UOS System Services Reference

Table of contents

UOS System Services Reference

UOS System Services Reference

October 2023

Contents

Contents

Preface

Preface

Intended Audience

This manual is intended for all software developers writing software for the UOS operating system.

System Services Descriptions

System Service Descriptions

System services provide basic UOS system functions, communications, and resource management.

Condition values returned by system services indicate whether the request completed successfully, or a specific error code. Some error codes can indicate that some, but not all, of the requested functions completed.

All system calls pass a pointer to a structure that indicates the system call being made. The beginning of the structure is the same for all system calls, but the portion of the structure after the header differs depending upon the call being made. The mechanism for making the call and passing the structure is platform-specific.

The runtime libraries (such as SYS and LIB) offer a simple interface for system calls that allow for a function call that constructs the appropriate structure and makes the call. Applications can use either method of making a system call, and both the funciton interface and system call structure are listed for each service.

Offset byte	Length in bytes	Description
0	4	Subsystem
4	4	Request ID
8	2	Length of call-specific following data, in bytes
10	8	Status (ignored on call, result status on return)

18	8	Call-specific flags
26	n	Call-specific data

The Subsystem indicates which UOS Executive component the request is directed to:

Subsystem	Value
MMC	1
FIP	2
USC	3
SSC	4
Kernel	5

ASCTIM

ASCTIM

Convert Binary Time to ASCII

This is a wrapper for the Starlet routine LIB$SYS_ASCTIM. See the documentation for the Starlet Library.

CLOSE

CLOSE

Close File

The CLOSE service disposes of a handle to a file, which ends file access via that handle.

Format

SYS$CLOSE fab err suc

Arguments

fab

The 64-bit address of a FAB block.

err

The 64-bit address of a callback routine that is called if an error occurs during the system call.

suc

The 64-bit address of a callback routine that is called when the operation completes.

System Request Structure

Field	Value
Subsystem	FIP
Request ID	UOS_FIP_Close
Length	24
Status	
Flags	Ignored
Integer1	Pointer to FAB control block

Integer2	Pointer to error completion routine
Integer3	Pointer to success completion routine

Description
When successfully invoked, the CLOSE service closes the file handle and may update various timestamps or perform other operations associated with closing the file. The associated file handle must have been set by a successful OPEN service call. Once complete, the file handle is no longer valid and attempts to use that handle will result in errors. However, it is possible that another OPEN service call may allocate a new handle with the same value.

Quotas Affected
FILLM

Privileges Required
None.

Condition Codes Returned

RMS$_NORMAL	Successful completion.
UOSErr_Missing_Value	No FAB address was supplied.
RMS$_FAB	FAB was incorrect format (size or code).

CREATESTREAM

CREATESTREAM
Create a stream

Creates an ancillary data stream on an open file.

Format
SYS$CREATESTREAM FAB Name Res

Arguments

FAB
An address of a File Access Block structure for an open file.

Name
An address of a SRB that points to the name for the newly created stream. Leading and trailing spaces are trimmed from this name. The name may not be null, and it must follow the rules for file names on the file system where the file is located. If a stream of the same name exists (case is not significant), an error occurs.

Res
An address of a 64-bit integer where the stream index of the newly created stream is to be written. If an error occurs, this value will be 0.

Description
This service creates an ancillary data stream for a file. The file must be currently open.

Condition codes returned

Code	Meaning
SS_NORMAL	Normal completion.

SS_ACCVIO	Memory access violation.
SS_DUPLNAM	A stream with that name already exists.
SS_NOPRIV	User does not have privileges or permission to modify the file.
SS_EXQUOTA	Operation cannot complete due to quotas.
RMS_FAB	The FAB block has an invalid layout.
RMS_BLN	The FAB block has an invalid length.
LIB_INVOPER	Operation attempted on a non-file resource.
UOSErr_Invalid_Handle	The FAB does not contain an valid file handle.
UOSErr_Invalid_Resource_Name	The name is null or contains characters that are not valid for a filename on the file system.

CRELNT

CRELNT
Create Symbol Table

Creates a process-private or public-sharable symbol table.

Format

SYS$CRELNT(Attr, Resnam, Reslen, Quota, Promsk, Tabnam, Partab, AcMode)

Arguments

Attr

Address of the attribute flags to apply to the newly created table. If the address is 0 or points to an integer 0 value, no flags are associated with the table. The valid flags are:

Flag	Description
LNM_M_CREATE_IF	Create the table only if it does not exist. If not specified, and a table with the same name and access mode already exists, it is deleted and a new table is created.
LNM_M_MAKE_DEFAULT	Applies only to process-private tables. If specified, the new table is made the new default (lnm$process).

Resnam

Address where to write the new table name. If 0, the name isn't written.

Reslen

Address of a 64-bit integer to receive the length of the result name.

Quota

Quota for the new table, indicating the limit of the table contents in bytes. 0 indicates no quota.

Promsk

Protection mask for the new table.

Tabnam

Address of a TSRB structure specifying the new table name. If this is 0 or points to a TSRB of a null string, the table name is created by UOS with a unique name of "lnm$x" where "x" is a hexadecimal value.

Partab

Address of a TSRB structure specifying the new table's parent. If this is "lnm$process_directory" the new table will be private to the current process.

AcMode

Address of the access mode of the new table. If not specified (the address is 0), the process' current access mode is used.

Description

This system service creates a symbol table. The SYSNAM privilege is required to create a table at an access mode more privileged than that of the calling process.

Condition Codes

Code	Meaning
SS_NORMAL	Normal completion
SS_ACCVIO	Could not access the memory associated with one of the parameters
SS_BADPARAM	An invalid parameter was passed
SS_NOLOGTAB	The specified parent table wasn't found
SS_NOPRIV	User lacked the privilege to perform the operation
SS_RESULTOVF	The table was created, but the result name location was specified and was too small to contain the full name
SS_SUPERSEDE	Normal completion. Existing table was superseded by the new table

CREPRC

CREPRC

Create a Process

Creates a subprocess or detached process on behalf of the current process.

Format

SYS$CREPRC pidadr, image, input, output, error, prvadr, quota, prcnam, baspri, uic, mbxunt, stsflg, itmlst, node, home_rad

Arguments

pidadr

A pointer to memory to receive the 32-bit PID of the create process. If 0, the new PID is not returned.

image

A pointer to an SRC structure that points to the name of the image to run. If this is 0, or points to a null string, or the specified image is not found, the default system shell is used.

input

A pointer to an SRC structure that points to the name of the file to assign to SYS$INPUT for the created process. If not specified or null, the same SYS$INPUT for the current process is used for the new process.

output

A pointer to an SRC structure that points to the name of the file to assign to SYS$OUTPUT for the created process. If not specified or null, the same SYS$OUTPUT for the current process is used for the new process.

error

A pointer to an SRC structure that points to the name of the file to assign to SYS$ERROR for the created process. If not specified or null, the same SYS$ERROR for the current process is used for the new process.

prvadr

A pointer to a 64-bit privilege mask to assign to the created process as the current privilege mask. If the address is 0, the default privilege mask for the user is assigned to the created process. Unless the calling process has the SETPRV privilege, the mask is restricted to the privileges granted to the calling process. If the privileges granted to the new process do not match what was requested, no error is

returned.

quota
A pointer to a quota block list. Each quota block consists of a one-byte quota index followed by a four-byte value. The list terminates with a quota index of PQL_LISTEND. If the list is empty or the address is 0, the new process receives the default quotas. If a value is less than the system minimum value for that quota, the minimum value is used. Values of 0 indicate no limit for that quota. If the value is non-zero and larger than the same quota for the current process, and the current process doesn't have IMPERSONATE or CMKRNL privileges, the quota is limited to that of the current process. No errors are returned if a quota is assigned a value other that what is specified. The quota indexes are:

Mnuemonic	Description
PQL_ASTLM	Max outstanding ASTs.
PQL_BIOLM	Max buffered I/O requests.
PQL_BYTLM	Max buffered I/O bytes.
PQL_CPULM	Max CPU seconds per session (in 10 ms units)
PQL_DIOLM	Direct (raw) I/O requests.
PQL_ENQLM	Max number of simultaneous lock requests.
PQL_FILLM	Max number of simultaneous open files.
PQL_JTQUOTA	Max size of Job table, in bytes.
PQL_PGFLQUOTA	Max pages that can be used by the process in page file.
PQL_PRCLM	Subprocess limit.
PQL_THREADLM	Max simultaneous thread limit.
PQL_TQELM	Max simultaneous Timers quota.
PQL_WSDEFAULT	Working set default size.
PQL_WSEXTENT	Max Working Set expansion quota in pages.
PQL_WSQUOTA	Maximum locked pages.

prcnam
A pointer to an SRB that points to the name to assign to the new process. If 0 or null, the name will be the user name, an underscore ("_") and a number. This number is determined by the PRC_M_NONRANDOM flag. If not set, the number is randomly generated and guaranteed to not match another process on the node. If the flag is set, the number is set to "1" unless there is already a matching process name. If there is a matching name, the number is set to "2", and so on. This process continues until a unique process name is generated.

baspri
A pointer to a byte containing the base priority for the new process. If 0, the current process base priority is used. Unless the calling process has the ALTPRI privilege, the specified priority is not allowed to exceed the priority of the calling process and is limited to the current process priority. If the priority actually assigned is not the same as what was specified, no error is returned.

uic
A pointer to a 64-bit User Identification Code. If 0, or it points to a value of 0, the current user's UIC is used. If a UIC other than the current user's UIC is passed, the process must have the IMPERSONATE or CMKRNL privilege or an protection violation results. If the process has the appropriate privilege and and different UIC is specified, the process is created as a detached process for that user.

mbxunt
Mailbox unit number to receive termination notice when process ends. If this is 0 or points to a value of

0, no termination notice is sent.

stsflg
A pointer to a 64-bit integer containing options for creating the process. The valid flags are:

Mnuemonic	Description
PRC_M_BATCH	Create a batch process. The IMPERSONATE privilege is required. This option is only intended for use with system programs.
PRC_M_DISAWS	Disable system-initiated working set adjustment.
PRC_M_HIBER	Force process to hibernate before it executes the image.
PRC_M_HOME_RAD	Assign process to specified home resource affinity domain (RAD).
PRC_M_IMGDMP	Create an image dump if the image abends. The dump file name will be the image name with a ".dmp" file extension.
PRC_M_IMPERSONATE	Create a detached process under another UIC.
PRC_M_INTER	Create an interactive process. This option is only intended for use with system programs.
PRC_M_NETWRK	Create a process that is a network connect object. The IMPERSONATE privilege is required. This option is only intended for use with system programs.
PRC_M_NOACNT	Do not perform accounting. The ACNT privilege is required.
PRC_M_NOPASSWORD	Don't prompt for authentication if running sys$system:login.exe. The new process is logged in under the user of the current process. This option is only intended for use with system programs.
PRC_M_NOUAF	Do not check authorization file if the process is detached and the image is sys$system:login.exe. However, disabled and expired accounts are still prevented from logging in, and access restrictions are applied. This option is only intended for use with system programs.
PRC_M_PARSE_EXTENDED	Sets the PARSE_STYLE_PERM and the PARSE_STYLE_IMAGE properties for the new process to EXTENDED.
PRC_M_PSWAPM	Inhibit process swapping.
PRC_M_SSFEXCU	Enable system service failure exception mode.
PRC_M_SSRWAIT	Disable resource wait mode.
PRC_M_SUBSYSTEM	Inherit any protected subsystem identifiers.
PRC_M_TCB	Mark the created process as part of the trusted computing base (TCB). The IMPERSONATE privilege is required.
PRC_M_TRUSTED	If not set, captive accounts cannot create a subprocess.
PRC_M_NONRANDOM	The generated name is non-random. See the description of prcnam parameter.
PRC_M_CLONE_SYMBOLS	Clone symbol table from current process.

itmlst
Reserved for future use.

node
Pointer to an SRB that points to the name of the node on which to create the process. If not specified, or the name is null, the process is created on the current node.

home_rad
Pointer to a resource affinity domain (RAD) identifier. If 0, the RAD (if any) is determined by UOS.

Description

This service returns creates a subprocess or detached process on behalf of the calling process. A subprocess can be created only on the current node, but a detached process can be created on the current node or another node. Subprocesses are children of the calling process, while detached processes are independent processes. Subprocesses are automatically terminated when the parent process terminates.

The presence of the uic parameter, node parameter, or the PRC_M_IMPERSONATE flag specifies that the created process is detached.

Condition Values Returned

Values	Meaning
SS_NORMAL	Successful completion.
SS_DUPLNAM	A process already exists with the specified name.
SS_EXPRCLM	The process has used up its quota of detach job creation.
SS_EXQUOTA	The process exceeded one or more quotas.
SS_IVSTSFLG	A reserved flag was specified.
SS_IVQUOTAL	The quota list was invalid.
SS_NOPRIV	The process attempted an operation it didn't have privileges for.
UOSErr_Out_Of_Resources	No process IDs were available.

\

DELLNT

DELLNT

Delete Symbol table

Deletes a symbol table.

Format

SYS$DELLNT(Tabnam, Parent, Default)

Arguments

Tabnam

Address of an SRB defining the name of the table to delete.

Parent

Address of an SRB defining the name of the parent table of the table to delete.

Default

Optional address of an SRB defining the name of the table to become the default table for the parent table. If not specified, and the deleted table was the default table, UOS chooses another table as the default.

Condition Codes

Code	Meaning
SS_NORMAL	Normal completion
SS_BADPARAM	No table or parent name was provided.
SS_NOLOGTAB	A specified table does not exist.
SS_NOPRIV	Calling process doesn't have a required privilege.

DELLNM

DELLNM
Delete Symbol from table

Deletes a symbol or all symbols in a specific table. The symbols deleted are those at the specified access mode, or more outer.

Format

SYS$DELLNM tablename, symbolname, acmode

Arguments

tablename

This is a pointer to a TSRB structure that indicates the table name. This value cannot be null.

symbolname

This is a pointer to a TSRB structure that indicates the symbol name. If the table name is a symbol directory then this value indicates the table name within that directory. In that case, all symbols of the specified access mode, or more outer, in this table are deleted.

acmode

This is an address of an access mode. The symbols deleted are those with this access mode or a more outer one. Note that the access mode is set to the most outer value of either the user's current access mode and the specified access mode. However, if the process has the SYSNAM privilege, the specified access mode is used, regardless of the process' access mode. If the passed acmode is 0, the process' acmode is used.

Description

the DELLNM service deletes all symbols with the specified name at the specified access mode, or more outer access mode. It can also be used to delete all symbols in a table with the specified access mode.

Privileges

SYSNAM is needed to delete a symbol name at an access mode less than the process' access mode. SYSNAM or SYSPRV is needed to delete a symbol from a system table.

Quotas

None.

Condition codes

Code	Meaning
SS_NORMAL	Normal completion.
SS_ACCVIO	Service cannot access memory pointed to by parameters
SS_BADPARAM	Incorrect parameter
SS_IVLOGTAB	Invalid table name
SS_NOLOGTAB	Specified table not found
SS_NOPRIV	Process lacks sufficient privilege for operation

DELPRC

SYS_DELPRC

Allows the process to terminate itself or another process.

Format

SYS$DELPRC {pid} {, name} {, flags}

Arguments

pid

An address pointing to a 64-bit integer containing the process ID to be deleted. Either the PID or the name must be specified if a process other than the current process is to be terminated. Ignored if 0 or if it points to a value of 0.

name

A pointer to an SRB which points to a process name string. If the PID provided is non-zero, it takes precedence over the passed name.

flags

A pointer to a 64-bit integer containing flags that modify the operation of the process termination. The flags are:

Mnuemonic	Meaning
DELPRC_M_EXIT	Exit handlers are called, according to mode. If no exit handler has been specified, the process is immediately terminated.
DELPROC_M_MODE	Indicates the ring in which to call the exit handler.
DELPROC_M_NOEXIT	Disables calling any exit handler.

Description

This service allows a process to terminate itself or another process. If neither PID nor name is specified, the calling process is terminated. Exit handlers are defined via mailboxes. If no exit handler mailbox has been specified, process termination is immediate. It is possible that, due the time necessary to process an exit handler, this system service may return with a successful completion status before the process is actually terminated. See the DCLEXH service for more information on exit handlers.

Any process with the same UIC as the calling process may be terminated with this service. However, GROUP privilege is needed to delete a process with a different UIC but a matching group to the calling process. WORLD privilege is required to delete a process with no matching group to the calling process.

Condition Values Returned

Code	Meaning
SS_NORMAL	Normal completion.
SS_NOPRIV	The calling process does not have the privilege to delete the specified process.
SS_NONEXPR	The specified process does not exist.

DEVICE_SCAN

DEVICE_SCAN

Scan for devices

Returns information about devices match a specified set of filters.

Format

SYS$DEVICE_SCAN returnA, lengthA, devnam, list, context

Argument

returnA

The 64-bit address of a buffer to receive the device name of the next matching device.

lengthA
The 64-bit address of a quadword to receive the length, in bytes, of the device name.

devnam
The 64-bit address of a TSRB structure pointing to the device specification to search for. This can contain wildcards. If 0 is passed, all devices are searched.

list
The 64-bit address of an array of descriptors that define the filter criteria. The last descriptor in the array must consist entirely of 0's. If this is 0, or there are no descriptors before the terminating descriptor, all devices will match.

context
The 64-bit address of a quadword containing the device scan context. If the memory contains a 0, a new device scan context is created and written to the address. In this case, the first matching device will be returned. If the value at the address is a valid context, the next matching device for that context is returned.

System Request Structure

Field	Value
Subsystem	USC
Request ID	UOS_USC_Device_Scan
Length	48
Status	
Flags	Ignored
SRB	16 bytes. Device specification to search for.
Integer1	8 bytes. Address of return buffer.
Integer2	8 bytes. Address of return length.
Integer3	8 bytes. Address of descriptor list.
Integer4	8 bytes. Address of context.

Descriptors
Descriptors are structures with the following layout:

Byte offset	Byte length	Description
0	4	Buffer length.
4	4	Item code
8	8	Buffer address.
16	8	Return length address.

Description
This service returns the names of devices that match the specified filters. Each call returns a device. The context is used to continue multiple calls.

Valid codes:

Code	Description
DVS_DEVCLASS	Device class. One of the following: DC_ANY = Any.all devices DC_DISK = Disk drives DC_TAPE = Tape drives DC_CARD = Card read/writer DC_TERM = Terminal

	DC_LP = Printer DC_REALTIME = Real-time device DC_AUDIO = Audio device DC_VIDEO = Video device DC_MAILBOX = Mailbox DC_REMCSL_STORAGE = Remote storage DC_MISC = Other device DC_SCOM = Synchronous communications device DC_BUS = Peripheral adapter

Required Privileges
None

Affected Quotas
None

Condition Values Returned

SS$_NORMAL	The service completed successfully.
SS$_ACCVIO	The address to receive the time cannot be written to.

DISPLAY

DISPLAY
Get File Attribute Information

This DISPLAY service obtains the attribute information for a file and places the information in FAB, NAML, and XAB control blocks, where present and appropriate.

Format
SYS$DISPLAY fab {err {suc}}

Argument
fab
The 64-bit address of a quadword to receive the current time in 64-bit format.

err
AST error completion routine that is invoked if the operation fails.

suc
AST success routine that is invoked if the operation succeeds.

System Request Structure

Field	Value
Subsystem	FIP
Request ID	UOS_FIP_Display
Length	24
Status	
Flags	Ignored
Integer1	Pointer to FAB control block
Integer2	Pointer to error completion routine
Integer3	Pointer to success completion routine

Description

If the NAML pointer is set in the FAB, it is updated with file information. If any XABs are associated with the FAB, they are filled with the information appropriate for each XAB.

Required Privileges
None

Affected Quotas
None

Condition Values Returned

SS$_NORMAL	The service completed successfully.
SS$_ACCVIO	The address of the FAB cannot be accessed.
RMS$_FAB	The FAB is invalid or missing
RMS$_BLN	The block length for a FAB, NAML, or XAB is invalid
RMS$_XAB	A XAB is invalid

FAO and FAOL

FAO and FAOL

Format ASCII Output

FAO and FAOL format parameters consisting of strings and integer values, according to directives embedded in a control string. The output is the control string with substitutions made depending upon the embedded directives and the parameter values.

FAO can take up to 17 parameters in the function call. FAO is passed a pointer to an array of parameters.

Format

SYS$LIB_FAO control outlen outbuf {p1..p17}
SYS$LIB_FAOL control outlen outbuf parameters

Arguments

control

Pointer to SRB that points to the text to be output, together with one or more FAO directives. Each directive begins with an exclamation point (!). To include a literal exclamation point, the !! directive must be used. There is no limit to the size of the string or how many directives it contains. The valid directives are listed below.

outlen

Defines the address of the maximum output buffer size (an int64) on call. On return, the actual size of the data written to the output buffer is written to the address. Note that the output will never exceed the value at the time the function is called.

outbuf

Defines the address of the output buffer. The converted control string is written here.

p1..p17

Up to 17 64-bit integer values that can represent actual data or pointers to string data. There must be one value for each directive in the string. If the string requires more than are supplied, the missing parameters are assumed to be 0. Not all directives require a parameter, and some constructs may require up to three. Extra parameters are ignored. The parameters are processed sequentially as the control string is processed from left to right. If more than 17 parameters are required, use the LIB_FAOL function instead.

parameters

A pointer to an array of 64-bit integer values that can represent actual data or pointers to string data. There must be one value for each directive in the string. If the string requires more than are supplied, the behavior of the function is undefined, but will probably cause an error. Not all directives require a parameter, and some constructs may require up to three. Extra parameters are ignored. The parameters are processed sequentially as the control string is processed from left to right.

Description
FAO converts integer values into binary, octal, decimal, or hexadecimal values, and can insert strings, and conditionally process directives. See the section below, describing the directives.

FAO Directives
FAO directives can appear anywhere in the control string and have the general form:
!ZZ
where the exclamation point (!) indicates the start of the directive and "ZZ" indicates a 1- or 2-character FAO directive. All alphabetic characters in a directive must be uppercase.

Width
FAO directives optionally can have a width, using this format:
!nZZ
where "n" is the decimal value specifying the width (in characters) for the value substituted for the directive. Example:
!3XB
This would display an integer byte values as hexadecimal (XB) with a width of 3 digits (it is zero-filled on the left).

Repeat
FAO directives optionally can have a repeat count, using this format:
!n(ZZ)
where "n" is the decimal value specifying the number of times that the directive is to be repeated. If the directive requires one or more parameters, sucessive parameters are used for each repetition - the same parameter is not reused for each repetition. Example:
!3(OB)
This would display 3 integer byte values as octal (OB).

Repeat with width
You can specify both a width and a repeat count, using this format:
!n(mZZ)
where "n" is the decimal value specifying the number of times that the directive is to be repeated and "m" is the decimal value specifying the with of the directive output, in characters. Example:
!5(10BB)
This would display five integer byte values as binary (BB), each of which is 10 characters wide.

Variable repeats and widths
You can specify either, or both, a width and a repeat count as variables by using a number sign (#) in place of the decimal value. When such a directive is processed, the next parameter is used in place of the number sign. Example:
!2(#BB)
This would display 2 integer byte values as binary, each of which is a number of characters wide that is defined by the next parameter. Note that even though the directive is repeated, only a single parameter is used for the width - the same width will be used for all iteraions.
!#(OB)
This would display a number of octal values equal to the next parameter.
!#(#OB)
This will read one parameter that will serve as the repeat count, and one more parameter for the width of each octal value output.

Indirect parameters
All string parameters are considered to be addresses of the data. All numeric parameters are assumed to be the actual value. A full 64-bits are required for each parameter value, even if less than 64-bits are required by the directive (the remaining bits are ignored). However, using the indirection symbol (@) in a directive,

FAO can be made to treat a parameter as an address that contains the numeric value. Note that only the required number of bytes are read from that address. Example:
!@UQ
In this case, the next parameter is used as an address to a quadword (64-bit) value.

FAO Directives
String Directives:

Directive	Description
!AB	Inserts a string. The parameter is a pointer to a TSRB structure.
!AC	Inserts a string. The parameter is a pointer to a string whose first byte is the length of the string, followed immediately by that many bytes if text.
!AD	Inserts a string, with periods (.) substituted for all nonprintable ASCII codes. Two parameters are required: the first is the length of the string and the second is the address of the string data.
!AF	Inserts a string. Two parameters are required: the first is the length of the string and the second is the address of the string data.
!AS	Inserts a string. The parameter is the address of a string descriptor for a CLASS_S (static) or CLASS_D (dynamic) string.
!AZ	Inserts a string. The parameter is a pointer to a zero-terminated (ASCIZ) string.

Note: All string lengths indicate number of bytes, not number of characters.

Zero-filled Numeric Directives:

Directive	Description
!BB	Convert a byte value to the ASCII representation of that value in base 2. Only the low byte of the parameter is used.
!BW	Convert a word value to the ASCII representation of that value in base 2. Only the lower two bytes of the parameter are used.
!BL	Convert a longword value to the ASCII representation of that value in base 2. Only the lower four bytes of the parameter are used.
!BQ	Convert a quadword value to the ASCII representation of that value in base 2.
!OB	Convert a byte value to the ASCII representation of that value in base 8. Only the low byte of the parameter is used.
!OW	Convert a word value to the ASCII representation of that value in base 8. Only the lower two bytes of the parameter are used.
!OL	Convert a longword value to the ASCII representation of that value in base 8. Only the lower four bytes of the parameter are used.
!OQ	Convert a quadword value to the ASCII representation of that value in base 8.
!OA	Same as !OQ.
!OI	Same as !OL.
!OH	Same as !OQ.
!OJ	Same as !OQ.
!XB	Convert a byte value to the ASCII representation of that value in base 16. Only the low byte of the parameter is used.
!XW	Convert a word value to the ASCII representation of that value in base 16. Only the lower two bytes of the parameter are used.
!XL	Convert a longword value to the ASCII representation of that value in base 16. Only the lower four bytes of the parameter are used.
!XQ	Convert a quadword value to the ASCII representation of that value in base 16.
!XA	Same as !XQ.
!XI	Same as !XL.
!XH	Same as !XQ.
!XJ	Same as !XQ.
!ZB	Convert a byte value to the ASCII representation of that value in base 10. Only the low byte of the parameter is used.

!ZW	Convert a word value to the ASCII representation of that value in base 10. Only the lower two bytes of the parameter is used.
!ZL	Convert a longword value to the ASCII representation of that value in base 10. Only the lower four bytes of the parameter is used.
!ZQ	Convert a quadword value to the ASCII representation of that value in base 10.
!ZA	Same as !ZQ.
!ZI	Same as !ZL.
!ZH	Same as !ZQ.
!ZJ	Same as !ZQ.

Blank-filled Numeric Directives:

Directive	Description
!UB	Convert an unsigned byte value to the ASCII representation of that value in base 10. Only the low byte of the parameter is used.
!UW	Convert an unsigned word value to the ASCII representation of that value in base 10. Only the lower two bytes of the parameter are used.
!UL	Convert an unsigned longword value to the ASCII representation of that value in base 10. Only the lower four bytes of the parameter are used.
!UQ	Convert an unsigned quadword value to the ASCII representation of that value in base 10.
!UA	Same as !UQ.
!UI	Same as !UL.
!UH	Same as !UQ.
!UJ	Same as !UQ.
!SB	Convert a signed byte value to the ASCII representation of that value in base 10. Only the low byte of the parameter is used.
!SW	Convert a signed word value to the ASCII representation of that value in base 10. Only the lower two bytes of the parameter are used.
!SL	Convert a signed longword value to the ASCII representation of that value in base 10. Only the lower four bytes of the parameter are used.
!SQ	Convert a signed quadword value to the ASCII representation of that value in base 10.
!SH	Same as !SL.
!SJ	Same as !SL.

Other Directives:

Directive	Description
!/	Inserts a new line (carriage return and linefeed). It takes no parameters.
!_	Inserts a horizontal tab (ASCII 9). It takes no parameters.
!^	Inserts a form feed. It takes no parameters.
!!	Inserts an exclamation point. It takes no parameters
!%S	Inserts the letter S if the most recently converted numeric value is not 1. If the character before the directive is upper case, an upper case S is inserted, otherwise a lowercase s is inserted.
!%T	Inserts the system time. The parameter is the datetime stamp. If the parameter is 0, the current time is inserted.
!%U	Same as !UQ.
!%I	Converts a UIC to the account name. If an invalid UIC is specified, the directive is treated as !UQ.
!%D	Inserts the system date and time. The parameter is the timestamp. If the parameter is 0, the current date/time is inserted.
!n%C	Conditional. See discussion of conditionals below.
!%E	Else portion of conditional. See discussion of conditionals below.
!%F	End of conditional. See discussion of conditionals below.
!n<	See next directive.

!>	The preceeding directive and this one are used together to define an output field that has a width of n. Within this field are displayed all directives between the !n< and !> directives. The field is blank-filled on the right to make it n characters wide if necessary. All directives within this field are left-justified and blank-filled. Note that these can be nested.
!n*c	Repeats the character c in the output n times.
!-	Reuse the most recently used parameter value.
!+	Skip the next parameter value.

Conditionals
!%nC, !%E, and !%F are used together to insert values depending upon parameter values. This is primarily for use with plurals. The general format is:
!%nCa!%Eb!%F
If n matches the last parameter value, then a is inserted, otherwise b is inserted. Example:
!ZB !%1Cchild!%Echildren!%F
In this example, if the first parameter is 1, the output would be:
1 child
But if the first parameter is not 1, the output would be:
n children
where "n" is the value of the first parameter.

The following table illustrates how the directives interact with width and filling.

Directive Type	Default output width	When explicit width is greater than default	When explicit width is less than default
!BB	8	Right justify and blank fill	Result truncated on left
!BW	16	Right justify and blank fill	Result truncated on left
!BL	32	Right justify and blank fill	Result truncated on left
!BQ	64	Right justify and blank fill	Result truncated on left
!OB	3	Right justify and blank fill	Result truncated on left
!OW	6	Right justify and blank fill	Result truncated on left
!OL	11	Right justify and blank fill	Result truncated on left
!OQ	22	Right justify and blank fill	Result truncated on left
!HB	2	Right justify and blank fill	Result truncated on left
!HW	4	Right justify and blank fill	Result truncated on left
!HL	8	Right justify and blank fill	Result truncated on left
!HQ	16	Right justify and blank fill	Result truncated on left
Unsigned zero-filled decimal	As many characters as are necessary	Right justify and blank fill	Field completely filled with asterisks (*)
Signed or	As	Right justify and	

unsigned decimal	many characters as are necessary	zero-filled	
Strings	As many characters as in the string	Left justify and blank fill to specified length	Truncate on right

FILESCAN

FILESCAN
Scan File Specification

Format

SYS$FILESCAN filespec, itemlist, fldflags, auxout, retlen

Arguments

filespec

The 64-bit address of a SRB structure pointing to the filename to scan.

itemlist

Item list specifying which components of the file specification are to be returned. This argument is the address of the first descriptor in the list. Each descriptor has the following layout:

Bytes	Description
0-3	Item Code. Indicates the file specification field to return. See table below for valid item codes.
4-7	Length. This is where the length of the field is written. If the corresponding field is missing, 0 is written here.
8-15	Address. The address of the start of the field is written here. If the corresponding field is missing, 0 is written here.

fldflgs

The address of where a bitmask is written that indicates which fields of the file specification were specified. If this value is 0, this is ignored. The fields are indicated by the following flag values:

Symbol name	Description
FSCN_V_DEVICE	Device name
FSCN_V_DIRECTORY	Directory name
FSCN_V_NAME	File name
FSCN_V_NODE	Node name
FSCN_V_NODE_ACS	Access control string of primary node
FSCN_V_NODE_PRIMARY	Primary (first) node name
FSCN_V_NODE_SECONDARY	Secondary (additional) node information
FSCN_V_ROOT	Root directory name string

FSCN_V_TYPE	File type
FSCN_V_VERSION	Version number

auxout

Auxillary output buffer. This argument is the address of an SRB structure which indicates where the complete file specification (as provided) is written. Any secondary node information is stripped from the output and quotations are reduced and simplified. If this value is 0, it is ignored. If provided, the values written to the item list are addresses within this auxillary buffer.

retlen

Auxillary output buffer. This argument is the address of an SRB structure which indicates where the complete file specification (as provided) is written. Any secondary node information is stripped from the output and quotations are reduced and simplified. If this value is 0, it is ignored. If provided, the values written to the item list are addresses within this auxillary buffer.

Description

The FILESCAN service searches a string for a file specification and parses the fields of that specification. The length and starting addresses of the fields requested are returned. If a field was requested in the item list but not found in the file specification, a length and address of 0 are written to the descriptor. The descriptor list is terminated with a descriptor that has an item code of 0.

The information returned describes the entire contiguous file specification. For example, to extract only the file name and type from the full string, you can use the address of the file name, for the length of the sum of the name and type to obtain the full file name. However, FSCN_NODE_PRIMARY and FSCN_NODE_ACS items contain no double colon (::), so you would have to add 2 to the sum of the lengths of those two fields to obtain the entire node specification.

FILESCAN does not check all aspects of validity in the specification. For instance, it does not verify that the node name specified corresponds to a valid node. Nor does it validate the access control string contents. Nor does it verify the existence of the path or specified file. It treats wildcard characters as any other valid character. It doesn't validate lengths either. Finally, multiple whitespace characters are not collapsed to a single space, nor trimmed from the beginning or end of the string. However, spaces, tabs, and delimiting characters must be enclosed in quotes if they are part of the file name or type, otherwise the character is treated as a terminator for the specification. Quotes used to indicate a node access control string require that the node name be enclosed in quotes and that the quotes delimiting the access control string must be doubled (""). For example, the node specification:

abcd"efg"

would need to be specified as:

"abcd""efg"""

FILESCAN does not assume default values for missing fields or perform logical name translations.

Here are the item codes that can be used in the passed descriptors:

Symbol name	Description
FSCN_DEVICE	Returns length and starting address of the device name, including the colon (:).
FSCN_DIRECTORY	Returns the length and starting address of the path, including all backslashes (\).
FSCN_FILESPEC	Returns the length and starting address of the full file specification.
FSCN_NAME	Returns the length and starting address of the file name, including no syntatical elements.
FSCN_NODE	Returns the length and starting address of the node, access control string, and double colon (::).
FSCN_NODE_ACS	Returns the length and starting address of the node access control string.
FSCN_NODE_PRIMARY	Returns the length and starting address of the primary node name. It doesn't include the double colon (::) or access control string.

FSCN_NODE_SECONDARY	Returns the length and starting address of the secondary node string.
FSCN_ROOT	Returns the length and starting address of the root diretory of the path, including backslashes (\).
FSCN_TYPE	Returns the length and starting address of the file type, including the leading dot (.).
FSCN_VERSION	Returns the length and starting address of the version, including the leading semicolon (;).

GETDVI

GETDVI

Get Device Information

Format

SYS$GETDVI efn handle device list iosb astadr astprm pathname size

Argument

efn

The 64-bit address of a quadword to receive the current time in 64-bit format.

handle

File handle associated with device. If 0, the device name is used to determine the device.

device

64-bit pointer to SRB pointing to the device name. Ignored if the handle is provided.

list

64-bit pointer to an array of descriptors indicating what data to return. The array must end with a descriptor consisting entirely of 0's.

iosb

64-bit pointer to IO Status Block to receive status.

astadr

Address of AST routine to be called when the operation completes.

astprm

64-bit parameter to pass to AST routine.

pathname

64-bit pointer to SRB that points to the device path name to use for mutlipath devices.

size

Number of descriptors in the array.

System Request Structure

Field	Value
Subsystem	FIP
Request ID	UOS_FIP_GetDVI
Length	68
Status	
Flags	Ignored

File handle	8 bytes. 64-bit address to receive time.
File stream	4 bytes. Ignored.
Buffer	8 bytes. 64-bit address of item list.
Buffer length	8 bytes. Number of descriptors in list.
Buffer1	8 bytes. Ignored.
Buffer1 length	8 bytes. Ignored.
File flags	4 bytes. Ignored.
Device buffer	8 bytes. 64-bit address of buffer containing device name.
Device length	8 bytes. Length of device name, in bytes.
Device flags	4 bytes. Ignored.

Descriptors

Each descriptor has the following layout:

Byte offset	Byte length	Description
0	2	Must be -1
2	2	Item code
4	4	Must be -1
8	8	Address of result buffer length.
16	8	Address of result buffer.
24	8	Address of result length.

Description

This service returns authorization information about a device. The valid descriptor codes are as follows:

Code	Result Type	Description
DVI_ACCESSTIME_RECORDED	Integer	Returns 1 if the volume supports recording access times, 0 otherwise.
DVI_ALL	Integer	Returns 1 if the device is allocated, 0 otherwise.
DVI_ALLDEVNAM	String	Returns the allocatable device name.
DVI_ALLOCLASS	Integer	Returns the allocation class of the host as an integer. An allocation class is a unique number that the system administrator assigns.
DVI_AVL	Integer	Returns 1 if the device is available (mounted, online, ready, etc.), 0 otherwise.
DVI_CLUSTER	Integer	Returns volume clustersize, in bytes.
DVI_DEVBUFSIZ	Integer	Returns the device buffer size, in bytes. This defaults to the terminal width, tape block size, store cluster size, etc.
DVI_DEVCHAR	Integer	Returns device characteristics.

Flag	Description
DEV_V_ALL	Device is allocated.
DEV_V_AVL	Device is available for use (mounted, online, ready, etc.).
DEV_V_CCL	This is a carriage-control Device.
DEV_V_DIR	Device is directory structured.
DEV_V_DMT	Device is marked for dismount.

<table>
<tr><td></td><td></td><td>
<table>
<tr><td>DEV_V_DUA</td><td>Device is dual-ported.</td></tr>
<tr><td>DEV_V_FOD</td><td>File structured device.</td></tr>
<tr><td>DEV_V_GEN</td><td>This is a generic device.</td></tr>
<tr><td>DEV_V_IDV</td><td>Device can provide input.</td></tr>
<tr><td>DEV_V_MNT</td><td>Device is mounted.</td></tr>
<tr><td>DEV_V_ODV</td><td>Device can provide output.</td></tr>
<tr><td>DEV_V_REC</td><td>Device is record oriented.</td></tr>
<tr><td>DEV_V_SQD</td><td>Device is sequential and block oriented.</td></tr>
<tr><td>DEV_V_RND</td><td>Device allows random access.</td></tr>
<tr><td>DEV_V_SWL</td><td>Device is software write locked.</td></tr>
<tr><td>DEV_V_TRM</td><td>Device is a terminal.</td></tr>
</table>
</td></tr>
<tr><td>DVI_DEVCLASS</td><td>Integer</td><td>Returns the device class as an integer. The following are the return values.
<table>
<tr><th>Mnuemonic</th><th>Meaning</th></tr>
<tr><td>DC_DISK</td><td>Disk</td></tr>
<tr><td>DC_TAPE</td><td>Tape</td></tr>
<tr><td>DC_CARD</td><td>Card reader/writer.</td></tr>
<tr><td>DC_TERM</td><td>Terminal</td></tr>
<tr><td>DC_LP</td><td>Printer</td></tr>
<tr><td>DC_REALTIME</td><td>Real-time device.</td></tr>
<tr><td>DC_AUDIO</td><td>Audio device</td></tr>
<tr><td>DC_VIDEO</td><td>Video device</td></tr>
<tr><td>DC_MAILBOX</td><td>Mailbox</td></tr>
<tr><td>DC_REMCSL_STORAGE</td><td>Remote storage.</td></tr>
<tr><td>DC_MISC</td><td>Other device not covered by other values.</td></tr>
<tr><td>DC_SCOM</td><td>Synchronous communications device.</td></tr>
<tr><td>DC_BUS</td><td>Peripheral adapter.</td></tr>
</table>
</td></tr>
<tr><td>DVI_DEVDEPEND</td><td>Integer</td><td>Device-dependent flags.</td></tr>
<tr><td>DVI_DEVDEPEND2</td><td>Integer</td><td>Additional device-dependent flags.</td></tr>
<tr><td>DVI_DEVLOCKNAM</td><td>String</td><td>Returns the device lock name which uniquely identifies each volume in a UOS cluster or single-node system.</td></tr>
<tr><td>DVI_DEVTYPE</td><td>Integer</td><td>Returns the device type.</td></tr>
<tr><td>DVI_ERASE_ON_DELETE</td><td>Integer</td><td>Returns 1 if the store erases files on delete, 0 otherwise.</td></tr>
<tr><td>DVI_FC_HBA_FIRMWARE_REV</td><td>Integer</td><td>Same as DVI_FIRMWARE_REV.</td></tr>
<tr><td>DVI_FIRMWARE_REV</td><td>Integer</td><td>Returns the device firmware revision.</td></tr>
<tr><td>DVI_FREEBLOCKS</td><td>Integer</td><td>Returns the number of free clusters on the store.</td></tr>
<tr><td>DVI_FULLDEVNAM</td><td>String</td><td>Returns a displayable device name.</td></tr>
<tr><td>DVI_HARDLINKS_SUPPORTED</td><td>Integer</td><td>Returns 1 if the file system on the device supports hardlinks.</td></tr>
</table>

<table>
<tr><td>DVI_MAXBLOCK</td><td>Integer</td><td>Returns the maximum cluster number on the store.</td></tr>
<tr><td>DVI_MAXFILES</td><td>Integer</td><td>Returns the maximum number of files supported on the store. 0 is returned if the device has no limit or files are not supported.</td></tr>
<tr><td>DVI_MOUNT_TIME</td><td>Integer</td><td>Returns the internal UOS time when the device was last mounted.</td></tr>
<tr><td>DVI_MOUNTCNT</td><td>Integer</td><td>Returns the number of times the device was mounted on this system.</td></tr>
<tr><td>DVI_OPCNT</td><td>Integer</td><td>Returns the number of operations on this device since the system was booted or the operation count was reset.</td></tr>
<tr><td>DVI_OWNUIC</td><td>Integer</td><td>Returns the UIC of the user who owns the device.</td></tr>
<tr><td>DVI_PID</td><td>Integer</td><td>Returns the PID of the owner of the device.</td></tr>
<tr><td>DVI_QLEN</td><td>Integer</td><td>Returns the length of the I/O queue on the device. This is specifically the queue length of the driver and/or hardware - not of the File Processor's I/O queue.</td></tr>
<tr><td>DVI_RECSIZ</td><td>Integer</td><td>Returns the record size of the device.</td></tr>
<tr><td>DVI_REMOTE_DEVICE</td><td>Integer</td><td>Returns 1 if the device is remote, or 0 otherwise.</td></tr>
<tr><td>DVI_SCSI_DEVICE_FIRMWARE_REV</td><td>String</td><td>Same as DVI_FIRMWARE_REV.</td></tr>
<tr><td>DVI_SERIALNUM</td><td>String</td><td>Returns the serial number of the volume.</td></tr>
<tr><td>DVI_STS</td><td>Integer</td><td>Returns the device status consisting of the following flags:
<table>
<tr><th>Flag</th><th>Description</th></tr>
<tr><td>UCB_V_ALTBSY</td><td>Unit is busy via alternate startio path.</td></tr>
<tr><td>UCB_V_BSY</td><td>Unit is busy.</td></tr>
<tr><td>UCB_V_CANCEL</td><td>I/O on unit is canceled.</td></tr>
<tr><td>UCB_V_CLUTRAN</td><td>OpenVMS Cluster state transition in progress.</td></tr>
<tr><td>UCB_V_DEADMO</td><td>Deallocate at dismount.</td></tr>
<tr><td>UCB_V_DELETEUCB</td><td>Delete this UCB when reference count equals 0.</td></tr>
<tr><td>UCB_V_DISMOUNT</td><td>Dismount in progress.</td></tr>
<tr><td>UCB_V_ERLOGIP</td><td>Error log is in progress on unit.</td></tr>
<tr><td>UCB_V_EXFUNC_SUPP</td><td>Unit supports the EXFUNC bit.</td></tr>
<tr><td>UCB_V_FAST_PATH</td><td>Unit supports FAST PATH Affinity.</td></tr>
<tr><td>UCB_V_FP_HWINT</td><td>Unit supports FAST PATH hardware interrupt CPU Affinity.</td></tr>
<tr><td>UCB_V_INT</td><td>Interrupt is expected.</td></tr>
<tr><td>UCB_V_INTTYPE</td><td>Receiver interrupt.</td></tr>
<tr><td>UCB_V_IOPOST_LOCAL</td><td>Unit supports I/O post processing on the current CPU.</td></tr>
</table></td></tr>
</table>

		UCB_V_LCL_VALID: Volume is valid on the local node. UCB_V_MNTVERIP: Mount verification is in progress. UCB_V_MOUNTING: Device is being mounted. UCB_V_MNTVERPND: Mount verification is pending on busy device. UCB_V_NO_ASSIGN: Unit cannot have channels assigned to it. UCB_V_ONLINE: Unit is on line. UCB_V_PATHVERIP: Path verification is in progress for this device. UCB_V_POWER: Power failed while unit busy. UCB_V_SNAPSHOT: Restart validation is in progress. UCB_V_SUPMVMSG: If set, suppress success type mount version messages. UCB_V_SVPN_END: Last byte used from page mapped by system virtual page number (SVPN). UCB_V_TEMPLATE: Template UCB from which other UCBs for this device type are made. UCB_V_TIM: Timeout is enabled. UCB_V_TIMOUT: Unit timed out. UCB_V_UNLOAD: Unload volume at dismount. UCB_V_VALID: Volume is software valid. UCB_V_WRONGVOL: Wrong volume detected during mount verification. UCB_V_WRTLOCKMV: Write-locked mount verification in progress.
DVI_TRANSCNT	Integer	Returns the transaction count for the volume.
DVI_TT_ALTYPEAHD	Integer	Returns 1 if the device is a terminal and allows typeahead, 0 otherwise.
DVI_TT_ANSICRT	Integer	Returns 1 if the device is a terminal and is an ANSI CRT, 0 otherwise.
DVI_TT_APP_KEYPAD	Integer	Returns 1 if the device is a terminal and the application keypad is set, 0 otherwise.
DVI_TT_AUTOBAUD	Integer	Returns 1 if the device is a terminal and set to auto baud detection, 0 otherwise.
DVI_TT_AVO	Integer	Returns 1 if the device is a terminal with advanced video option, 0 otherwise.
DVI_TT_BLOCK	Integer	Returns 1 if the device is a terminal in block mode, 0

		otherwise.
DVI_TT_BRDCSTMBX	Integer	Returns 1 if the device is a terminal and broadcast messages are sent to a mailbox.
DVI_TT_CRFILL	Integer	Returns 1 if the device is a terminal and sends fill characters after CR.
DVI_TT_DECCRT	Integer	Returns 1 if the device is a DEC CRT terminal.
DVI_TT_DECCRT2	Integer	Returns 1 if the device is a DEC CRT terminal.
DVI_TT_DECCRT3	Integer	Returns 1 if the device is a DEC CRT terminal.
DVI_TT_DECCRT4	Integer	Returns 1 if the device is a DEC CRT terminal.
DVI_TT_DIALUP	Integer	Returns 1 if the device is a dialup terminal, 0 otherwise.
DVI_TT_DISCONNECT	Integer	Returns 1 if the device is a terminal and is diconnected when a hangup or disconnect occurs.
DVI_TT_DMA	Integer	Returns 1 if the device is a terminal with DMA, 0 otherwise.
DVI_TT_DRCS	Integer	Returns 1 if the device is a terminal that supports loadable characters, 0 otherwise.
DVI_TT_EDIT	Integer	Returns 1 if the device is a terminal and supports edit mode.
DVI_TT_EDITING	Integer	Returns 1 if the device is a terminal in edit mode, 0 otherwise.
DVI_TT_EIGHTBIT	Integer	Returns 1 if the terminal uses the 8-bit ASCII character set otherwise use 7-bit ASCII code.
DVI_TT_ESCAPE	Integer	Returns 1 if the device is a terminal that supports escape sequences, 0 otherwise.
DVI_TT_FALLBACK	Integer	Returns 1 if the terminal output is transformed from the 8-bit multinational character set to a 7-bit ASCII character set, 0 otherwise.
DVI_TT_HALFDUP	Integer	Returns 1 if the device is a terminal in half-duplex mode, 0 otherwise.
DVI_TT_HANGUP	Integer	Returns 1 if the modem is diconnected when the user logs off.
DVI_TT_HOSTSYNC	Integer	Returns 1 if Ctrl/Q and Ctrl/S are used to control data flow.
DVI_TT_INSERT	Integer	Returns 1 if the device is a terminal and the default mode for insert or overstrike is set at the beginning of each read operation.
DVI_TT_LFFILL	Integer	Returns 1 if the device is a terminal and fill characters are send after LF, 0 otherwise.
DVI_TT_LOCALECHO	Integer	Returns 1 if the device is a terminal with local echo set, 0 otherwise.
DVI_TT_LOWER	Integer	Returns 1 if the device is a terminal that supports lowercase characters, 0 otherwise.
DVI_TT_MBXDSABL	Integer	Returns 1 if mailboxes associated with the terminal do not receive notification of unsolicited input or hangup, 0 otherwise.
DVI_TT_MECHFORM	Integer	Returns 1 if the device is a terminal with mechanical formfeed, 0 otherwise.
DVI_TT_MECHTAB	Integer	Returns 1 if the device is a terminal with mechanical tab, 0 otherwise.
DVI_TT_MODEM	Integer	Returns 1 if the device is a terminal connected via a

		modem, 0 otherwise.
DVI_TT_MODHANGUP	Integer	Returns 1 if the device is a terminal with modified hangup, 0 otherwise.
DVI_TT_NOBRDCST	Integer	Returns 1 if the device is a terminal and broadcast messages are not sent to it, 0 otherwise.
DVI_TT_NOECHO	Integer	Returns1 if the device is a terminal and echo is disabled, 0 otherwise.
DVI_TT_NOTYPEAHD	Integer	Returns 1 if the device is a terminal that doesn't accept typeahead, 0 otherwise.
DVI_TT_OPER	Integer	Returns 1 if the device is a terminal marked as an operator, 0 otherwise.
DVI_TT_PAGE	Integer	Returns the terminal page size. For character terminals, this is the number of lines. For graphical terminals, this is the number of pixels in the Y dimension.
DVI_TT_PASTHRU	Integer	Returns 1 if the terminal is in binary I/O mode, 0 otherwise.
DVI_TT_PHYDEVNAM	String	Returns the physical device name of the terminal.
DVI_TT_PRINTER	Integer	Returns 1 if the terminal has an attached printer, 0 otherwise.
DVI_TT_READSYNC	Integer	Returns 1 if UOS explicitly solicits all read operations by Ctrl/Q and terminates the operation by Ctrl/S.
DVI_TT_REGIS	Integer	Returns 1 if the device is a terminal with Regis graphics support, 0 otherwise.
DVI_TT_REMOTE	Integer	Returns 1 if this is a remote device, 0 otherwise.
DVI_TT_SCOPE	Integer	Returns 1 if the device is a video terminal, 0 otherwise.
DVI_TT_SECURE	Integer	Returns 1 if no process is connected to the terminal after the BREAK key is pressed, 0 if BREAK is a null key.
DVI_TT_SETSPEED	Integer	Returns 1 if either LOG_IO or PHY_IO privilege is required to change terminal speed, 0 otherwise.
DVI_TT_SIXEL	Integer	Returns 1 if the device is a terminal with Sixel support, 0 otherwise.
DVI_TT_SYSPWD	Integer	Returns 1 if the device is a terminal and the login procedure should require the system password before the user name prompt is displayed, 0 otherwise.
DVI_TT_TTSYNC	Integer	Returns 1 if the device is a terminal and output is controlled by terminal-generated Ctrl/Q or Ctrl/S, otherwise 0.
DVI_TT_WRAP	Integer	Returns 1 if the device is a terminal and UOS sends a CRLF when the right margin is exceeded.
DVI_UNIT	Integer	Returns unit number of device.

Affected Quotas
None

Condition Values Returned

SS$_NORMAL	The service completed successfully.
SS$_ACCVIO	The address to receive the time cannot be written to.

GETJPI

GETJPI
Get Job/Process Information

GETJPI returns information about one or more processes on the system or cluster.

Format

SYS$GETJPI efn pidadr prcnam list iosb astadr astprm

Return Value

Either an integer or a string, depending upon the item requested. The table below indicates the valid items and the corresponding return values.

Arguments

efn

Event flag to set upon completion.

pidadr

Address of PID of the process to return information about. If pidadr is -1, GETJPI assumes a wildcard operation and returns the requested information for each process on the system that the calling process has privilege to access. Each call will return information about the next process.
If used with PROCESS_SCAN, you can search for specific processes by passing the process scan context instead of the address of a process ID.
If pidadr is 0 and prcnam is not null, prcnam is used to identify the target process. If pidadr is 0 and prcnam is null, the current process is used as the target process.
If pidadr is a valid address, and the value in the memory pointed to by pidadr is 0, the current process is used as the target process. In this case, the current process' ID is written back to that memory location.

prcnam

An SRB pointing to a string containing the process name. If pidadr is non-zero, this parameter is ignored.

list

Pointer to a list of descriptors which indicate what information to return. The valid item codes are listed below.

iosb

I/O status block to receive the final completion status.

astadr

Address of AST routine to execute when GETJPI completed. If this is 0, no AST routine is called.

astprm

AST parameter to be passed to the AST service routine specified by the astadr parameter.

System Request Structure

Field	Value
Subsystem	USC
Request ID	UOS_USC_GetJPI
Length	56
Status	
Flags	Ignored

PID	8 bytes. Process ID.
List	8 bytes. Address of item list.
Length	8 bytes. Ignored
Flags2	4 bytes. Ignored
Buffer	8 bytes. Address of buffer containing the process name.
Length	8 bytes. Length of process name.
Flags	4 bytes. Ignored.
Inte	8 bytes. Pointer to IOSB.

Descriptors

Offset byte	Size in bytes	Description
0	2	Must be -1
2	2	Item code
4	4	Must be -1
8	8	Result buffer address
16	8	Result buffer size
24	8	Result length

Description
The Get Job/Process Information service returns information about one or more processes on the system or cluster. Using this service along with PROCESS_SCAN allows you to perform selective process searches.

The calling process must have GROUP privilege to obtain information about processes belonging to one of the groups to which the caller belongs. The calling process must have WORLD privilege to obtain information about other processes on the system that do not belong to one of the groups to which the caller belongs.

If both pidadr and prcnam are specified, pidadr is used and prcnam is ignored. If neither is provided, the current process is the target process. Generally it is preferable to use pidadr instead of prcnam as there is less overhead in passing parameters as well as avoiding the overhead in searching the process tables for the specified name.

The following item codes are valid for the passed descriptor list:

Item	Return type	Information returned
JPI_ACCOUNT	Integer	The account name.
JPI_APTCNT	Integer	Active page table count.
JPI_ASTACT	Integer	Access modes with active ASTs. Bitmask indicating ring of ASTs: bit 0 = ring 0, bit 1 = ring 1, etc.
JPI_ASTCNT	Integer	Remaining AST quota.
JPI_ASTEN	Integer	Access modes with ASTs enabled. Bitmask indicating ring of ASTs: bit 0 = ring 0, bit 1 = ring 1, etc.
JPI_ASTLM	Integer	AST limit quota.
JPI_AUTHPRI	Integer	Maximum authorized priority.
JPI_AUTHPRIV	Integer	Authorized privileges. 64-bit privilege mask.
JPI_BIOCNT	Integer	Remaining buffered I/O quota.
JPI_BIOLM	Integer	Buffered I/O limit quota.
JPI_BUFIO	Integer	Count of buffered I/O operations.
JPI_BYTCNT	Integer	Remaining buffered I/O byte count quota.
JPI_BYTLM	Integer	Buffered I/O byte count quota.
JPI_CASE_LOOKUP_IMAGE	Integer	Information about the file name lookup case sensitivity for

		the life of the currently running image: BLIND or SENSITIVE.
JPI_CASE_LOOKUP_PERM	Integer	Information about the file name lookup case sensitivity for the life of the process: BLIND or SENSITIVE.
JPI_CLASSIFICATION	Integer	Current MAC classification
JPI_CHAIN	Integer	UOS ignores this item and moves to the next one.
JPI_CLINAME	String	Current command language interpreter (shell).
JPI_CPULIM	Integer	CPU time limit.
JPI_CPUTIM	Integer	CPU time used, in nanoseconds.
JPI_CREPRC_FLAGS	Integer	Flags specified by the stsflg argument in the CREPRC system call that created the process.
JPI_CURPRIV	Integer	Current process privileges.
JPI_CURRENT_CAP_MASK	Integer	Current capabilities mask. 64-bit privilege mask.
JPI_DFPFC	Integer	Default page fault cluster size.
JPI_DFWSCNT	Integer	Default working set size.
JPI_DIOCNT	Integer	Remaining direct I/O count.
JPI_DIOLM	Integer	Direct I/O limit.
JPI_DIRIO	Integer	Count of direct I/O operations.
JPI_EFCS	Integer	Local event flags 0-31.
JPI_EFCU	Integer	Local event flags 32-63.
JPI_EFWM	Integer	Event flag wait mask.
JPI_ENQCNT	Integer	Lock request quota remaining.
JPI_ENQLM	Integer	Lock request quota limit.
JPI_EXCVEC	Integer	Address of a list of exception vectors.
JPI_FAST_VP_SWITCH	Integer	Number of times process has issued vector processor that enabled an inactive vector processor without the expense of a vector context switch.
JPI_FILCNT	Integer	Remaining open file quota.
JPI_FILLM	Integer	Open file quota.
JPI_FINALEXC	Integer	Address of a list of final exception vectors. There are four vectors in the list: one for each ring, from 0 to 3. Each vector is 64-bits.
JPI_FREP0VA	Integer	First free page at end of executable/data address space.
JPI_FREP1VA	Integer	Last free page before start of stack space.
JPI_FREPTECNT	Integer	Number of pages available for virtual memory expansion.
JPI_GPGCNT	Integer	Global page count in working set.
JPI_GRP	Integer	Group(s) the process user belongs to. An array of group IDs.
JPI_HOME_RAD	Integer	Home resource affinity domain (RAD).
JPI_IMAGECOUNT	Integer	Number of image rundowns.
JPI_IMAGE_AUTHPRIV	Integer	Authorized privilege for running image. A 64-bit privilege mask.
JPI_IMAGE_PERMPRIV	Integer	Default privilege for running image. A 64-bit privilege mask.
JPI_IMAGE_WORKPRIV	Integer	Current (working) privilege for running image. A 64-bit privilege mask.
JPI_IMAGNAME	String	File name and path of current image.
JPI_IMAGPRIV	Integer	Privileges current image was installed with. A 64-bit privilege mask.
JPI_JOBPRCCNT	Integer	Number of subprocesses owned by job.
JPI_JOBTYPE	Integer	Execution mode of the process at the root of the job tree: JPI_K_DETACHED JPI_K_NETWORK JPI_K_BATCH JPI_K_LOCAL

		JPI_K_DIALUP JPI_K_REMOTE
JPI_LAST_LOGIN_I	Integer	Date/Time of last interactive login.
JPI_LAST_LOGIN_N	Integer	Date/Time of last noninteractive login.
JPI_LOGIN_FAILURES	Integer	Number of login failures prior to the start of the current session.
JPI_LOGIN_FLAGS	Integer	Bitmask containing additional information relating to the login sequence.
JPI_LOGINTIM	Integer	Process creation time.
JPI_MASTER_PID	Integer	PID of the top of the job's process tree.
JPI_MAXDETACH	Integer	Maximum number of detached processes allowed for the user who owns the process.
JPI_MAXJOBS	Integer	Maximum number of active processes allowed for user who owns the process.
JPI_MODE	Integer	Current process mode: JPI_K_OTHER JPI_K_NETWORK JPI_K_BATCH JPI_K_INTERACTIVE
JPI_MSGMASK	Integer	Current message mask.
JPI_MULTITHREAD	Integer	Current multuthread limit.
JPI_NODENAME	String	Name of the cluster node on which the process is running.
JPI_NODE_CSID	Integer	Cluster ID of the cluster node on which the process is running.
JPI_NODE_VERSION	Integer	UOS version number of the cluster node on which the process is running.
JPI_OWNER	Integer	PID of process' owner.
JPI_PAGEFLTS	Integer	Page fault count.
JPI_PAGFILCNT	Integer	Remaining page file quota.
JPI_PARSE_STYLE_PERM	Integer	Values set by the SET_PROCESS_PROPERTIESW.
JPI_PARSE_STYLE_IMAGE	Integer	Values set by the SET_PROCESS_PROPERTIESW.
JPI_PERMANENT_CAP_MASK	Integer	Permanent capabilities mask.
JPI_PERSONA_AUTHPRIV	Integer	Authorized privilege mask of the persona.
JPI_PERSONA_ID	Integer	The ID of the persona.
JPI_PERSONA_PERMPRIV	Integer	Default privilege mask of the persona.
JPI_PERSONA_WORKPRIV	Integer	Current privilege mask of the active persona.
JPI_PGFLQUOTA	Integer	Page file quota.
JPI_PHDFLAGS	Integer	Flags word.
JPI_PID	Integer	Process ID.
JPI_PPGCNT	Integer	Process page count.
JPI_PRCCNT	Integer	Number of subprocesses owned by process. This does not include subprocesses owned by any subprocesses of the process.
JPI_PRCLM	Integer	Subprocess quota.
JPI_PRCNAM	String	Process name. Because this can be up to 16 characters long, the return buffer should be large enough to hold 16 bytes.
JPI_PRI	Integer	Current priority.
JPI_PRIB	Integer	Base priority.
JPI_PROC_INDEX	Integer	Same as JPI_PID.
JPI_PROCPRIV	Integer	Default privileges.
JPI_SCHED_CLASS_NAME	String	Name of the scheduling class if process is class scheduled.

Null otherwise.

JPI_SHRFILLM	Integer	Maximum number of open shared files for the job to which the process belongs.
JPI_SITESPEC	Integer	Per-process site-specific integer.
JPI_SLOW_VP_SWITCH	Integer	Number of times process has issued a vector instruction that enabled a inactive vector processor with a fill vector context switch.
JPI_STATE	Integer	Process state:

State	Meaning
SCH_C_CEF	Common event flag wait
SCH_C_COLPG	Collided page wait
SCH_C_COM	Computable (Idle)
SCH_C_COMO	Computable, outswapped
SCH_C_CUR	Current process
SCH_C_FPG	Free page wait
SCH_C_HIB	Hibernate wait
SCH_C_HIBO	Hibernate wait, outswapped
SCH_C_IOR	Blocked for I/O read
SCH_C_IOW	Blocked for I/O write
SCH_C_LEF	Local event flag wait
SCH_C_LEFO	Local event flag wait, outswapped
SCH_C_MWAIT	Mutex and miscellaneous resource wait
SCH_C_PFW	Page fault wait
SCH_C_RUN	Running, but not current process (on multi-processor systems)
SCH_C_SUSP	Suspended
SCH_C_SUSPO	Suspended, outswapped

JPI_STS	Integer	First 32-bits of process status flags, A bitmask containing zero or more of the following:

Flag	Meaning
PCB_V_ASTPEN	AST pending
PCB_V_BATCH	Process is a batch job
PCB_V_DELPEN	Delete pending
PCB_V_DISAWS	Disable automatic working set adjustment
PCB_V_FORCPEN	Force exit pending
PCB_V_HARDAFF	Process bound to a particular CPU
PCB_V_HIBER	Hibernate after initial image activate
PCB_V_INQUAN	Initial quantum in progress
PCB_V_INTER	Process is an interactive job
PCB_V_LOGIN	Log in without reading authorization file
PCB_V_NETWRK	Process is a network connect object
PCB_V_NOACNT	No accounting for process
PCB_V_NODELET	No delete
PCB_V_PHDRES	Process header resident
PCB_V_PREEMPTED	Kernel mode suspend has overridden supervisor mode suspend.
PCB_V_PSWAPM	Process swap mode (1=noswap)
PCB_V_PWRAST	Power fail AST

		PCB_V_RECOVER Process can recover locks PCB_V_RES Resident, in balance set PCB_V_RESPEN Resume pending, skip suspend PCB_V_SECAUDIT Mandatory security auditing PCB_V_SOFTSUSP Process is in supervisor mode suspend PCB_V_SSFEXC System service exception enable (kernel) PCB_V_SSFEXCE System service exception enable (exec) PCB_V_SSFEXCS System service exception enable (super) PCB_V_SSFEXCU System service exception enable (user) PCB_V_SSRWAIT System service resource wait disable PCB_V_SUSPEN Suspend pending PCB_V_WAKEPEN Wake pending, skip hibernate PCB_V_WALL Wait for all events in mask
JPI_STS2	Integer	Second 32-bits of process status flags. A bitmask containing zero of more of the following: **Flag** **Meaning** PCB_V_NOUNSHELVE Process does not automatically unshelve files
JPI_TABLENAME	String	File specification of process CLI table containing valid UCL commands.
JPI_TERMINAL	String	Login terminal name for interactive users.
JPI_TMBU	Integer	Termination mailbox unit number.
JPI_TOKEN	Integer	Token size (TRADITIONAL or EXPANDED).
JPI_TQCNT	Integer	Remaining timer queue entry quota.
JPI_TQLM	Integer	Timer queue quota.
JPI_TT_ACCPORNAM	String	Access port name for terminal associated with process.
JPI_TT_PHYDEVNAM	String	Physical device name of the terminal asssociated with process.
JPI_UAF_FLAGS	Integer	User authorization file (UAF) flags for user who owns process.
JPI_UIC	Integer	User ID code.
JPI_USERNAME	String	User name of process.
JPI_VIRTPEAK	Integer	Peak virtual address size, in bytes.
JPI_VOLUMES	Integer	Count of current privately mounted volumes.
JPI_VP_CONSUMER	Integer	Flag indicating if the process is a vector consumer.
JPI_VP_CPUTIM	Integer	Total amount of time, in nanoseconds, process has accumulated as a vector consumer.
JPI_WSAUTH	Integer	Maximum authorized working set size.
JPI_WSAUTHEXT	Integer	Maximum authorized working set extent.
JPI_WSEXTENT	Integer	Current working set extent.
JPI_WSPEAK	Integer	Working set peak.
JPI_WSQUOTA	Integer	Working set size quota.
JPI_WSSIZE	Integer	Current working set limit.

Description

GETJPI returns process/job information about a specified process. By using the context, the caller can

iterate through processes with the PROCESS_SCAN service.

The calling process must have WORLD privilege to obtain information about other processes that are not part of the current job.

Affected Quotas
None

Condition Values Returned

SS$_NORMAL	The service completed successfully.
SS$_ACCVIO	The address to receive the time cannot be written to.
SS$_BADPARAM	An invalid item code was passed in a descriptor.
SS$_NONEXPR	Specified process not found.
SS$_NOMORENODE	No more nodes matching the filters for the context were found.
UOSErr_Invalid_Context	The passed context was not a valid process context.

GETSTREAMINDEX

GETSTREAMINDEX

Get Stream index from name.

The GETSTREAMINDEX returns the index of an ancillary data string given a name.

Format
SYS$GETSTREAMINDEX fab src, dst

Arguments
fab
The 64-bit address of a FAB block.

srb
The 64-bit address of a SRB structure that points to the stream name.

dst
The 64-bit address of a 64-bit integer to receive the index corresponding to the passed name. completes.

System Request Structure

Field	Value
Subsystem	FIP
Request ID	UOS_FIP_Get_Stream_Index
Length	24
Status	
Flags	Ignored
Integer1	Pointer to FAB control block
Integer2	Pointer to SRB structure.
Integer3	Pointer to integer to receive result index.

Description
When successfully invoked, the GETSTREAMINDEX service returns the index of the stream associated with the passed name. The FAB must correspond to an open file. The file must be on a store with a file struture that supports ancillary data streams or else an error will result. For instance, this service fails if used on a

terminal device.

Quotas Affected
None

Privileges Required
None.

Condition Codes Returned

RMS$_NORMAL	Successful completion.
UOSErr_Missing_Value	No FAB address was supplied.
RMS$_FAB	FAB was incorrect format (size or code).
UOSErr_Quota_Exceeded	Buffered I/O limit was exceeded.
UOSErr_Invalid_Handle	FAB does not contain a valid file handle.
UOSErr_Invalid_Operation	The handle is not associated with a file on a file system that supports ancillary data streams.

GETSTREAMLENGTH

GETSTREAMLENGTH

Return length of stream.

This service returns the length of an ancillary data stream, in bytes.

Format

SYS$GETSTREAMLENGTH fab stream dst

Arguments

fab

The 64-bit address of a FAB block.

stream

The 64-bit address of a 64-bit integer containing the index of the stream whose length is to be returned.

dst

The 64-bit address of a 64-bit integer to receive the index corresponding to the passed name. completes.

System Request Structure

Field	Value
Subsystem	FIP
Request ID	UOS_FIP_Get_Stream_Length
Length	24
Status	
Flags	Ignored
Integer1	Pointer to FAB control block
Integer2	Pointer to integer containing the stream index.
Integer3	Pointer to integer to receive result index.

Description
When successfully invoked, the GETSTREAMLENGTH service returns the length of the specified stream. The FAB must correspond to an open file. The file must be on a store with a file struture that supports ancillary data streams or else an error will result. For instance, this service fails if used on a terminal device. The stream index must correspond to an existing stream.

Quotas Affected
None

Privileges Required
None.

Condition Codes Returned

RMS$_NORMAL	Successful completion.
UOSErr_Missing_Value	No FAB address was supplied.
RMS$_FAB	FAB was incorrect format (size or code).
UOSErr_Quota_Exceeded	Buffered I/O limit was exceeded.
UOSErr_Invalid_Handle	FAB does not contain a valid file handle.
UOSErr_Invalid_Operation	The handle is not associated with a file on a file system that supports ancillary data streams.

GETSTREAMNAME

GETSTREAMNAME
Return name of stream

Returns the name of an ancillary stream associated with an index.

Format
SYS$GETSTREAMNAME FAB Idx Dst Len

Arguments

FAB
An address of a File Access Block structure for an open file.

Idx
An address of a 64-bit integer indicating the stream index for which to return a name.

Dst
An address of a buffer where to write the name.

Len
An address of a 64-bit integer value indicating the maximum length of the receiving buffer, in bytes. Upon return, the actual length of data returned is written to this address. The name returned for stream index 0 (the default data stream), is a null string.

Description
This service returns the name of the stream associated with the passed index in an open file. The file must be currently open, and the index must match a valid ancillary data stream index. If the length of the stream name exceeds the passed buffer length, the result is trimmed to that length. The returned length will always be the number of bytes actually written to the result buffer, which may be less than

the length of the stream name.

Condition codes returned

Code	Meaning
RMS_FAB	The FAB block has an invalid layout.
RMS_BLN	The FAB block has an invalid length.
LIB_INVOPER	Operation attempted on a non-file resource.
SS_ACCVIO	Memory access violation.
SS_NORMAL	Successful completion.
UOSErr_Invalid_Handle	The FAB does not contain an valid file handle.

GETSYI

GETSYI
Get System Information

GETSYI returns information about the local system or about a node within a cluster.

Format
SYS$GETSYI efn csiadr name list iosb astadr astprm

Return Value
Either an integer or a string, depending upon the item requested. The table below indicates the valid items and the corresponding return values.

Arguments
efn
Event flag to set upon completion.

csiadr
The address of memory containing the Cluster ID of the node for which to return information. If 0, or if the memory contains 0, the name parameter is used to determine the node.

name
Address of TSRB for the name of the node for which to return information. If 0, or if the string is null, the current node is assumed. Note that a non-zero Cluster ID will override this.

list
Address of a descriptor list containing the code(s) indicating what information to return. This memory contains one or more item descriptors, terminating with a null item descriptor. The valid codes are listed in the table below.

name
Address of TSRB for the name of the node for which to return information. If 0, or if the string is null, the current node is assumed. Note that a non-zero Cluster ID will override this.

iosb
Address of an IO Status Block to receive completion status.

astadr
Address of an AST routine to call upon completion. If 0 is passed, no AST is called.

astprm

Parameter to pass to the AST routine specified with the astadr parameter.

System Request Structure

Field	Value
Subsystem	Kernel
Request ID	UOS_Kernel_GetSYI
Length	56
Status	
Flags	Ignored
CSID	8 bytes. Cluster ID.
List	8 bytes. Address of item list.
Length	8 bytes. Ignored
Flags2	4 bytes. Ignored
Buffer	8 bytes. Address of buffer containing the node name.
Length	8 bytes. Length of node name.
Flags	4 bytes. Ignored.
Inte	8 bytes. Pointer to IOSB.

Descriptors

Offset byte	Size in bytes	Description
0	2	Must be -1
2	2	Item code
4	4	Must be -1
8	8	Result buffer address
16	8	Result buffer size
24	8	Result length

Description
The Get System Information service returns information about the local system or about other nodes in a cluster. Except where otherwise noted, each item returns a 64-bit integer value. The following are the valid item codes:

Code	Result type	Description
SYI_ACTIVE_CPU_BITMAP	Special	A bitmask with each bit indicating a member of the CPUs partipating in UOS scheduling activities. If a given bit is set, that CPU is in the active set. The size of the result is determined by the number of CPUs supported by the system. To determine the number of bytes required, use SYI_MAX_CPUS, round that return value up to a multiple of 64, and divide that by 8.
SYI_ACTIVE_CPU_MASK	Integer	Equivalent to SYI_ACTIVE_CPU_BITMAP but only returns the first 64 CPUs (8 bytes).
SYI_ACTIVECPU_CNT	Integer	The count of active CPUs.
SYI_ARCHFLAG	Integer	Architecture flags for the system.
SYI_ARCH_NAME	String	Name of the CPU architecture, as a character string.
SYI_ARCH_TYPE	Integer	Type of the CPU architecture.
SYI_AVAIL_CPU_BITMAP	Special	A bitmask indexed by CPU number. If a given bit is set, that CPU is in the active set and participating in scheduling activities. The size of the result is determined by the

		number of CPUs supported by the system. To determine the number of bytes required, use SYI_MAX_CPUS, round that return value up to a multiple of 64, and divide that by 8.
SYI_AVAIL_CPU_MASK	Integer	The same as SYI_AVAIL_CPU_BITMAP except that only 64 CPUs are supported (a single 64-bit integer result).
SYI_AVAILCPU_CNT	Integer	The count of CPUs available to the system.
SYI_BOOT_DEVICE	String	Name of the device that UOS was booted from, as a character string.
SYI_BOOTTIME	Integer	Timestamp of when UOS was booted.
SYI_CHARACTER_EMULATED	Integer	1 if this is a VAX with character instruction set emulation, 0 otherwise.
SYI_CLUSTER_EVOTES	Integer	Total number of votes in the cluster.
SYI_CLUSTER_FSYSID	Integer	System ID for the first node to boot in the cluster (the founding node).
SYI_CLUSTER_FTIME	Integer	Timestamp of when the first node in the cluster was booted.
SYI_CLUSTER_MEMBER	Integer	1 if the node is the member of a cluster, 0 otherwise.
SYI_CLUSTER_NODES	Integer	Total number of nodes in the cluster.
SYI_CLUSTER_QUORUM	Integer	Total quorum for the cluster.
SYI_CLUSTER_VOTES	Integer	Total number of votes in the cluster. This is the value of the system parameter VOTES.
SYI_CONSOLE_VERSION	Integer	Console firmware version.
SYI_CONTIG_GBLPAGES	Integer	Total number of free, contiguous global pages.
SYI_CPU	Integer	Processor type. For extended information about processor types, see the SYI_XCPU item.
SYI_CPU_AUTOSTART	Special	A list of CPUs that will brought into the active set if it transitions into the current instance from outside or is powered up while owned. A bitmask indexed by CPU. Any set bit indicates the CPU will be brought into the active set.
SYI_CPUCAP_MASK	Special	An array of 64-bit integers, indexed by CPU. Each value is a bitmask indicating CPU capabilities. To determine the amount of room for the result buffer, in bytes, use the number of CPUs from SYI_AVAILCPU_CNT, and multiply by 8.
SYI_CPUTYPE	Integer	The processor type. See the CPU_* constants.
SYI_CWLOGICALS	Integer	1 if the clusterwide logical name database has been initialized on the system, 0 otherwise.
SYI_DAY_OVERRIDE	Integer	Returns 1 if SET DAY has been used to override the default primary and secondary day types in the UAF that are used to control user logins. 0 if the the UAF file records are honored for each user.
SYI_DAY_SECONDARY	Integer	Returns 1 if SET DAY has been used to specify that the current day is to be considered a Secondary day for login purposes. 0 if the current day is considered a Primary day for login purposes. If the SYI_DAY_OVERRIDE item returns 0, the number returned by this item is meaningless.
SYI_DECIMAL_EMULATED	Integer	Returns 1 if this is a VAX CPU with decimal instruction set emulation, 0 otherwise.
SYI_DECNET_FULLNAME	String	Node name.
SYI_DECNET_VERSION	Integer	Network version.
SYI_D_FLOAT_EMULATED	Integer	Returns 1 if this is a VAX with D Float instruction emulation, 0 otherwise.
SYI_DEF_PRIO_MAX	Integer	Maximum priority for the default scheduling policy.
SYI_DEF_PRIO_MIN	Integer	Minimum priority for the default scheduling policy.
SYI_ERLBUFFERPAG_S2	Integer	Number of system pages used for each S2 errorlog buffer.

SYI_ERRORLOGBUF_S2	Integer	Number of S2 errorlog buffers.
SYI_ERLBUFFERPAGES	Integer	Number of system pages used for each S0 errorlog buffer.
SYI_ERRORLOGBUFFERS	Integer	Number of S0 errorlog buffers.
SYI_F_FLOAT_EMULATED	Integer	Returns 1 if this is a VAX with F Float instruction emulation, 0 otherwise.
SYI_FREE_GBLPAGES	Integer	Current count of free global pages. The system parameter GBLPAGES sets the maximum global pages.
SYI_FREE_GBLSECTS	Integer	Current count of free global section table entries. The system parameter GBLSECTIONS sets the maximum global section entries.
SYI_FREE_PAGES	Integer	Total number of free pages.
SYI_G_FLOAT_EMULATED	Integer	Returns 1 if this is a VAX with G Float instruction emulation, 0 otherwise.
SYI_H_FLOAT_EMULATED	Integer	Returns 1 if this is a VAX with H Float instruction emulation, 0 otherwise.
SYI_HP_ACTIVE_CPU_CNT	Integer	The count of CPUs in the hard partition that are not in firmware console mode.
SYI_HP_ACTIVE_SP_CNT	Integer	The count of active UOS instances currently executing within the hard partition.
SYI_HP_CONFIG_SP_CNT	Integer	The maximum count of soft partitions within the current hard partition.
SYI_HW_MODEL	Integer	System model type.
SYI_HW_NAME	String	System model name as a character string.
SYI_IO_PRCPU_BITMAP	Special	A bitmask with each bit indicating a member of the CPUs partipating in UOS scheduling activities. The bit that is set indicates the preferred CPU available for Fast Path operations. The size of the result is determined by the number of CPUs supported by the system. To determine the number of bytes required, use SYI_MAX_CPUS, round that return value up to a multiple of 64, and divide that by 8.
SYI_IO_PREFER_CPU	Integer	This is the same as the SYI_IO_PRCPU_BITMAP item except that only the first 64 CPUs are returned (one 64-bit integer).
SYI_MAX_CPUS	Integer	The maximum number of CPUs that can be recognized by the system.
SYI_MAX_PFN	Integer	The highest numbered PFN in use by UOS. This may be influenced by the PHYSICAL_MEMORY system parameter.
SYI_MEMSIZE	Integer	Number of pages of memory available to UOS.
SYI_MODIFIED_PAGES	Integer	Number of modified pages.
SYI_MULTITHREAD	Integer	Value of the MULTITHREAD system parameter.
SYI_NODE_AREA	Integer	Network area for the node.
SYI_NODE_CSID	Integer	Cluster ID of the node in the form of a string containing a hexadecimal value.
SYI_NODE_EVOTES	Integer	Number of votes alloted to the node.
SYI_NODE_HWVERS	Integer	Hardware version of the specified node.
SYI_NODE_NUMBER	Integer	Network number for the specified node.
SYI_NODE_QUORUM	Integer	Node's quorum. This is derived from the node's sytem parameter EXPECTED_VOTES.
SYI_NODE_SWINCARN	Integer	Software incarnation number for the node in the form of a string containing a hexadecimal value.
SYI_NODE_SWTYPE	String	Type of UOS software for the node. This is a 4-byte ASCII string.
SYI_NODE_SWVERS	String	Software version of the specified node. This is a 4-byte

		ASCII string.
SYI_NODE_SYSTEMID	Integer	System ID.
SYI_NODE_VOTES	Integer	Number of votes alloted to the node. This is determined by the node's system parameter VOTES.
SYI_NODENAME	String	Node name (not including double colon).
SYI_NPAGED_FREE	Integer	Number of free bytes in the non-paged pool.
SYI_NPAGED_LARGEST	Integer	Size of largest contiguous area of free memory in the non-paged pool.
SYI_NPAGED_TOTAL	Integer	Total size (in bytes) of non-paged pool.
SYI_NPAGED_INUSE	Integer	Total number of bytes currently used in the non-paged pool.
SYI_PAGED_FREE	Integer	Number of free bytes in the paged pool.
SYI_PAGED_INUSE	Integer	Total number of bytes currently used in the paged pool.
SYI_PAGED_LARGEST	Integer	Size of largest contiguous area of free memory in the paged pool.
SYI_PAGED_TOTAL	Integer	Total size (in bytes) of non-paged pool.
SYI_PAGE_SIZE	Integer	Number of bytes in a physical page of memory.
SYI_PAGEFILE_FREE	Integer	Number of free pages in the currently installed paging files.
SYI_PAGEFILE_PAGE	Integer	Total number of pages in the currently installed paging files.
SYI_PARTITION_ID	Integer	Soft partition ID for systems that support partitioning.
SYI_PFN_MEMORY_MAP	Integer	Same as SYI_PFN_MEMORY_MAP64.
SYI_PFN_MEMORY_MAP64	Integer	Returns a structure defining the system's memory layout.
SYI_PHYSICALPAGES	Integer	Total number of PFNs.
SYI_PMD_COUNT	Integer	Total number of physical memory descriptors.
SYI_POTENTIAL_CPU_BITMAP	Special	A bitmask indexed by CPU number. If a given bit is set, that CPU is in the potential set. The size of the result is determined by the number of CPUs supported by the system. To determine the number of bytes required, use SYI_MAX_CPUS, round that return value up to a multiple of 64, and divide that by 8.
SYI_POTENTIAL_CPU_MASK	Integer	The same as SYI_POTENTIAL_CPU_BITMAP, but only returns the first 64 CPUS (one 64-bit integer).
SYI_PRESENT_CPU_BITMAP	Special	A number representing a bitmask indexed by CPU number. If a given bit is set, that CPU is in the present set. The size of the result is determined by the number of CPUs supported by the system. To determine the number of bytes required, use SYI_MAX_CPUS, round that return value up to a multiple of 64, and divide that by 8.
SYI_PRESENT_CPU_MASK	Integer	The same as SYI_PRESENT_CPU_BITMAP, but only for the first 64 CPUs (a 64-bit integer).
SYI_PRIMARY_CPUID	Integer	The ID of the primary processor for the node.
SYI_PROCESS_SPACE_LIMIT	Integer	64-bit virtual address after the last byte of process private address space.
SYI_PSXFIFO_PRIO_MAX	Integer	Maximum priority for the POSIX FIFO scheduling policy.
SYI_PSXFIFO_PRIO_MIN	Integer	Mimimum priority for the POSIX FIFO scheduling policy.
SYI_PSXRR_PRIO_MAX	Integer	Maximum priority for the POSIX round-robin scheduling policy.
SYI_PSXRR_PRIO_MIN	Integer	Mimimum priority for the POSIX round-robin scheduling policy.
SYI_PT_BASE	Integer	The virtual address of the base of the page tables.
SYI_PTES_PER_PAGE	Integer	Maximum number of CPU-specific pages that can be mapped by one page table page.
SYI_QUANTUM	Integer	Maximum amount of processor time a process can receive while other processes are waiting.

SYI_REAL_CPUTYPE	Integer	Actual CPU type of the primary CPU on the node.
SYI_SCS_EXISTS	Integer	Returns 1 to indicate if the system communication subsystem (SCS) is currently loaded on the node, 0 otherwise.
SYI_SID	Integer	System ID.
SYI_SWAPFILE_FREE	Integer	Number of free pages in currently installed swap files.
SYI_SWAPFILE_PAGE	Integer	Number of pages in the currently installed swap files.
SYI_SYSTEM_UUID	Special	The 128-bit Universal Unique Identifier for the node (16 bytes).
SYI_SYSTYPE	Integer	The family or system hardware platform.
SYI_TOTAL_PAGES	Integer	Total number of physical memory pages.
SYI_USED_GBLPAGCNT	Integer	Number of pages currently in use in the global page table.
SYI_USED_GBLPAGMAX	Integer	Maximum number of pages ever in use in the global page table.
SYI_USED_PAGES	Integer	Total number of used pages.
SYI_VERSION	String	UOS version, returned as a character string.
SYI_VECTOR_EMULATOR	Integer	Returns 1 if the vector instruction emulator facility (VVIEF) is installed on the node, 0 otherwise.
SYI_VP_MASK	Integer	Bitmask indicating which processors have vector coprocessors.
SYI_VP_NUMBER	Integer	Number of vector processors in the system.
SYI_XCPU	Integer	Extended CPU processor type.
SYI_XSID	Integer	Extended system type information.

Description
GETSYI returns system information about the current node or another node in a cluster.

Affected Quotas
ASTLM

Condition Values Returned

SS$_NORMAL	The service completed successfully.
SS$_ACCVIO	The address to receive the time cannot be written to.
SS$_BADPARAM	A descriptor contained an invalid item code.

GETTIM

GETTIM
Get Time

Returns the current system time in a 64-bit integer format.

Format
SYS$GETTIM adr

Argument
adr
The 64-bit address of a quadword to receive the current time in 64-bit format.

System Request Structure

Field	Value
Subsystem	Kernel
Request ID	UOS_Kernel_Get_Time
Length	8
Status	
Flags	Ignored
Int	8 bytes. 64-bit address to receive time.

Description
The Get Time service returns the current system time in 64-bit format. This the number of nanoseconds since Jan 1, 0, at midnight (projecting the Gregorian calander back).

The frequency that this value is updated depends upon the hardware of the system that UOS is running on. Nevertheless, each call will return a greater value than the previous call.

Required Privileges
None

Affected Quotas
None

Condition Values Returned

SS$_NORMAL	The service completed successfully.
SS$_ACCVIO	The address to receive the time cannot be written to.

GETUAI

GETUAI
Get User Authorization Information

Returns the current system time in a 64-bit integer format.

Format
SYS$GETUAI nullarg1 context username list nullarg2 nullarg3 nullarg4

Argument
nullarg1
Placeholder reserved for future use.

context
Address of a quadword context. If the value at the address is 0, a new context is written there and the first matching user is returned. Otherwise, the next matching user for the context is returned.

username
Address of a SRB structure which points to the username. If this is 0, all users are iterated through. This can contain wildcards.

list
Address of a descriptor list indicating what data to return for matching users. The list is an array of descriptors. The last descriptor in the list must be all 0's.

nullarg2 nullarg3 nullarg4
Placeholders reserved for future use.

System Request Structure

Field	Value
Subsystem	USC
Request ID	UOS_USC_GETUAI
Length	32
Status	
Flags	Ignored
SRB	16 bytes. SRB structure pointing to username.
Integer1	8 bytes. 64-bit address of context.
Integer2	8 bytes. 64-bit address of item list.

Descriptors

Each descriptor has the following layout:

Byte offset	Byte length	Description
0	2	Must be -1
2	2	Item code
4	4	Must be -1
8	8	Address of result buffer length.
16	8	Address of result buffer.
24	8	Address of result length.

Description

This service returns authorization information about users. The valid descriptor codes are as follows:

Code	Result type	Description
UAI_ACCOUNT	String	Account name
UAI_ACCOUNT_Length	Integer	Length of account name.
UAI_ASTLM	Integer	AST queue limit
UAI_BIOLM	Integer	Buffered I/O limit
UAI_BYTLM	Integer	Buffer I/O byte limit
UAI_CLITABLES	String	CLI table name for account
UAI_CLITABLES_LENGTH	Integer	CLI table name length for account
UAI_CPUTIM	Integer	Maximum CPU time limit per session
UAI_DEFCLI	String	Default shell filename
UAI_DEFCLI_LENGTH	Integer	Default shell filename length
UAI_DEFDEV	String	Name of default device
UAI_DEFDEV_LENGTH	Integer	Length of the name of default device
UAI_DEFDIR	String	Default directory name
UAI_DEFDIR_LENGTH	Integer	Default directory name length
UAI_DEF_PRIV	Integer	Default privileges
UAI_DFWSCNT	Integer	Default working set size in pages
UAI_DIOLM	Integer	Direct I/O count limit
UAI_ENCRYPT	Integer	Encryption algorithm ID for the primary password
UAI_ENCRYPT2	Integer	Encryption algorithm ID for the secondary password
UAI_ENQLM	Integer	Lock queue limit
UAI_EXPIRATION	Integer	Date/time of account expiration
UAI_FILLM	Integer	Open file limit

UAI_FLAGS	Integer	Account flags (bitmap consisting of the following):
	UAI_V_AUDIT	Audit all actions
	UAI_V_AUTOLOGIN	User can only log in via terminals defined by the ALF (Automatic Login Facility)
	UAI_V_CAPTIVE	Captive account
	UAI_V_DEFCLI	Only default shell
	UAI_V_DISACNT	Account is disabled
	UAI_V_DISCTLY	Disable ^Y
	UAI_V_DISIMAGE	Disable RUN
	UAI_V_DISRECONNECT	Disable reconnect
	UAI_V_DISREPORT	Do not show last login messages.
	UAI_V_DISWELCOME	Do not show welcome message
	UAF_DisAuth	Cannot change password
	UAI_V_RESTRICTED	Restricted account
UAI_JTQUOTA	Integer	Initial byte quota for job symbol table
UAI_LASTLOGIN_I	Integer	Date/time of last interactive login
UAI_LASTLOGIN_N	Integer	Date/time of last noninteractive login
UAI_LGICMD	String	Name of default login command file
UAI_LGICMD_LENGTH	Integer	Name of default login command file
UAI_LOGFAILS	Integer	Count of login failures since last login
UAI_MAXACCTJOBS	Integer	Max processes for all sessions of this user
UAI_MAXDETACH	Integer	Max detached processes
UAI_MAXJOBS	Integer	Max processes for each session
UAI_OWNER	String	Name of account owner
UAI_OWNER_LENGTH	Integer	Length of name of account owner
UAI_PBYTLM	Integer	Paged buffer I/O byte count
UAI_PGFLQUOTA	Integer	Paging file quote (in pages)
UAI_PRCCNT	Integer	Subprocess creation limit
UAI_PRI	Integer	Default base priority
UAI_PRIV	Integer	Privileges for user
UAI_PWD	Integer	Hashed primary password
UAI_PWD_DATE	Integer	Date/time of last password change (-1 means pre-expired)
UAI_PWD_LENGTH	Integer	Minimum password length
UAI_PWD_LIFETIME	Integer	Password lifetime in UOS time format
UAI_PWD2	Integer	Hashed secondary password
UAI_PWD2_DATE	Integer	Date/time of last secondary password change (-1 means pre-expired)
UAI_QUEPRI	Integer	Maximum job queue priority
UAI_SALT	Integer	Random password salt
UAI_SHRFILLM	Integer	Shared file limit
UAI_TQCNT	Integer	Timer queue entry limit
UAI_UIC	Integer	UIC
UAI_USER_DATA	String	Up to 255 bytes of customer information
UAI_WSEXTENT	Integer	Working set extent in pages
UAI_WSQUOTA	Integer	Working set quota in pages

Description

GETUAI returns authorization information about a specified user. By using the context, the caller can iterate through user accounts. A valid login can be checked by seeing if the UAI_V_DISUSER flag is 0.

Affected Quotas
None

Condition Values Returned

SS$_NORMAL	The service completed successfully.
SS$_ACCVIO	The address to receive the time cannot be written to.

LOOKUP

LOOKUP
File lookup

Format
SYS$LOOKUP name context reslen

Argument
name
The address of a SRB structure which points to the file specification to lookup.

context
Address of a quadword context. If the value at the address is 0, a new context is written there and the first matching user is returned. Otherwise, the next matching file for the context is returned.

reslen
Address a 8-byte integer buffer to hold the length of the filename result.

System Request Structure

Field	Value
Subsystem	FIP
Request ID	UOS_FIP_Lookup
Length	32
Status	
Flags	Ignored
SRB	16 bytes. SRB structure pointing to filename.
Integer1	8 bytes. 64-bit address of 8-byte context.
Integer2	8 bytes. 64-bit address of 8-byte result length buffer.

Description
The LOOKUP service begins a new file lookup if the contents of the 8-byte buffer pointed to by the context are 0, in which case a new context is written to that address. If non-zero, the next file matching the original lookup is looked up. The length of the result filename is written to the result length buffer. The LOOKUP_NAME service can be used to obtain that filename. Note that the LOOKUP_CLOSE service should be used to clean up the context when the series of lookups are completed.

Required Privileges
Depends upon file protection codes.

Affected Quotas
None

Condition Values Returned

SS$_NORMAL	The service completed successfully.
SS$_ACCVIO	The address to receive the context cannot be written to or read from.

LOOKUP_CLOSE

LOOKUP_CLOSE

Close File Lookup Context

Format

SYS$LOOKUP_CLOSE context

Argument

context

Address of a quadword context that was used for a lookup with the LOOKUP service.

System Request Structure

Field	Value
Subsystem	FIP
Request ID	UOS_FIP_Lookup
Length	16
Status	
Flags	Ignored
Integer1	8 bytes. 64-bit address of 8-byte context.
Integer2	8 bytes. 64-bit address of 8-byte result length buffer.

Description

The LOOKUP_CLOSE service deletes a file lookup context and frees all resources associated with that context. Once this call completes, the context is no longer valid for lookup operations.

Required Privileges

None

Affected Quotas

None

Condition Values Returned

SS$_NORMAL	The service completed successfully.
SS$_ACCVIO	The address of the context cannot be read from.
UOSErr_Invalid_Context	The context value is invalid.

LOOKUP_NAME

LOOKUP_NAME

Obtain Filename from last File lookup

Format

SYS$LOOKUP_NAME context buffer

Argument

context
Address of a quadword context that was used for a lookup with the LOOKUP service.

buffer
Address a buffer large enough to hold the filename. The buffer size must be at least as large as the value returned by the last LOOKUP service call.

System Request Structure

Field	Value
Subsystem	FIP
Request ID	UOS_FIP_Lookup
Length	16
Status	
Flags	Ignored
Integer1	8 bytes. 64-bit address of 8-byte context.
Integer2	8 bytes. 64-bit address of 8-byte result length buffer.

Description
The LOOKUP_NAME service writes the filename that was found by the last LOOKUP service for the specified context. This call does not advance the context to the next matching file.

Required Privileges
None

Affected Quotas
None

Condition Values Returned

SS$_NORMAL	The service completed successfully.
SS$_ACCVIO	The address of the context cannot be read from.
UOSErr_Invalid_Context	The context value is invalid.

OPEN

OPEN
Open a File

The OPEN service returns a handle through which a file can be accessed.

Format
SYS$OPEN fab err suc

Arguments
fab
The 64-bit address of a FAB block.

err
The 64-bit address of a callback routine that is called if an error occurs during the system call.

suc
The 64-bit address of a callback routine that is called when the operation completes.

System Request Structure

Field	Value
Subsystem	FIP
Request ID	UOS_FIP_Open
Length	24
Status	
Flags	Ignored
Integer1	Pointer to FAB control block
Integer2	Pointer to error completion routine
Integer3	Pointer to success completion routine

Description
A file must be opened to perform most file operations. If the appropriate XAB blocks are linked to the FAB, they are filled with information appropriate to the file. The CLOSE service will remove the handle and clean up associated data.

Note that if the SYS library function is called, the PARSE service is also called to set up the NAML block. If the service is called directly, the PARSE function should be manually called before to ensure that the NAML block is properly initialized.

If the file system supports last-access-timestamps, it will update that timestamp when the file is opened.

The following items are read or written from the FAB (also see the PARSE service for other affected fields):

Block	Field	R/W	Description
FAB	FAB_L_DNA	Read	Default file specification string address.
FAB	FAB_B_DNS	Read	Default file specification string length.
FAB	FAB_L_FOP	Read	Passed flags process by OPEN are: One of the following options must be chosen. **Flag** **Description** FAB_V_GET Allow read from file. FAB_V_PUT Allow PUT in RMS files. Read/write from binary files. FAB_V_TRN Allow file truncation. FAB_V_UPD Allow UPDATE in RMS files. Read/write from binary files. FAB_V_DEL Allow DELETE in RMS files. Read/write from binary files. Zero or more of the following options may be included. **Flag** **Description** FAB_V_CIF Create file if it doesn't exist
FAB	FAB_L_FOP	Write	One or more of the following flags are set if the open succeeds: **Flag** **Description** FAB_V_CTG File is contiguous. FAB_V_RCK File has read-checks. FAB_V_WCK File has write-checks.
FAB	FAB_Q_HANDLE	Write	File handle.
NAML	NAML_W_FID	Write	File ID from file system.
XABDAT	XAB_Q_ACC	Write	Last access timestamp.
XABDAT	XAB_Q_BDT	Write	Last backup timestamp.
XABDAT	XAB_Q_CDT	Write	Creation timestamp.
XABDAT	XAB_Q_EDT	Write	Expiration date/time.

XABDAT	XAB_Q_RDT	Write	Last modified timestamp.
XABFHC	XAB_Q_EOF	Write	End of file (logical file size).
XABFHC	XAB_Q_SIZ	Write	Size of file on store.
XABFHC	XAB_Q_USZ	Write	Uncompressed file size.
XABFHC	XAB_Q_USZ	Write	File clustersize.
XABFHC	XAB_L_CRE	Write	Creator UIC.
XABFHC	XAB_W_FLG	Write	File flags.
XABPRO	XAB_W_PRO	Write	File protection.
XABRDT	XAB_L_UIC	Write	Last modified timestamp.

Privileges Affected
The process must be allowed access via the file's protection code and/or ACLs, or the process must possess the READALL privilege (for read-access to the file) or the BYPASS privilege. GRPPRV and/or SYSPRV privileges allow the process to access the file via the file's group or system protection fields.

Quotas Affected
FILLM

Condition Codes

SS$_NORMAL	Service completed normally.
SS$_BADPARAM	No address was specified for the handle.
SS$_NOSUCHNODE	A node was specified that wasn't found.
UOSErr_Device_Not_Mounted	Specified device was not mounted.
UOSErr_Invalid_Filename	The filename is not valid for the file system.
UOSErr_Device_Not_File_Structured	Tried opening a file when the device is not file-structured.
UOSErr_Missing_Value	No FAB was provided (address was 0) or the FAB indicated that NAML contained the file name, but no NAML block was provided.
UOSErr_Device_Not_Found	The device was not found.

PARSE

PARSE
Parse a File Specification

This service is provided by the Starlet library. See LIB$SYS_PARSE.

PROCESS_SCAN

PROCESS_SCAN
Process Scan

Process_Scan is a system call that works with the GETJPI call. It is used to iterate through processes. Process_Scan creates a context that GETJPI uses - each call to GETJPI using that context will get the next matching process ID. Filters can be added to the context when it is created. Each filter is used to filter out undesired processes. When Process_Scan is called, it is passed an item list that indicates one or more filters to use (or no filters, if all process IDs are to be iterated over). Each filter consists of an item to compare, the value to compare against, and the type of comparison to do. Each item in the list is either a value descriptor or a reference descriptor.

Format

SYS$PROCESS_SCAN context {,list}

Arguments

context

The 64-bit address of the memory to receive a 64-bit context.

list

The 64-bit address of an array of descriptors indicating how to filter the processes. See below for descriptor layout and a list of valid codes. The array of descriptors must end with a descriptor that consists of nothing by 0s. Note that list can be 0 to indicate that all processes should be iterated through.

System Request Structure

Field	Value
Subsystem	USC
Request ID	UOS_USC_Process_Scan
Length	16
Status	
Flags	Ignored
Int1	8 bytes. Address of context.
Int2	8 bytes. Address of item list.

Descriptors

Two types of descriptors are used. Which one is used depends upon the item code.

Value Descriptors:

Offset byte	Size in bytes	Description
0	2	Item code
2	2	Must be 1
4	4	Must be -1
8	8	Value
16	8	Flags

Reference Descriptors:

Offset byte	Size in bytes	Description
0	2	Item code
2	2	Must be 1
4	4	Must be -1
8	8	Value length
16	8	Value address
24	8	Flags

The following item codes can be used for comparisons:

Item	Type	Description
PSCAN_ACCOUNT	Reference	User account name
PSCAN_AUTHPRI	Value	Authorized base priority
PSCAN_CURPRIV	Reference	64-bit privilege mask
PSCAN_HW_MODEL	Value	Hardware model number.
PSCAN_HW_NAME	Reference	Hardware name
PSCAN_JOBPRCCNT	Value	Subprocess count for entire job

PSCAN_JOBTYPE	Value	Job type.
PSCAN_KT_COUNT	Value	Kernel thread count
PSCAN_MASTER_PID	Value	PID of master process
PSCAN_MODE	Value	Process mode
PSCAN_MULTITHREAD	Value	Maximum thread count
PSCAN_NODE_CSID	Value	Node's cluster ID number (is 0 if not a cluster member)
PSCAN_NODENAME	Reference	Node's name
PSCAN_OWNER	Value	Process ID of immediate parent process
PSCAN_PRCCNT	Value	Subprocess count of process
PSCAN_PRCNAM	Reference	Process name
PSCAN_PRI	Value	Current process priority level
PSCAN_PRIB	Value	Base process priority level
PSCAN_STATE	Value	Process state
PSCAN_STS	Value	Process status
PSCAN_TERMINAL	Reference	Terminal name
PSCAN_USERNAME	Reference	User name

For instance, the PSCAN_ACCOUNT item can be used to compare the account name for the user associated with a process.

Description
Flags indicate the type of comparison to be made and other options. A single comparison type can be combined with one or more option flags. Trying to combine different comparisons in the same filter will result in something unexpected - or even an error. For instance, PSCAN_M_NEQ and PSCAN_M_GEQ together not only makes no sense, but would produce unexpected results or an exception.

The types of comparisons:

Comparison	**Meaning**
PSCAN_M_EQL	Equal to the value
PSCAN_M_LSS	Less than the value
PSCAN_M_LEQ	Less than or equal to the value
PSCAN_M_GTR	Greater than the value
PSCAN_M_GEQ	Greater than or equal to the value
PSCAN_M_NEQ	Not equal to the value

Thus, we could have a filter that would compare the account name of a process' user with "SYSTEM" and a comparison of PSCAN_M_NEQ to only iterate over processes whose account name is not "SYSTEM".

The following flags can be combined with each other and with one comparison value, although some combinations may not be useful.

Option	**Meaning**
PSCAN_M_BIT_ALL	All bits in the comparison value must be set in the item
PSCAN_M_BIT_ANY	Any bit in the comparison value is set in the item
PSCAN_M_CASE_BLIND	Perform string comparisons case-insensitive
PSCAN_M_OR	Logically or this filter with the next one
PSCAN_M_PREFIX_MATCH	Match against the portion of the item to the length of the specified comparison value
PSCAN_M_WILDCARD	Treat wildcard characters as wildcards for matching

Thus, we could use PSCAN_M_EQL or PSCAN_M_WILDCARD with the PSCAN_USERNAME item and a comparison value of "A*" to filter out all processes whose username doesn't start with "A". This particular filter could also be accomplished with a prefix match. Note that if we compared with "A*" but didn't include

the wildcard flag, no processes would match, because it would look for a user name that is literally "A*", but user names cannot contain asterisks.

When a context contains multiple filters, a process must meet every requirement. However, if the PSCAN_M_OR flag is used, then the filter comparison is logically *or*ed with the next filter. Any number of filters can be *or*ed together this way.

Required Privileges
None.

Affected Quotas
None

Condition Values Returned

SS$_NORMAL	The service completed successfully.
SS$_ACCVIO	The address to receive the context cannot be written to, or a descriptor references an address that cannot be read from.
SS$_BADPARAM	A descriptor has an invalid format.

QIO

QIO
Queue I/O Request

The QIO service queues a request to a handle associated with a device.

Format
SYS$QIO efn handle func iosb astadr astprm p1 p2 p3 p4 p5 p6

Arguments
efn
Event flags to set upon completion of the operation.

handle
Address of the handle on which to queue the operation.

func
The device-specific function to perform on the device. This value is a combination of a function and a subfunction. The IO_Function_Mask value can be used to mask the function. The rest of the value is the subfunction.

iosb
Address of the IO Status block to receive final completion status.

astadr
Asynchronous address to call upon completion. 0 if no callback.

astprm
Parameter to pass to asynchronous completion routine.

p1..p6
Values specific to the function being requested. Unused values should be 0.

System Request Structure

Field	Value

Subsystem	FIP
Request ID	UOS_FIP_QIOW
Length	96
Status	
Flags	Ignored
Integer1	Pointer to event flags
Integer2	Pointer to handle
Integer3	Function
Integer4	Address of AST routine
Integer5	Pointer to parameter to pass to AST routine
Integer6	p1
Integer7	p2
Integer8	p3
Integer9	p4
Integer10	p5
Integer11	p6
Integer12	Pointer to IOSB

Description
QIO handles device-level operations as opposed to file or file system operations.

Privileges Required
The required privileges depend upon which operations are requested. The calling process must also possess read and/or write access to the device.

Quotas Affected
ASTLM
BIOLM
BYTLM
DIOLM

Condition Codes Returned

SS$_NORMAL	Successful completion
UOSErr_Invalid_Operation	The requested function isn't valid for the device.

Other error codes can be returned from the specific device.

Terminal Functions

IO_SETMODE function

This function sets the terminal mode. The subfunctions are:

Subfunction	Description
IOM_CTRLYAST	Sets an AST for control-Y from the terminal. As soon as this AST is called, the mode is reset. p1 = the AST routine to call. p2 = the context to pass to the AST. p3 = the access mode of the call.
IOM_CTRLCAST	Sets an AST for control-C from the terminal. As soon as this AST is called, the mode is reset. p1 = the AST routine to call. p2 = the context to pass to the AST.

	p3 = the access mode of the call.
IOM_CLI_CTRLYAST	Sets a CLI AST for control-Y from the terminal. As soon as this AST is called, the mode is reset. p1 = the AST routine to call. p2 = the context to pass to the AST. p3 = the access mode of the call.
IOM_CLI_CTRLCAST	Sets a CLI AST for control-C from the terminal. As soon as this AST is called, the mode is reset. p1 = the AST routine to call. p2 = the context to pass to the AST. p3 = the access mode of the call.

IO_READPROMPT function

This function reads from the termianl, optionally writing a prompt to the device first. There are no subfunctions, Parameters:

p1	Receiving buffer address
p2	Receiving buffer size, in bytes. The actual number of bytes written to the buffer is returned in IOSB.r_io_64.r_bcnt_32.l_bcnt.
p3	unused
p4	unused
p5	Prompt buffer address
p6	Prompt buffer length

IO_WRITEPBL function

This function writes a string to the terminal. There are no subfunctions. Parameters:

p1	Text buffer address
p2	Text buffer size, in bytes.
p3	unused
p4	unused
p5	unused
p6	unused

READ

READ
Read From a File

The READ service retrieves the specified number of bytes from a file into a buffer.

Format

SYS$READ rab err suc

Arguments

fab

The 64-bit address of a RAB block.

err

The 64-bit address of a callback routine that is called if an error occurs during the system call.

suc
The 64-bit address of a callback routine that is called when the operation completes.

System Request Structure

Field	Value
Subsystem	FIP
Request ID	UOS_FIP_Read
Length	24
Status	
Flags	Ignored
Integer1	Pointer to RAB control block
Integer2	Pointer to error completion routine
Integer3	Pointer to success completion routine

Description
To read data from a file, you must allocate a buffer large enough to receive the data and set up a RAB structure properly. Reading from a file requires that the file be opened with the OPEN service first so that file handle has been associated with that file.

The following fields are accessed in the RAB and associated structures.

Block	Field	Description
RAB	RAB_W_ISI	File handle for open file to read from.
RAB	RAB_W_USZ	Amount of data to read.
RAB	RAB_Data_Stream	Which file stream to read. The main file data stream is 0.
RAB	RAB_L_UBF	Address of buffer to write data into.
RAB	RAB_L_BKT	Position to read from. If the file is open in binary mode, 0 indicates reading from the first position of the file. If an RMS file, 0 indicates to read the file starting after the last byte read via the handle.

Privileges Affected
None.

Quotas Affected
BIOLM
BYTLM

Condition Codes Returned

SS$_NORMAL	Successful completion
UOSErr_Invalid_Handle	Handle passed was not a valid file handle or was not associated with this process.
UOSErr_Quota_Exceeded	The operation caused a quota to be exceeded.

SEARCH

SEARCH
Search for a File

The SEARCH system service scans a directory looking for matches to a file specification and then sets the NAML block appropriately. The PARSE service should be called prior to this service so that the NAML block is correctly initialized.

Format

SYS$SEARCH fab err suc

Arguments

fab

The 64-bit address of a FAB block.

err

The 64-bit address of a callback routine that is called if an error occurs during the system call.

suc

The 64-bit address of a callback routine that is called when the operation completes.

Description

The SEARCH service searches for a file, or files, matching a specification. If a match is found, the service sets the appropriate fields of the NAML block associated with the passed FAB block. The following table indicates the input/output fields used by this service.

Block	Field	R/W	Description
FAB	FAB_L_NAM	Read	Address of the NAML block.
FAB	FAB_L_STS	Write	Completion status code.
NAML	NAML_L_LONG_RESULT	Read	Address of where to write the matching file name
NAML	NAML_L_LONG_RESULT_SIZE	Write	Length of result string.
NAML	NAML_L_LONG_RESULT_ALLOC	Read	Maximum length of result string.
NAML	NAML_L_LONG_DEV	Read	Device name.
NAML	NAML_L_LONG_DEV_SIZE	Read	Device name length.
NAML	NAML_L_LONG_DIR	Read	Path.
NAML	NAML_L_LONG_DIR_SIZE	Read	Path length.
NAML	NAML_L_LONG_NAME	Read	File's name
NAML	NAML_L_LONG_NAME_SIZE	Read	Length of file name.
NAML	NAML_L_LONG_NODE	Read	Node name.
NAML	NAML_L_LONG_NODE_SIZE	Read	Length of node name.
NAML	NAML_L_LONG_TYPE	Read	File's extension.
NAML	NAML_L_LONG_TYPE_SIZE	Read	Length of file extension.
NAML	NAML_L_LONG_VER	Read	File's version.
NAML	NAML_L_LONG_VER_SIZE	Read	Length of file version.
NAML	NAML_L_USER_CONTEXT	Read/ Write	On a new search, the new context is written here. Otherwise, it provides the context for the next lookup.

Note: This service is an alternative means of looking up files, provided for compatibility with VMS. It calls the LOOKUP, LOOKUP_NAME, and LOOKUP_CLOSE services.

Required Privileges

None.

Affected Quotas

None

Condition Values Returned

RMS_SUC	The service completed successfully.
RMS_FAB	The FAB block is invalid.

RMS_NAML The NAML block is invalid.

SET_CONTIGUOUS

SET_CONTIGUOUS

Change contiguous state of file

Sets an existing file to contiguous or non-contiguous layout on a store.

Format

SYS$Set_Contiguous Name Value

Arguments

Name

An address of a SRB that points to the name of the file to affect.

Value

An address of a 64-bit integer that is 0 to set the file to non-contiguous or 1 to set the file to contiguous.

Description

This service alters the contiguous state of the specified existing file. When converting from non-contiguous to contiguous, there must be a contiguous area on the store large enough to contain the entire file. When converting from contiguous to non-contiguous, there must be enough non-contiguous space on the store to contain the allocation chain for the file. If there is insufficient space on the store to perform this operation, the service fails without returning an error. Files on sequential stores are always inherently contiguous and are unaffected by this service.

Condition codes returned

Code	Meaning
SS_NOSUCHNODE	An invalid node was specified.
SS_ACCVIO	Memory access violation.
SS_NORMAL	Normal completion of service.
UOSErr_Device_Not_File_Structured	A non-file was specified.

SET_NODE_NAME

SET_NODE_NAME

Set computer's node name.

This service changes the computer's node name.

Format

SYS$SET_NODE_NAME name

Parameters

name

Address of an SRB that points to the name to use as the node name for the computer. The node name must begin with an alphabetic character and can contain alpha, numeric, dollar sign ($), and underscore (_) characters. The name should be unique on the network/cluster.

Description

The SYS_SET_NODE_NAME service sets the current node name of the computer. This happens immediately and might terminate existing network connections.

Condition codes returned

SS_NORMAL
SS_NOPRIV

Privileges required

PHY_IO

SETPRV

SETPRV

Set Process Privileges

The SETPRV service enables or disabled the specified privileges for the calling process.

Format

SYS$SETPRV enflag privaddr permanent curprivaddr

Arguments

enflag

If 0, the passed privileges are disabled. If 1, the passed privileges are enabled. If any other value is passed, an error is returned.

privaddr

Address of the 64-bit privilege mask indicating which privileges to enabled or disable.

permanent

Indicates whether the privilege change should affect the current image (value of 0), or the process privileges (value of 1.) In the case of permanent privileges being affected, both the process privileges and current privileges are affected.

curprivaddr

Address of where to return the current privilege mask prior to the requested changes. If 0, the current privileges are not returned. The value of the permanent parameter indicates if the current image privileges or the current process privileges are returned.

System Request Structure

Field	Value
Subsystem	USC
Request ID	UOS_USC_SetPrv
Length	32
Status	
Flags	Ignored
Integer1	8 bytes. 64-bit address of 8-byte integer that indicates the enabled/disabled flag.
Integer2	8 bytes. 64-bit address of 8-byte privilege mask.
Integer3	8 bytes. 64-bit address of 8-byte integer that indicates the permanent flag.
Integer4	8 bytes. 64-bit address of 8-byte integer where the current privilege mask is written.

Description

This system service enables or disabled privileges. When a process is created, the user authorized privileges, process authorized privileges, and current privileges are all set to the UAF privilege mask defined for the user (thus, they are all the same).

SETPRV cannot change the user authorized privileges. Only privileges existing in the user authorized privileges can be set, unless the user authorized privileges include the SETPRV privilege. Any other privileges not in the user authorized privileges bitmask cannot be changed by this service.

When an image is run, any privileges assigned to that image are combined with the current process privileges until the image ends - at which point the process privileges are copied to the current privileges. While the image is running, this service can only be used to modify privileges possessed in the user authorized privileges and/or the privileges authorized for the image.

Condition Values Returned

SS$_NORMAL	All privileges were enabled or disabled as requested.
SS$_NOTALLPRIV	One or more of the specified privileges were not changed.
SS$_IVSTSFLG	A value other than 0 or 1 was passed to the permanent or enflag parameters.

TRNLNM

TRNLNM

Translate Symbol

TRNLNM returns information about a symbol name.

Format

SYS$TRNLNM flags table name acmode itemlst

Arguments

flags

Address of 64-bit flag to control the operation of TRNLNM. If this is 0, no flags are considered to be set. The following flags can be specified:

Flag	Description
LNM_M_CASE_BLIND	If set, TRNLNM doesn't distinguish between upper and lower case characters in the symbol name. By default, TRNLNM matches by case.
LNM_INTERLOCKED	If set, TRNLNM will wait to complete until any current clusterwide symbols are complete.

table

The address of an SRB structure pointing to the name of a symbol table. If this is 0 or the symbol table name is null, the process, job, group, system, and cluster symbol tables are searched, in that order.

name

Address of an SRB structure that points to the symbol name to translate.

acmode

The address of the 64-bit access mode to be used in the translation. If this is 0, the access mode is assumed to the outermost mode (USER). When provided, all symbols with an access mode less privileges than this are ignored. The valid access modes are:

Mneumonic	Description
PSL_C_KERNEL	Kernel (executive) mode. Ring 0.

PSL_C_DRIVER	Driver mode. Ring 1.
PSL_C_EXEC	Synonym for PSL_C_DRIVER.
PSL_C_USER	User mode. Ring 3.
PSL_C_SUPER	Supervisor mode. Ring 2.

itemlst

A pointer to a descriptor list containing items defining what the service should return.
A descriptor list is an array of descriptors with the following layout:

System Request Structure

Field	Value
Subsystem	SSC
Request ID	UOS_SSC_TRNLNM
Length	56
Status	
Flags	Ignored
SRB1	16 bytes. SRB structure pointing to the table name.
SRB2	16 bytes. SRB structure pointing to the symbol name.
Integer1	8 bytes. Flags.
Integer2	8 bytes. Access mode.
Integer3	8 bytes. 64-bit address of descriptor list.

Descriptors

Byte offset	Byte length	Description
0	2	Must be -1.
2	2	Item code.
4	4	Must be -1.
8	8	Length of result buffer.
16	8	Address of result buffer.
24	8	Address where to write result length.

Item codes

Code	Description
LNM_ACMODE	Returns the access mode associated with the specified symbol.
LNM_ATTRIBUTES	Returns the attributes of the specified symbol. The following mnuemonics can be used to compare with these attributes.
LNM_M_CONCEALED	The matching symbol is a concealed logical.
LNM_M_CONFINE	The symbol will not be copied to a spawned process.
LNM_M_CRELOG	The symbol was created using the CRELOG system service.
LNM_M_EXISTS	The symbol exists.
LNM_M_NO_ALIAS	The symbol cannot be defined in the same table with an outer access mode.
LNM_M_TABLE	The symbol is a symbol table name.
LNM_M_CLUSTERWIDE	The symbol was found in a clusterwide table.
LNM_M_TERMINAL	The symbol is not subject to recursive translation.
LNM_CHAIN	This is ignored by UOS. It is provided for VMS compatibility.
LNM_INDEX	Defines which equivalence value is to be returned for following descriptors. If not

	specified, index 0 is used. This can be used to return several equivalence values by alternating LNM_INDEX and LNM_STRING items, changing the index with each one.
LNM_LENGTH	Returns the length of the equivalence value for the specified symbol.
LNM_MAX_INDEX	The maximum index for equivalence values for the specified symbol.
LNM_NAME	Returns the case-sensitive name of the symbol matching the one specified.
LNM_STRING	Returns the equivalence name (value) for the specified symbol.
LNM_TABLE	Returns the name of the table containing the symbol name being translated.

Description
The TRNLNM system service returns information about the given symbol name.

Condition Codes

SS$_NORMAL	Service completed normally.
SS$_BADPARAM	Null symbol name passed or an invalid item code was encountered.
SS$_BUFFEROVF	Output buffer is too small to receive data.
SS$_IVLOGTAB	Symbol table not found.
SS$_NOLOGNAM	Symbol not found.

WRITE

WRITE

The WRITE service transfers a specified number of bytes to a file or device.

Format
SYS$WRITE rab, err, suc

Arguments
rab
RAB structure that defines what to write.

err
Address of AST routine to call if the operation is unsuccessful. If non-zero, this overrides RAB_AST_Err field of the RAB.

suc
Address of AST routine to call when the operation completess successfully. If non-zero, this overrides RAB_AST_Success field of the RAB.

Description
The address of the data and its size must be passed to the service. Writing to/past the end of a file results in the extending the file. The following RAB fields are used by this service:

RAB field	Description
RAB_Size	Byte size of the RAB structure
RAB_AST_Err	Address of AST routine to call on error
RAB_AST_Success	Address of AST routine to call when operation completes successfully
RAB_Data_Stream	Data stream to write to
RAB_L_BKT	File offset to write to
RAB_W_ISI	Handle to open file
RAB_B_RAC	Access type. 0 = binary file access.

RAB_L_ROP	Record processing options (see RAB_V_*)
RAB_L_UBF	Address of data to write
RAB_W_USZ	Number of bytes to write

Condition Values Returned

RMS_RAB
UOSErr_Invalid_Handle
UOSErr_Memory_Address_Error
UOSErr_Quota_Exceeded

Other errors specific to the device or file