE THAN Ø
%   (THE ERROR VALUE) IF AN ERROR WAS ENCOUNTERED.
%
BEGIN

OWN ARRAY STDFN[Ø:45], % STANDARD FORM NAME
          PKNARRY[Ø:3],% PACKNAME FOR ON PART
          GSA[Ø:1999]; % ARRAY FOR GETSTATUS
EBCDIC ARRAY EGSA[Ø] = GSA[*];
POINTER S,ST,FTP;
OWN REAL I;
REAL J,IX,T1,T2,
     FKIND,NEEDQUOTES;
BOOLEAN USERCODE,SYSTEMFILE,B;
TRUTHSET LEVELCHRS (ALPHA OR "-_");
OWN ARRAY FNAMEPOINTERS[1:14];
LABEL EXIT,CONTU;
DEFINE
       SUBTYPEF     =     [15:8]#,
       TYPEF        =     [7:8]#,
       MASK         =     1               %POINTER WORD
                          & 1 [7:1]       %ROWS IN USE
                          & 1 [1:1]       %CREATION DATE
                          & 1 [15:1]      %ACCESS DATE
                          & 1 [11:1]      %END OF FILE COUNT IN SEGS
                          & 1 [21:1]      %TIMESTAMP
                          & 1 [12:1]      % # OF BITS IN LAST SEG
                          & 1 [2:1]       %DISK BLOCKING
                          & 1 [5:1]       %ROW SIZE
                          & 1 [6:1]       %CRUNCH STATUS
                          & 1 [22:1]      %TOTAL SECTOR ALLOCATION
                          & 1 [41:1]      %FILELENGTH
                          & 1 [18:1]#,    %FILE MISC: FRAMESIZE,ETC.
       ERRORF       =     [47:1]#,
       ADDINFOF     =     [46:8]#,
       SUBVALUE1F   =     [35:3]#,
       SUBVALUE2F   =     [38:2]#,
       SUBVALUE3F   =     [36:1]#,
       LINKF        =     [32:17]#,
       LEVELF       =     [3:4]#,
       ONPARTLINKF  =     [43:11]#,
       INFOF        =     [15:16]#,
       EXTFRAMESZ   =     GSA[I+18].[23:8]#,
       BLKSZ        =     GSA[I+2].[47:16]#,
       MAXSZ        =     GSA[I+2].[15:16]#,
       DEFINEND     =     #;
```

```
FTP:=POINTER(FILETITLE);
IF I=Ø THEN
    BEGIN
    ST:=POINTER(STDFN);
    IF FTP="=" THEN
        REPLACE ST BY 48"Ø3ØØØØØØ"
    ELSE
    IF (B:=DISPLAYTOSTANDARD(FTP,ST)) THEN
        BEGIN
        DIRREQUEST:=REAL(B);
        GO TO EXIT;
        END;
    GSA[Ø]:=2ØØ;
    GSA[1]:=Ø;
    REPLACE POINTER(GSA[2ØØ]) BY ST FOR STDFN[Ø].[47:8];
    B:=GETSTATUS(Ø & 1 [42:1] & 3 TYPEF & 1 SUBTYPEF,Ø,MASK,GSA);
CONTU:
   IF B THEN
        BEGIN
        DIRREQUEST:=GSA[1].ADDINFOF;
        GO TO EXIT;
        END;
    END;
IF I=Ø THEN
   I:=2;
DO
    BEGIN
    IF I GEQ GSA[Ø] THEN
        BEGIN
        I:=Ø;
        IF GSA[Ø].ERRORF=1 THEN
            BEGIN % CONTINUATION
            GSA[Ø]:=2ØØ;
            B:=GETSTATUS(Ø & 1 [42:1] & 3 TYPEF & 4 SUBTYPEF,
                         Ø,MASK,GSA);
            GO TO CONTU;
            END
        ELSE
            DIRREQUEST:=-1;
        GO TO EXIT;
        END;
```

```
IX:=I;
IF GSA[I].LEVELF=Ø THEN
    I:=GSA[I].LINKF+1;
IF GSA[I].LEVELF=1 THEN % FIRST ENTRY
    BEGIN
    REPLACE POINTER(PKNARRY) BY " ON ",
        EGSA[(J:=GSA[GSA[1].ONPARTLINKF].LINKF)+1]
        FOR REAL(EGSA[J],1), ".";
    IF GSA[I].SUBVALUE1F=3 THEN
        USERCODE:=TRUE
    ELSE
        IF GSA[I].SUBVALUE1F=2 THEN
            SYSTEMFILE:=TRUE;
    END;
S:=FTP+(FNAMEPOINTERS[GSA[I].LEVELF]);
SCAN EGSA[(J:=GSA[I].LINKF)+1] FOR NEEDQUOTES:REAL(EGSA[J],1)
    WHILE IN LEVELCHRS;
IF USERCODE THEN
    REPLACE S:S BY "("
ELSE
IF SYSTEMFILE THEN
    REPLACE S:S BY "*";
IF FNAMEPOINTERS[GSA[I].LEVELF] GEQ 1 THEN
    IF S-1 NEQ ")" THEN
        REPLACE S-1 BY "/";
IF NEEDQUOTES ISNT Ø THEN
    REPLACE S:S BY """;
REPLACE S:S BY EGSA[J+1] FOR J:=REAL(EGSA[J],1);
IF NEEDQUOTES ISNT Ø THEN
    REPLACE S:S BY """;
IF USERCODE THEN
    REPLACE S:S BY ")"
ELSE
    REPLACE S:S BY "/";
IF USERCODE OR SYSTEMFILE THEN
    J:=*+2
ELSE
    J:=*+1;
IF NEEDQUOTES ISNT Ø THEN
    J:=*+2;
FNAMEPOINTERS[GSA[I].LEVELF+1]:=FNAMEPOINTERS[GSA[I].LEVELF]+J;
USERCODE:=SYSTEMFILE:=FALSE;
FKIND:=GSA[I].SUBVALUE2F;
```

```
      IF IX NEQ I THEN
          BEGIN
          SEGS:=GSA[I+11]+(IF GSA[I+12] > Ø THEN 1 ELSE Ø);
          ADATE:=GSA[I+15];
          CDATE:=GSA[I+1];
          TSTAMP:=GSA[I+21];
          IF BLKSZ = Ø THEN
               LASREC:=(GSA[I+41]+MAXSZ-1) DIV MAXSZ-1
          ELSE
               BEGIN
               % CHARACTERS PER SECTOR
               T1:=(48 DIV EXTFRAMESZ)*3Ø;
               % SECTORS PER BLOCK
               T1:=(BLKSZ+T1-1) DIV T1;
               % CALCULATE LASTRECORD
               T2:=GSA[I+11] MOD T1
                  * (48 DIV EXTFRAMESZ)*3Ø
                  + (GSA[I+12] + (EXTFRAMESZ - 1))
                  DIV (48 DIV (48 DIV EXTFRAMESZ));
               T2:=MIN(T2,BLKSZ)
                  + GSA[I+11] DIV T1 * (BLKSZ DIV MAXSZ) * MAXSZ;
               LASREC:=(T2+MAXSZ-1) DIV MAXSZ-1;
               END;
          ALLOC:=GSA[I+22];
          END;
        I : = IX + 1;
      END UNTIL FKIND NEQ 2;
    REPLACE S-1 BY POINTER (PKNARRY) UNTIL = 4"ØØ";
EXIT:
    END;
```

Glossary

In this glossary, definitions taken from outside sources are preceded by an abbreviation enclosed in parentheses. Definitions from *Dictionary of Computing* are preceded by DOC. Definitions from *American National Dictionary for Information Processing Systems* are preceded by ANDIPS. Definitions from *Vocabulary for Data Processing, Telecommunications, and Office Systems* are preceded by VDP.

A

active

Pertaining to the state of a process that is executing normally, and is neither scheduled nor suspended.

actual segment descriptor (ASD)

A pointer to the location of a data or code item in memory or on a disk.

adapter

(1) A hardware unit that connects a data communications line to a line support processor (LSP). An LSP can have up to 16 adapters, numbered from 0 to 15. A data communications data link processor (DCDLP) has four adapters, numbered from 0 to 3. (2) A Network Systems Corporation device that is connected to a HYPERchannel network and used for networking applications between computer systems of different manufacturers.

ADDLWORD

The extra parameter words that appear in the array passed to GETSTATUS and SETSTATUS for certain Request Types and SUBTYPEs. When applicable, the number of ADDLWORDs supplied for a particular request must be designated in bit [38:06] of the first word of the request. ADDLWORDs are required for certain calls, such as SETSTATUS *PG* (Purge) unit calls. They are optional for some calls, such as GETSTATUS *PD* (Print Directory) file calls, and are never used for other calls, such as SETSTATUS *DR* (Date Reset) calls.

address

(1) The identification of a location in storage (memory). (2) A sequence of bits, a character, or a group of characters that identifies a network station or a group of stations, a user, or an application. (3) The location of a device in the system configuration. (4) The identification of the location of a disk sector.

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.

algorithm
(1) A sequence of instructions describing the steps needed to complete a particular task. (2) In Network Definition Language II (NDLII), the part of a program that contains the adapter control and line control processes for one type of line and specifies the line protocol for that type of line.

APL
A Programming Language. A procedure-oriented language that can produce very short but powerful programs.

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

arithmetic expression
An expression containing any of the following: a numeric variable, a numeric elementary item, a numeric literal, identifiers and literals separated by arithmetic operators, two arithmetic expressions separated by an arithmetic operator, or an arithmetic expression enclosed in parentheses.

arithmetic operator
(1) (DOC) An operator that specifies an operation with numeric inputs and outputs; for example, ADD, SUBTRACT, or DIVIDE. (2) A single character or a fixed 2-character combination belonging to the following set: + (addition), – (subtraction), * (multiplication), / (division), or ** (exponentiation).

array
An ordered collection of a fixed number of common elements under one name, each element having the same data type. Access for each element is through an index to the common name.

ascending order
An arrangement of items in which the order progresses consecutively from the lowest-valued item to the highest-valued item. *Contrast with* descending order.

ASD
See actual segment descriptor.

ASD memory
The memory architecture used on A Series systems. Memory is treated as a single continuous region that is indexed by the ASD table. Memory management is very flexible and is handled automatically by the Master Control Program (MCP).

ASD number
A number used as an index to the actual segment descriptor (ASD) table. The ASD number is contained in code descriptors and data descriptors.

ASD table
A memory-resident table that contains the actual segment descriptors (ASDs) for the system.

attribute

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

autodial

The capability of a terminal, modem, computer, or similar device to place a call over the switched telephone network, and to establish a connection, without operator intervention.

B

binary

A characteristic or condition for which there are two alternatives. A binary number system uses a base of 2 and the digits 0 and 1.

bit

The most basic unit of computer information. The word *bit* is a contraction of *binary digit.* A bit can have one of two values: binary 0 (sometimes referred to as OFF) and binary 1 (sometimes referred to as ON).

bit rate

The speed at which bits are transmitted over a communication channel.

block

(1) A group of physically adjacent records that can be transferred to or from a physical device as a group. (2) A program, or a part of a program, that is treated by the processor as a discrete unit. Examples are a procedure in ALGOL, a procedure or function in Pascal, a subroutine or function in FORTRAN, or a complete COBOL program.

BNA

The network architecture used on A Series, B 1000, and V Series systems as well as CP 9500 and CP 2000 communications processors to connect multiple, independent, compatible computer systems into a network for distributed processing and resource sharing.

Boolean

Pertaining to variables, data items, and attributes having a value of TRUE or FALSE.

byte

(1) (ANDIPS) A binary character string operated upon as a unit and usually shorter than a computer word. (2) On Unisys A Series systems, a measurable group of 8 consecutive bits having a single usage. In data communications, a byte is often referred to as a character or an octet.

C

call

(1) To transmit addressing signals to establish a connection between stations. (2) In data communications, a sequence of events that begins when a user initiates a call request signal at an exchange that results in a connection. A call concludes when the connection

is released. Also, a call is an attempt to reach another network user, whether or not the connection is successful. (3) A programmatic request for another procedure or program to execute.

called program
A program that is the object of a CALL statement and is combined at object time with the calling program to produce a run unit.

calling program
A program that executes a CALL statement to another program.

CANDE
See Command and Edit.

catalog
(1) One of the sections in the catalog file. The catalog is a file that contains the following information for all files on the system: the available versions of a file, and the backup copies of the available versions of a file. (2) The central disk file that stores the information about the disks in the system and the disk files. This file is named SYSTEM/CATALOG/<family index number> on cataloging systems and SYSTEM/ACCESS/<family index number> on noncataloging systems.

CD-ROM
Compact disc read-only memory. A high-density read-only storage medium. The date is stored on a removable polycarbonate disk, and is read by a laser beam.

character
(1) The actual or coded representation of a digit, letter, or special symbol in display form. (2) In data communications, 8 contiguous bits (1 byte). (3) *See also* octet.

character array
In ALGOL, an array whose elements are ASCII, EBCDIC, or hexadecimal characters.

code segment dictionary
A memory structure that is associated with a process and that indexes the memory addresses of the various segments of program code used by that process. The same code segment dictionary can be shared by more than one process, provided that each process is an instance of the same procedure. A code segment dictionary is also referred to as a D1 stack.

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.

Communications Management System (COMS)
A general message control system (MCS) that controls online environments on A Series systems. COMS can support the processing of multiprogram transactions, single-station remote files, and multistation remote files.

compile
To convert a program written in a source language, such as COBOL or ALGOL, to machine code that can be executed by a computer.

compile time
The time during which a compiler analyzes program text and generates an object code file.

compiler
A computer program that translates instructions written in a source language, such as COBOL or ALGOL, into machine-executable object code.

COMS
See Communications Management System.

CONTROLLER
An invisible, independent runner program that is responsible for processing system commands, routing system messages, and scheduling jobs.

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 comm
See data communications.

data communications (data comm)
The transfer of data between a data source and a data sink (two computers, or a computer and a terminal) by way of one or more data links, according to appropriate protocols.

Data Communications ALGOL (DCALGOL)
A Unisys language based on ALGOL that contains extensions for writing message control system (MCS) programs and other specialized system programs.

data communications data link processor (DCDLP)
A data communications processor (DCP) that combines the functions of a network support processor (NSP) and a line support processor (LSP) into one physical data link processor (DLP) and supports up to four lines of communication.

data communications processor (DCP)
A hardware component that was replaced by the network support processor (NSP).

data link processor (DLP)
A processor that serves as the system interface to a specific peripheral device, controller, or communications network.

Data Management System (DMS)
The system responsible for storing and retrieving data while protecting data security and integrity.

database
An integrated, centralized system of data files and program utilities designed to support an application. The data sets and associated index structures are defined by a single description. Ideally, all the permanent data pertinent to a particular application resides in a single database. The database is considered a global entity that several applications can access and update concurrently.

database stack (DBS)
A stack that contains all the information necessary for the Data Management System II (DMSII) Accessroutines to manage a database.

DATACOMINFO file
A file that contains a complete description of the data communications configuration, including algorithms, editors, and translate tables. This is the file that the Interactive Datacomm Configurator (IDC) modifies and from which the Master Control Program (MCP) initializes the data communications subsystem.

DBS
See database stack.

DCALGOL
See Data Communications ALGOL.

DCDLP
See data communications data link processor.

DCP
See data communications processor.

DCWRITE
A system intrinsic that passes a specified message to the data communications controller (DCC).

declaration
A programming language construct used to identify an object, such as a type or variable, to the compiler. A declaration can be used to associate a data type with the object so that the object can be used in a program.

descending order
An arrangement of items in which the order progresses consecutively from the highest-valued item to the lowest-valued item. *Contrast with* ascending order.

directory
(1) A table of contents listing the files contained on a device. The device is usually a disk or a tape. (2) A list of file names organized into a hierarchy according to similarities in their names. File names are grouped in a directory if their first name constants (and associated usercodes) are identical. These groups are divided into subdirectories consisting of those file names whose first two name constants are identical, and so on.

discontinue
(1) To terminate a referenced task. (2) To cause a process to terminate abnormally. A process can be discontinued by operator commands, by statements in related processes, or by the system software.

disk
A random-access data storage device consisting of one or more circular platters that contain information recorded in concentric circular paths called tracks. Data on a disk are accessed by movable read/write heads. Some disks are removable. *Synonym for* disk pack, pack.

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.

disk header
See disk file header.

disk subsystem
Those software components of the system that are directly involved with the creation, maintenance, and deletion of disk files. This software also creates and maintains data structures such as disk labels, disk file headers, directories, and catalogs.

display form
A file title that contains one or more identifiers separated by slashes and that can include a usercode in parentheses, an asterisk, or an *ON <family name>* clause. The following is an example of a display form file title: (SMITH)REPORT/JULY ON ACCOUNTS.

distributed systems service (DSS)
One of a collection of services that are provided on Unisys hosts to support communications across multihost networks. DSS can be services such as file handling, station transfer, and mail transfer.

DLP
See data link processor.

DMS
See Data Management System.

DSS
See distributed systems service.

E

E-mode processor (EMP)
The type of processor used in A Series architecture.

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.

EBCDIC array
In ALGOL, an array whose elements are EBCDIC characters.

EIOU
See external input/output unit.

EMP
See E-mode processor.

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.

end of job (EOJ)
(1) The termination of processing of a job. (2) In the Communications Management System (COMS) and X.25, the control code that signals the receiver that a job has completed.

end of task (EOT)
The termination of processing of a task.

end of transmission (EOT)
A control code that tells the receiver that all user data (text) has been sent.

end-of-text character (ETX)
A keyboard character used to signal the end of input.

Entry and Medium Systems (EMS)
A designation referring to the Micro A and A 1 through A 10 systems.

EOF
See end of file.

EOJ
See end of job.

EOT
(1) *See* end of task. (2) *See* end of transmission.

ETX
See end-of-text character.

execution
The act of processing statements in a program.

external input/output unit (EIOU)
An I/O unit that has been added to the system by means of a generalized interface that causes the unit to be handled by a library instead of being managed directly by the MCP code file.

F

family
(1) One or more disks logically grouped and treated as a single entity by the system. Each family has a name, and all disks in the family must have been entered into the family with the RC (Reconfigure Disk) system command. (2) The name of the disk or disk pack on which a physical file is located.

family index
A 3-digit number the system assigns to a disk when the disk is added to a family. This family index value must be in the range 1 to 255, inclusive. The base pack is assigned the family index number, the first continuation pack is assigned 002, and so on. A family index is also referred to as a family index number.

family index number
See family index.

family name
(1) The name, consisting of up to 17 alphanumeric characters, assigned by an installation to identify a family of disks. (2) The name (label) of the disk or disk pack on which a physical file is located. The family name of a file is determined by the value of the FAMILYNAME file attribute. (3) The name of the logical group of disk packs on which a physical file is located. A family name consists of from 1 to 17 alphanumeric characters and is assigned by the installation.

field
A consecutive group of bits within a word or a component of a record that represents a logical piece of data.

file
(1) A named group of related records. (2) *See* logical file, physical file.

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

format
(1) The organization of an array of storage points in memory. Formats, and other memory structures, make it possible for the Master Control Program (MCP) to identify and move areas of memory. (2) The specific arrangement of a set of data.

function
(1) An assigned purpose, activity, or significance. (2) A subroutine that returns a value.

H

halt/load
A system-initialization procedure that temporarily halts the system and loads the operating system from a disk to main memory.

HC unit
See host control (HC) unit.

HDP
See host dependent port.

HDU
See host data unit.

hex
See hexadecimal.

hexadecimal (hex)
Pertaining to the base 16 numbering system. Decimal digits 0 through 9 are represented by the characters 0 through 9. Decimal digits 10 through 15 are represented by the characters A through F.

host
(1) An independent system in a BNA network. Each host has its own operating system and resources and is identified by a hostname. (2) The computer system containing the message control system (MCS) to which a given station is connected. (3) An I/O subsystem requestor, such as an I/O processor (IOP) or a network support processor (NSP). (4) The computer system that contains the database system with which a workstation interacts.

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.

host data unit (HDU)
The A 12 and A 15 system host interface to the I/O subsystem. An HDU is configured with up to three host dependent ports (HDPs), each of which supports two message level interface (MLI) cables.

host dependent port (HDP)
A hardware module capable of interfacing a processor to the message level interface (MLI).

HYPERchannel
A specialized data link processor (DLP) that enables communication between systems through HYPERchannel adapters. HYPERchannel is a message-level I/O channel-to-channel communications interface between A Series systems. It can also provide an interface to other systems for which a HYPERchannel adapter exists.

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 processor (IOP)

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

IDC

See Interactive Datacomm Configurator.

independent runner (IR)

A Master Control Program (MCP) procedure that is initiated as an independent process. The procedure is executed in its own process stack rather than in the stack of a user process. An independent runner (IR) can be either visible or invisible. If the IR is visible, its status can be interrogated. If the IR is invisible, it does not appear in mix displays.

index

(1) A value used to specify a particular element of an array variable. (2) A computer storage location, the contents of which identify a particular element in a table.

integer

A whole number.

Interactive Datacomm Configurator (IDC)

A Unisys interactive, menu-driven utility that enables the user to create, interrogate, and modify data communications network configurations.

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.

IOP

See I/O processor.

IR

See independent runner.

ISC

See intersystem control.

J

job

(1) A group of one or more tasks under the control of a single Work Flow Language (WFL) program. The system assigns each job a mix number and treats each job as a discrete unit of work. (2) *See* WFL job.

job file
A disk file that is associated with a job and contains the job log. The job file for a Work Flow Language (WFL) job also serves as the object code file for the job, and includes job restart information, data specifications, and a copy of the WFL source program.

job queue
A structure in the system software that stores a list of jobs that have been compiled and are waiting to be initiated.

L

LEM
See line expansion module.

library
(1) A collection of one or more named routines or entry points that are stored in a file and can be called by other programs. (2) A program that exports objects for use by user programs.

library maintenance
A procedure that copies disk files from a disk to a disk, from a disk to a tape, from a tape to a disk, and from a tape to a tape. The procedure is invoked by the Work Flow Language (WFL) ADD or COPY statements.

line
(1) A row of text in a printout. (2) A data transmission link between two computers or between a computer and its associated terminals. (3) In Network Definition Language II (NDLII), a logical construct representing a particular line adapter and all data structures associated with that line adapter.

line expansion module (LEM)
A hardware module that enables the attachment of up to seven bases to a single message level interface (MLI) port.

line support processor (LSP)
The data communications subsystem processor that manages communication with the host and initiates processes that control the input of messages to and the output of messages from data communications lines.

link
(1) In Data Management System II (DMSII), a field that enables one data set record to refer to another. (2) In the Communications Management System (COMS), to join the application program to COMS.

logging
The process of recording events and, often, their times of occurrence.

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 station number (LSN)
(1) In Network Definition Language II (NDLII), a unique number assigned to each station in a network. Each station has an LSN assigned according to the order in which the stations are defined in NDLII. The first defined station is 0000. (2) In the Interactive Datacomm Configurator (IDC), a unique number assigned to each station structure. When IDC creates the DATACOMINFO file from the Network Information File II (NIFII), it assigns an LSN to each structure sequentially, beginning with the number 2. The numbers allocated by IDC are the same as those used by the Master Control Program (MCP) to identify a station.

LSN
See logical station number.

LSP
See line support processor.

M

maintenance display terminal (MDT)
The name given to the system control terminal (SCT) when it is in maintenance mode. The MDT enables the operator to access the maintenance subsystem.

mapping
(1) A transformation from one set to another set. (2) A correspondence. (3) A description of the way in which different record types of a database are associated with one another.

Master Control Program (MCP)
An operating system on A Series systems. The MCP controls the operational environment of the system by performing job selection, memory management, peripheral management, virtual memory management, dynamic subroutine linkage, and logging of errors and system utilization.

MCP
See Master Control Program.

MCS
See message control system.

MDT
See maintenance display terminal.

memory
A temporary storage area where data and programs are placed while they are being processed.

memory storage unit (MSU)
The smallest unit of memory that can be readied or saved on an A Series system. An MSU contains 1 million to 4 million words, depending on the type of MSU. MSU types cannot be mixed on a system.

memory subsystem module (MSM)
A high-speed, random-access storage facility on host data unit (HDU) and resource management module (RMM) systems that contains storage boards and processing cards for the memory subsystem.

message
(1) Any combination of characters and symbols designed to communicate information from an originator to one or more destinations. (2) The text sent to the user from a program. A message can be either displayed on the screen or printed. (3) In data communications, any information-containing data unit, in an ordered format, sent by means of a communications process to a named network entity or interface. A message contains the information (text portion) and controls for routing and handling (header portion). (4) In Data Communications ALGOL (DCALGOL), a special form of array. Two types of messages are recognized by a message control system (MCS): those used with DCALGOL *DCWRITE* statements and those generated elsewhere in the data communications subsystem that appear in an MCS queue.

message area
In the Communications Management System (COMS), an area of the communication structure in which the message is contained.

message control indicator
A value used to select a type of output for a message.

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 level interface (MLI)
The interface between the host system, the I/O subsystem, and the data communications subsystem.

microcode
A sequence of elementary instructions that corresponds to a specific computer operator. For example, a sequence of microcode is executed for each A Series system operator.

mix
The set of processes that currently exist on a particular computer. The mix can include active, scheduled, and suspended processes.

mix number
A 4-digit number that identifies a process while it is executing. This number is stored in the MIXNUMBER task attribute.

MLI
See message level interface.

MLS
See multilingual system.

MSM

See memory subsystem module.

MSU

See memory storage unit.

multilingual system (MLS)

A system for developing and accessing output messages, online help text, and menu screens in different natural languages, such as English, French, and Spanish.

N

network information file (NIF)

See network information file II.

network information file II (NIFII)

The file generated when a Network Definition Language II (NDLII) program is compiled. This file contains line support processor (LSP) and network support processor (NSP) code, data structures, and other information. A NIFII is also generally referred to as a network information file (NIF).

network initialization file

A file that provides network configuration information to individual nodes, which enables the nodes to initialize themselves into a complete network by communicating with each other. This file is also referred to as the INIT file or NETINIT file.

network support processor (NSP)

A data communications subsystem processor that controls the interface between a host system and the data communications peripherals. The NSP executes the code generated by the Network Definition Language II (NDLII) compiler for line control and editor procedures. An NSP can also control line support processors (LSPs).

NIF

See network information file II.

NIFII

See network information file II.

node

(1) A data structure that consists of a list, a set of properties, and a block part. Either the list or the properties can be absent. (2) In a data communications network, a point at which one or more functional units interconnect with data transmission lines.

noncharacter array

A single dimension array whose size is measured in words, such as ARRAY A [0:10].

nonresident file

A file stored on a backup tape, or a backup copy of a file that is stored on a different disk family from that on which the primary copy of the file is stored. This term does not always pertain to the RESIDENT file attribute.

NSP

See network support processor.

O

octet

(1) In data communications, 8 contiguous bits (1 byte). (2) *See also* character.

ODT

See operator display terminal.

operator display terminal (ODT)

(1) A system control terminal (SCT) configured for direct communication with the operating system. The ODT is used primarily by operations personnel for entering commands that control and direct the system and its resources. (2) The name given to the system control terminal (SCT) when it is used as an ODT.

overlay

To load code or data into a memory area that was previously allocated to other code or data, and to write any data that previously occupied the area to a disk file if necessary.

P

p-bit

See presence bit.

pack (PK)

(1) A random-access data storage device consisting of one or more circular platters that contain information recorded in concentric circular paths called tracks. Data on a pack are accessed by movable read/write heads. Some packs are removable. (2) *Synonym for* disk pack, disk.

page

(1) A portion of a segmented array. (3) A structure in memory identified by unique address locations, or by subdivisions of a program running in memory.

paged array

An array that is automatically divided (*paged* or *segmented*) at run time into smaller segments.

parameter

(1) A quantity or item of information that can be given a different value each time a process is repeated. (2) An identifier associated in a special way with a procedure. A parameter is declared in the procedure heading and is automatically assigned a value when the procedure is invoked. (3) An object or value that is passed from an actual parameter and received by a formal parameter. (4) An element of a command, statement, or procedure that enables a user to determine the exact functionality of that command, statement, or procedure. A parameter can be variable or constant, and required or optional.

peripheral
(1) A device used for input, output, or file storage. Examples are magnetic tape drives, disk drives, printers, or operator display terminals (ODTs). (2) *Synonym for* peripheral device.

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

PK
See pack.

pointer
A variable with KIND equal to POINTER. A pointer is used to store a reference to a designated location in a designated array with a designated character size.

port
A point of entry into a computer, network, or other electronic device.

presence bit (p-bit)
A bit in a descriptor that indicates whether the address in the descriptor references a location in main memory or on a disk. If the presence bit is equal to 1, the address is in physical memory. If the presence bit is equal to 0, the address is either on a disk, or no memory or disk area is assigned for the descriptor yet. A p-bit is used only in the actual segment descriptor (ASD) table on ASD systems because of the memory architecture.

priority
A characteristic associated with a process that determines its precedence in the use of system resources. A process with higher priority executes more quickly than it would if it had lower priority.

privileged process
A process that bypasses normal system security checks, thereby gaining access to a large number of system resources. A process initiated by a privileged program or run under a privileged usercode is a privileged process.

privileged program
An object code file marked as privileged with the PP (Privileged Program) system command.

privileged user
A user with the PU usercode attribute assigned to his or her usercode in the USERDATAFILE. No file-access security checking is normally performed for actions taken under a usercode with privileged status.

procedure
A block that can be invoked by statements elsewhere in the same program or, in some cases, by statements in another program. In most instances, a procedure has a procedure heading and a procedure body. Examples are a procedure in ALGOL, a procedure or function in Pascal, a subroutine or function in FORTRAN, or a complete COBOL program.

process

(1) The execution of a program or of a procedure that was initiated. The process has its own process stack and process information block (PIB). It also has a code segment dictionary, which can be shared with other processes that are executions of the same procedure. (2) A software application; that is, any activity or systematic sequence of operations that produces a specified result.

programmable read-only memory (PROM)

A type of memory that can be modified once for specific purposes, and then can only be read.

PROM

See programmable read-only memory.

pseudostation

A station created by the Master Control Program (MCP) that can be attached to, and controlled by, a message control system (MCS) like a "real" station. Unlike a real station, however, a pseudostation is not declared in the SOURCENDLII file or the DATACOMINFO file, has no line assigned, and does not need a corresponding physical terminal on the local host.

Q

queue

A data structure used for storing objects; the objects are removed in the same order they are stored.

R

read

(1) The process of acquiring or interpreting data from an outside medium. (2) (ANDIPS) To acquire or to interpret data from a storage device, from a data medium, or from another source.

ready

In BNA, the condition that enables a station to receive data that is sent to it.

ready queue (READYQ)

A list, in order of priority, maintained by the Master Control Program (MCP), of the processes that are waiting for service from a processor.

READYQ

See ready queue.

real

In data management, pertaining to signed or unsigned, fractional or whole values in single-precision, floating-point form.

rebuild

(1) In the disk subsystem, a concept that refers to either of the following: a family rebuild, in which the system constructs the file access structure table (FAST) entry for a family by reading its system directory; or a catalog rebuild (on a cataloging system), in which the system updates the file access structure table (FAST) entry with information about cataloged files. (2) In database management, a recovery process in which the entire database is loaded from one or more sets of dump tapes. The recovery process then applies the audit trail after-update record images to move the database forward in time.

remote file

A file with the KIND attribute specified as REMOTE. A remote file enables object programs to communicate interactively with a terminal.

resident file

The primary copy of the file that is stored on a disk, regardless of whether or not the disk is online. Backup copies of files stored on another disk family are not considered resident. This term does not always pertain to the RESIDENT file attribute.

resource management module (RMM)

A hardware module that interfaces with the I/O subsystem and schedules tasks on the E-mode processor (EMP) by way of a message protocol.

RMM

See resource management module.

row

(1) The amount of contiguous disk space that is allocated at one time to a disk file as it is being created or expanded. The Master Control Program (MCP) uses the term *row* in its processing. (2) *See also* area.

run time

(1) The time during which an object code file is executed. (2) *Synonym for* execution time.

S

save memory

An area of memory that cannot be overlaid as long as the item with which it is associated is allocated.

save storage

See save memory.

SCT

See system control terminal.

sector

A subdivision of a track on a disk. A sector is the minimum addressable area on a disk pack. Unisys A Series system sectors are 30 words, or 180 bytes, long. *Synonym* for segment.

segment
A subdivision of a track on a disk. A segment is the minimum addressable area on a disk pack. Unisys A Series system segments are 30 words, or 180 bytes, long. *Synonym* for sector.

segment dictionary
See code segment dictionary.

segmented array
See paged array.

session
The interactions between a user and a message control system (MCS) during a particular period of time that is assigned an identifying session number. Logging on initiates a new session; logging off terminates a session. Each Menu-Assisted Resource Control (MARC) or Command and Edit (CANDE) dialogue at a terminal accesses a different session.

SIM
See storage interface module.

stack
A region of memory used to store data items in a particular order, usually on a last-in, first-out basis.

standard form file title
An internal form that the system uses to code file titles and other names. It is the result of a DISPLAYTOSTANDARD call.

station
(1) The outer end of a communication line. A station can correspond to a single terminal connected on a single line, or several stations can be connected on a line. (2) The combination of functional units comprising the data terminal equipment (DTE), data circuit-terminating equipment (DCE), and their common interface. (3) In the Interactive Datacomm Configurator (IDC), a data structure that describes the attributes of a physical terminal. (4) In data communications, a data structure that relates a logical abstraction to either a physical device or a pseudostation.

storage interface module (SIM)
A subdivision of memory storage on A 12 and A 15 systems. A SIM controls from one to four memory storage units (MSUs) on A 12 systems and from one to eight MSUs on A 15 systems.

subscript
A number that is an index into an array.

subscripted queue
A queue that is formed by subscripting a queue array and that is represented by a queue reference word accessed through one or more data descriptors.

suspended process
A process that has temporarily stopped executing and cannot continue until appropriate operator or programmatic action is taken. A process can be suspended deliberately by an

operator command or a statement in a program. In addition, the operating system can suspend a process automatically, for example, if the process has requested a file that is missing.

switched line
In data communications, a communications link for which the physical path, established by dialing, can vary with each use.

sync
See synchronous transmission.

synchronous transmission
In data communications, a mode of data transmission in which each bit of data is transmitted at a frequency determined by an external clock source.

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.

system control terminal (SCT)
A terminal used to enter information. An SCT can be used three ways: as an operator display terminal (ODT) to interface with the operating system, as a maintenance display terminal (MDT) to interface with the maintenance subsystem, or as a remote display terminal (RDT) to interface with remote support. The windows providing these uses are available once the automatic initialization sequence has finished.

system directory
(1) A special structure on each disk family that the system uses to locate files on that family. The system directory, also referred to as the flat directory, contains a disk file header for each permanent file in the family. (2) The logical directory for nonusercoded files.

system file
A file with a special security status that protects it from being removed, retitled, or replaced except by selected system interfaces. For example, the job description (JOBDESC) file can be removed only by the ??RJ (Remove JOBDESC File) system command.

T

tanking
In the transaction processing system (TPS), the operation in which the transaction processor (TP) library stores transactions in a tank journal and does not process them against the database. A tank journal can be any transaction journal except the TRHISTORY journal.

task

(1) A dependent process. (2) 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. (3) *See also* process.

task attribute

Any of a number of items that describe and control various aspects of the execution of a process. Various operator commands, message control system (MCS) commands, and programming language statements can be used to interrogate and modify the values of task attributes. The task attributes of a process are stored in the process information block (PIB).

task control processor (TCP)

A special purpose processor that schedules tasks on the E-mode processor (EMP) by way of a message protocol.

TCP

See task control processor.

terminal

An I/O device designed to receive or send source data in a network.

timestamp

An encoded, 48-bit numerical value for the time and date. Various timestamps are maintained by the system for each disk file. Timestamps note the time and date a file was created, last altered, and last accessed.

trunk

A cable connecting HYPERchannel adapters used to send data between computer systems of different manufacturers.

U

UINFO table

See unit information (UINFO) table.

unit information (UINFO) table

An internal data structure in the Master Control Program (MCP) code that maintains state information for every peripheral that is controlled directly by the MCP.

unit number

A number assigned by an installation to a peripheral device, such as a disk drive, and used to identify the device. The unit number commonly appears in conjunction with an acronym indicating the type of unit, which provides a unique identifier for a particular peripheral.

universal time (UT)

The time zone formerly known as Greenwich Mean Time.

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.

UT
See universal time.

V

vertical parity check
A parity check that verifies the validity of individual characters for a block.

volume
The medium of a mass storage device such as a disk, disk pack, or tape reel. The term *volume* is not restricted to the volume library on a cataloging system or the volume directory on a system with tape volume security. For example, on the BTOS family of workstations, the hard disk is a volume, and each floppy disk is a volume. When a volume is initialized, it is assigned a volume name and an optional password.

volume directory
A section of the catalog that tracks the status of tapes on a system that uses the tape security subsystem.

volume library
A section of the catalog that keeps track of all tapes and volumed disks used on a cataloging system.

W

WFL
See Work Flow Language.

WFL job
(1) A Work Flow Language (WFL) program, or the execution of such a program. (2) A collection of Work Flow Language (WFL) statements that enable the user to run programs or tasks.

word
A unit of computer memory. On A Series systems, a word consists of 48 bits used for storage plus tag bits used to indicate how the word is interpreted.

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.

write
(1) The process of transferring information to an output medium. (2) To record data in a storage device or location, or in a register.

Bibliography

A Series ALGOL Programming Reference Manual, Volume 1: Basic Implementation (form 8600 0098). Unisys Corporation.

A Series CANDE Operations Reference Manual (form 8600 1500). Unisys Corporation.

A Series Communications Management System (COMS) Operations Guide (form 8600 0833). Unisys Corporation.

A Series Communications Management System (COMS) Programming Guide (form 8600 0650). Unisys Corporation.

A Series DCALGOL Programming Reference Manual (form 8600 0841). Unisys Corporation.

A Series Disk Subsystem Administration and Operations Guide (form 8600 0668). Unisys Corporation.

A Series File Attributes Programming Reference Manual (form 8600 0064). Unisys Corporation. Formerly *A Series I/O Subsystem Programming Reference Manual.*

A Series I/O Subsystem Programming Guide (form 8600 0056). Unisys Corporation. Formerly *A Series I/O Subsystem Programming Reference Manual.*

A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide (form 8600 0288). Unisys Corporation.

A Series Security Administration Guide (form 8600 0973). Unisys Corporation.

A Series System Commands Operations Reference Manual (form 8600 0395). Unisys Corporation.

A Series SYSTEMSTATUS Programming Reference Manual (form 8600 0452). Unisys Corporation.

A Series Task Attributes Programming Reference Manual (form 8600 0502). Unisys Corporation. Formerly *A Series Work Flow Administration and Programming Guide.*

A Series Work Flow Language (WFL) Programming Reference Manual (form 8600 1047). Unisys Corporation.

American National Dictionary for Information Processing Systems (technical report). American National Standards Committee X3, Information Processing Systems. Washington, DC: Computer and Business Equipment Manufacturers Association (CBEMA), 1982.

Dictionary of Computing. Frank J. Galland (ed.). New York: John Wiley & Sons, 1982.

Vocabulary for Data Processing, Telecommunications, and Office Systems (form GC10-1699-6). Poughkeepsie, New York: International Business Machines Corporation, 1981.

Index

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

86000346-000