

## SPECIFICATION

### WILD

#### WILD CARD LOOKUP ROUTINE

#### ABSTRACT

WILD is designed to be used as a subroutine called from any program which needs to perform wild-card LOOKUPS or ENTERS on any device. It is needed since the logic for selecting and reading directories has become quite involved under the 5-series monitor. This document is a complete specification describing how to call WILD and what the results are.

This document reflects the software as of Version 5.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this manual.

The software described in this document is furnished to purchaser under a license for use on a single computer system and can be copied (with inclusion of DIGITAL's copyright notice) only for use in such system, except as may otherwise be provided in writing by DIGITAL.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by DIGITAL.

Copyright (C) 1972,1973,1974 Digital Equipment Corp., Maynard, Mass.

## CONTENTS

1.0	INTRODUCTION	5
1.1	FUNCTION	5
1.2	GLOSSARY	6
1.3	REFERENCES	11
2.0	FILE SPECIFICATION AREA	12
2.1	GENERAL PHILOSOPHY	12
2.2	DETAILED LAYOUT	12
2.3	DEFAULTS	13
3.0	WILD-CARD LOOKUP SUBROUTINE CALL FORMAT	14
3.1	GENERAL PHILOSOPHY	14
3.2	DETAILED LAYOUT	15
3.3	DEFAULTS	16
3.4	SECONDARY ENTRY POINTS	16
3.5	INTERNAL FLAGS	17
3.6	STRUCTURE LIST SUBROUTINES	17
4.0	SECONDARY FILE WILDCARD SUBROUTINE CALL FORMAT	18
4.1	GENERAL PHILOSOPHY	18
4.2	DETAILED LAYOUT	19
4.3	DEFAULTS	20
4.4	SECONDARY ENTRY POINTS	20
5.0	REGISTER, CHANNEL, AND CORE USAGE	20
5.1	REGISTER USAGE	20
5.2	CHANNEL USAGE	21
5.3	CORE USAGE	21
6.0	CRITICAL ALGORITHMS	21
6.1	DEVICE SELECTION	21
6.2	DIRECTORY SCAN	22

6.3	ARGUMENT FETCH AND STORE	22
6.4	CHANNEL USAGE	23
6.5	DIRECTORY MATCHING LOGIC	23
6.6	SECONDARY WILD-CARD LOGIC	23
7.0	DEFENSIVE HALTS	24
8.0	ERROR MESSAGES	24
9.0	CODING CONVENTIONS	25
10.0	SUMMARY OF ROUTINES CALLED	26

## APPENDICES

A.	SAMPLE PROGRAM	A-1
B.	ASSEMBLY INSTRUCTIONS	B-1
C.	SUGGESTIONS FOR THE FUTURE	C-1

## 1.0 INTRODUCTION

WILD is designed to be used as a subroutine called from any program which needs to perform wild-card LOOKUPS on any device. It is needed since the logic for selecting and reading directories has become quite involved under the 5-series monitor. Among the special cases handled correctly by WILD and no other user-mode program, are sub-file directories, NEW:, HLP:, SYSA:, and controller, controller-class, and file structure abbreviations. Most Digital-supplied utilities will eventually be converted to use WILD if they perform wild-card operations.

The principal purpose of this document is to give sufficient information to allow the reader to call upon WILD as a module without having to resort to the listing.

## 1.1 FUNCTION

As the 5-series monitor's disk structures have become more and more complicated, it has become obvious that all the directory handling code ought to be centralized in one or two places. Since the code to handle wild-card arguments is non-trivial, it was decided to keep it out of the monitor. Thus, the code to handle directories for wild-card LOOKUPS has been concentrated in one module, WILD. This module was designed to be loaded with any program and to perform the general wild-card LOOKUP logic for it. Since the exact handling of the LOOKUP is very program dependent, WILD performs all of the wild-card LOOKUP except the OPEN and LOOKUP itself.

The user presents WILD with a block of arguments which point to the arguments that have typically been scanned by SCAN. WILD analyzes this data base, and selects files from the file system one at a time. It returns the results of the selection in an OPEN block and an extended LOOKUP block and returns control to the calling program. Upon reentry, WILD continues its selection process until it has completed the request. At this point, it returns to the caller with a code indicating that no more files exist. The caller would then typically call SCAN once again to obtain the next command.

As a convenience, WILD contains several routines which can be used by the calling program to handle various conditions. These include LOOKUP and OPEN errors, checking the /BEFORE and /SINCE switches, and reading the data file one byte at a time. Several of WILD's locations are declared INTERNAL to allow the caller control. These include the buffer header, the range of a concatenated set of file specifications, and a flag indicating how much wild-carding is currently invoked.

There are three principal problems which WILD solves. The first is the logic for interpreting the device names and selecting the correct structures. This involves recognizing the various abbreviations and handling each case. WILD handles structure abbreviations, controller-class abbreviations, and controller abbreviations. In each case, the correct subset of system structures is selected. For devices such as "D:" or "DSK:", WILD uses the job's search-list.

The second problem which WILD handles is the various ersatz devices, such as "SYS:", "SYSA:", and "HLP:". For each special device, WILD obtains from the monitor which search-list to use (job, sys, or all) and what directory simulates the device (e.g., [1,4] for SYS:, [2,5] for HLP:).

The third problem solved by WILD is the logic for scanning directories. In particular, WILD can scan up and down SFDs and will do so as instructed by the user via the argument blocks from SCAN. For DECTapes, WILD scans the DECTape directory and returns similar results to the caller (including a flag for the caller to perform a short, rather than an extended LOOKUP). For all other devices, WILD performs no wild-card logic, but does format the OPEN and LOOKUP blocks appropriately. In the case of magtapes, WILD copies the /DENSITY and /PARITY switches into the OPEN block.

WILD also contains a routine that aids the caller in handling wild-cards on an ENTER, (i.e., on the output side of a command specification). This routine (.SCWLD) also supports wild-cards on a secondary input file (e.g., the comparison file in FILCOM or CAM). Once presented with the current input file name, the file specification matching the current input file name, and the output (or secondary input) file specification, .SCWLD determines the OPEN and ENTER (or LOOKUP) blocks. .SCWLD can be called any number of times for each call to the wildcard lookup routine.

## 1.2 GLOSSARY

- All Search List** The (not well-ordered) list of all n structures currently known to the system and physically mounted. This list is exactly the output from the SYSSTR UUU. Its order is that encountered at once-only time as modified by MOUNT and DISMOUNT/REMOVE commands.
- /BEFORE Switch** Switch to SCAN which specifies that the user wishes to consider only those files whose creation dates are before a specified date and time. WILD supports the DATE75 project in checking the /BEFORE switch.
- Caller** The program which called WILD. The person who invoked the caller is referred to as the user. Typically, the user types commands to SCAN which stores them in core. The caller then calls WILD to study this block and select files. Thus the user has specified the request but the caller actually invoked WILD.
- CHKACC UUU** A monitor call which takes a file protection, a desired operation, the owner's number, and the requester's number and validates whether that operation is legal. This is useful if the monitor's checking is suppressed by the JACCT bit (full file access).

**Concatenated File Specification**

A series of file specifications which are treated together as one specification. This allows an "or" effect with the wild-cards. The user currently indicates this to SCAN by separating the specifications by "+" instead of ",".

**Controller Class Name**

All file structures residing on all controllers of a given type. In particular "DP" refers to all disk-pack file structures, and "FH" refers to all RC10 drum file structures.

**Controller Name**

All file structures residing on a specific controller. In particular, the names "DPA:", or "FHB:".

**Date Format**

See internal date-time format.

**DATE75 project**

A current project which entails converting the official DECsystem-10 calendar from a 12-bit date format to a 15-bit date format.

**Default Directory**

The directory in which the monitor performs LOOKUPS if the directory parameter is "0". This is typically the user's logged-in UFD.

**Directory**

An ordered list of files. These are represented by the name, extension, and location of their attributes. Directories are the MFD, UFD's and SFD's.

**Directory Specification**

The way that the user specifies the directory to SCAN. It is always typed within "[" and "]". Fields are separated by comma. The first two fields are the project and programmer numbers and are octal. They specify which UFD. Additional fields are SFDs in order from the UFD down. The following notations are allowed:

[pj,pg]	UFD
[-]	default
	directory
[pj,pg,sfd...sfd]	full path to directory

If pj or pg is omitted, it defaults to the corresponding logged in value. The only directories scanned for files are those at the end of the directory specification and those whose next level of sfd is "\*" (i.e., matches null). Thus, to scan all levels of a user, type "[pj,pg,\*,\*,\*,\*,\*]".

**ENTER**

A monitor call which opens a file for writing on a device. A wild-card ENTER opens a file whose name is composed partially as a functions of the user's command and partially as a function of the file selected by the last wild-card LOOKUP.

**/ERPROTECTION Switch**

A switch to SCAN which indicates that protection check errors messages should be produced. When the UFD is wild, the use of this switch will cause output of all protection check error messages. Otherwise, since the /OKPROTECTION switch is the default, these error messages will be suppressed.

**Ersatz Devices**

Device names such as "SYS:" and "HLP:" are known to the system. They are simulated by appearing as a special directory with a predefined search list. For example, "SYS:" is directory [1,4] and the SYS Search List.

**Extended LOOKUP**

A LOOKUP with at least six arguments. This can be used only for disks.

**File Specification**

What the user types to SCAN. This includes the file's device, name, extension, directory, and switches.

**File Specification Area**

The area of core in which SCAN stores the result of scanning the user's file specification, This instructs WILD as to the files to select.

**File Structure**

The smallest removable unit in the file system. This is one or more volumes, such as disk packs. A file structure is a self-contained entity with its own MFD and set of files.

**File Structure Abbreviation**

An abbreviation of one or more file structures. This refers to all those structures in the all search list whose names match the abbreviation. For example, if there were structures "PRIV:" and "PACK:", "P:" would refer to both structures but "PR:" would mean just "PRIV:".

**Full Path**

The sequence of directories from the UFD to the directory containing the file in question.

**Generic Device DSK:**

The names "D:", "DS:", and "DSK:" are predefined to be a set of structures following the job search list.

**Internal Date-Time Format**

The format for storing a combined date and time internally. This format is used by SCAN, QMANGER, and other programs. It has the property that it is one 35-bit ("integer") quantity such that the difference between two points in time in internal format is constant if they are a constant time apart. The format is: in the left half word, the number of days since November 17, 1858(1); in the right half a fraction of the day since midnight. This results in a resolution of approximately one third of a second. The date field will not be exceeded before 2217 A.D.

**JACCT Program** A program running under the JACCT privilege bit. This is set by the monitor for special system programs such as LOGIN. This bit gives the caller full file access; i.e., it allows the caller to LOOKUP and read any file in the system regardless of the file's protection code.

**Job Search List** An ordered list of structures for this job. This is setup at LOGIN time and may be modified by the MOUNT, DISMOUNT, and SETSRC commands.

**Logged-in UFD** The UFD which corresponds to the project-programmer number under which the user is logged in.

**LOOKUP** A monitor call which selects a specific file on a device. A wild-card LOOKUP is a series of monitor calls which select a set of files which satisfy some conditions such as naming pattern, directory, or creation date. A secondary LOOKUP opens a file whose name is composed partially as a function of the user's command and partially as a function of the file selected by the last wild-card LOOKUP.

**MFD (Master-file Directory)**  
The directory containing all the UFDs.

**/OKNONE** A switch to SCAN which indicates to WILD that it is acceptable if no files match this file specification. Unless this switch is specified, WILD detects this case and issues a warning message.

**/OKPROTECTION Switch**  
Switch to SCAN which suppresses all protection check error messages. This switch is the default when the UFD is wild.

**ONCE-Only Time** That time during system initialization prior to scheduling any jobs.

**OPEN** A monitor call which associates an I/O channel with a specific device or class of devices. In the case of disk-like devices, the association with a specific structure or unit is done at LOOKUP time rather than OPEN time.

-----

1 This is the origin date used by the so-called Smithsonian calendar. This calendar is in use by several computer systems and many astrophysics programs. Its origin was selected because November 18,1858 was the date of the first "Harvard Plates", which were the first accurate astronomical photographs. Hence, this date standard minimizes the date field while leaving all astrophysical measurements as positive dates.

**/PHYSICAL** A switch to SCAN which instructs WILD to use physical UOs for all operations.

**Physical Unit Name** The name of a specific unit of rotating memory. This refers to the structure which is mounted on that unit. For example, if "PRIV:" is mounted on "DPA0:" and "DPB3:", then "DPA0:" would refer to "PRIV:".

**Physical UOs** It is sometimes necessary to ignore any logical names assigned by the user. In order to accomplish this, the OPEN and CALLI UOs can optionally be specified with the request "physical only". This is done in response to the /PHYSICAL switch. It is also used when the logical-to-physical correspondence has already been made, for example, when the device name was returned from the SYSSTR UO.

**SCAN** A module which accepts a user's command line and stores it in core in binary.

**Search List** One of the All Search List, Job Search List, or SYS Search List.

**Secondary LOOKUP** See LOOKUP

**SFD (Sub-file Directory)** A directory within another directory. Each user has a UPD. Within that, he may have as many SFDs as he wishes. He may repeat this to any level up to an installation maximum. The installation maximum cannot be greater than 5 SFDs deep.

**Short LOOKUP** A LOOKUP with four arguments. This can be used on any device.

**/SINCE** A switch to SCAN which indicates that the user wishes to consider only those files created after the specified date and time. WILD supports the DATE75 project when checking the /SINCE switch.

**/STRS** A switch to SCAN which indicates that the user wants to consider all copies of a file of a specified name. This happens automatically if any wild-cards are specified. If wild-cards are not specified, the normal behavior is to LOOKUP only the first file of this name found in the specified search list. With this switch, even if the same name is used on several structures, each will be found.

**Structure** A File Structure.

**Super-USETI** A style of reading a disk unit or file structure which reads by giving absolute addresses rather than locations within a file.

Switches	The part of a file specification which is preceded by a slash. SCAN recognizes the following standard switches for WILD: BEFORE, OKNONE, OKPROTECTION, PHYSICAL, SINCE, and STRS.
SYS Search List	The ordered list of structures on which device "SYS:" is simulated. This is also used for several of the ersatz devices because it is a constant, well-ordered list.
Target Directory	A disk directory which is included by a wild-card.
Time Format	See internal date-time format.
Time Switches	The switches /BEFORE and /SINCE.
UPD (User-file Directory)	The top-level directory for each user. Also, the top-level directory for those ersatz devices which appear as one directory.
Unit	See Physical Unit Name.
User	The person who invokes the program which calls WILD. See also Caller.
Volume	The medium which is mounted on a unit. Examples are a reel of magnetic tape and a disk pack.
Wildcard Construction	The method by which a filename, extension, or directory name is replaced totally with an asterisk or partially with a question mark to designate a cross-section of certain filenames, extensions, or directories.
Wildcard ENTER	See ENTER
Wildcard LOOKUP	See LOOKUP.

### 1.3 REFERENCES

DECsystem-10 Monitor Calls Manual, especially Chapter 6 (disk).

SCAN detailed specifications (under development).

DECsystem-10 Operating System Commands Manual, especially descriptions of DIRECT and QUEUE commands.

DOC files distributed with WILD (WLD?.RND), and SCAN (SCN?.RND).

## 2.0 FILE SPECIFICATION AREA

### 2.1 GENERAL PHILOSOPHY

The file specification area is a block of storage 24(octal) words long. This area contains the scanned result of one file specification. The device occupies one word, but most of the remaining parts of the file specification occupy two words in order to save the wild-card information in a convenient format. The two-word pairs typically contain the name to match against in the first word, and the test mask in the second. Thus, if the user types "FOOBAR", the first word contains SIXBIT /FOOBAR/ while the second contains -1, meaning match on all bits. If the user types "FO?BAR", the first word contains SIXBIT /FO?BAR/ (the "?" is ignored by WILD, but is convenient for messages) and the second contains 777700,,777777 to indicate that anything can match the third character. If the user types "\*", the first word contains SIXBIT /\*/ and the second contains 0.

Care should be exercised in setting up and analyzing the specification area. The conventions are quite rigid (and not necessarily obvious) to indicate when a field is unspecified. This is defined in the detailed layout. Two (optional) words of the specification contain the /BEFORE and /SINCE switch arguments. These are represented in the internal format: LH=days since Nov. 17, 1858 and RH=fractions of a day. Thus, the 36-bit word is continuous with constant increment with time (in units of approx. 1/3 second).

The remaining switches of interest to WILD are stored in a word known as the "mod" word. This has a matching word which contains the "mod" mask. This mask is a 1 if the corresponding "mod" switch has been specified, and the "mod" word contains the value.

### 2.2 DETAILED LAYOUT

Name (1)	Word	Definition
.FXDEV	0	Device name; 0 if ":" missing
.FXNAM	1	file name for match; name missing if 0
.FXNM	2	mask for file name match; missing if .FXNAM=0
.FXEXT	3	LH: extension for match RH: mask for extension match. "." missing if word 0

-----  
 1 These names are defined in the file SCNMAC.MAC. Also defined are the two symbols .FXLND=6 (length of full path) and .FXLEN=24 (length of standard area).

.FXMOD	4	standard file switch values
.FXMOM	5	mask for which switches were typed
.FXDIR	6	name of UFD for match; directory present if FX.DIR=1; LH and/or RH = 0 to default to logged in project and/or programmer
.FXDIM	7	mask for UFD match
	10,12..20	name for SFDs to match; 0 if end of list
	11,13..21	mask for SFD match
.FXBFR	22	/BEFORE in internal date/time format; 0 if missing
.FXSNC	23	/SINCE in internal date/time format; 0 if missing

## Bits within mod word:

FX.NDV	1B0	no device specified by user
FX.NUL	1B1	null extension (". " but no name)
FX.DIR	1B2	directory specified ("[" and "]" but not [-])
FX.PHY	1B3	/PHYSICAL
FX.NOM	1B4	/OKNONE
FX.DFX	1B5	(internal) directory has been "fixed-up"
FX.TRM	7B8	concatenated file specification; appears on all but last of a set
.FXTRA	1	And
.FXTRO	2	Or
.FXTRN	3	Not
.FXTRC	4	Concatenate
FX.STR	1B9	/STRS
FX.PRT	1B10	/OKPROTECTION
FX.PAR	1B24	/PARITY: EVEN
FX.DEN	3B26	/DENSITY (1=200, 2=556, 3=800)
FX.PRT	777B35	/PROTECTION

## 2.3 DEFAULTS

SCAN always defaults the device to "DSK:" if anything was typed in the specification. The caller must supply its default if 0.

The name and name mask must be defaulted by the caller if the name word is 0. Typically it is a user error if no name is given, but in some cases, "\*" is an appropriate default. If a previous file specification had a name, then it usually carries over as the default.

If the user omits the extension (no "."), the caller should supply a default. If this file is not found, and there is no wild-carding, then the same file should be tried again with the null extension. If the user types "." with no extension, he has indicated the null extension explicitly. This is flagged by the FX.NUL bit.

- Usually no defaults are necessary for the mod word or its mask.

Usually no defaults are necessary for the time switches.

If a directory is specified (FX.DIR=1), then the LH of the first directory word is defaulted to the user's logged-in project, and the RH to his programmer number. WILD does this if the directory is 0 in that half.

Note that there is special handling of the directory in the following cases:

1. If no wild-cards are in effect, and if no directory is specified, then the resulting directory will be 0 so that the monitor will supply the defaults.
2. If the device is a ersatz device which overrides the directory specified (e.g., "SYS:"), then the directory is as specified by the monitor.
3. If the extension is ".UPD" with no wild-cards, then the first directory word is taken as the file name and the MFD is used as the directory.

### 3.0 WILD-CARD LOOKUP SUBROUTINE CALL FORMAT

#### 3.1 GENERAL PHILOSOPHY

The principal entry point is named ".LKWLD".

The calling sequence has been designed to allow graceful future expansion in a compatible fashion. The call is with a length and location pointer in AC1. This points to a block of storage. When fetching arguments, WILD assumes that arguments beyond the length are 0, and most arguments take on default behavior when they are 0. When WILD finds a file, it gives a skip return with the caller's OPEN and extended LOOKUP blocks setup. AC1 is returned with the device characteristics (DEVCHR). AC2 is set to -1 if an extended LOOKUP is needed (disk), 0 if a short LOOKUP is needed (DECTape), or +1 if no LOOKUP is needed (all other devices including NUL:). For magtapes, the OPEN block includes the /PARITY and /DENSITY settings. For all devices, the file mode is 0 so the caller can set any mode with an IORM instruction. When the end of the user's specifications is reached, WILD takes the non-skip return with AC1 containing -1.

The arguments to WILD include the lengths of the file specification block and the length of the LOOKUP block so that both can grow gracefully with time. WILD requires that when several file specifications are given, they are stored in consecutive locations in core and are all the same length. The arguments point to the first word of the first and last specifications given by the user. One of the arguments is the channel to be used by the byte-read routine. This will not be used if the byte-read routine is never called.

One of the arguments is the location of a specification pointer. This pointer must be zeroed before the first call to WILD and thereafter must be left untouched. It is updated by WILD to indicate which file specification matched the current file. The initial 0 value is used as a flag for WILD to initialize itself.

### 3.2 DETAILED LAYOUT

The argument block is pointed to by ACL. The pointer contains the length in the left half and the location of the argument block in the right half.

- block+0 LH: location of a word whose contents is the address of the first file specification area.
- block+0 RH: location of a word whose contents is the address of the first word of the last file specification area (if 0, or if contents=0, then just one specification).
- block+1 LH: location of the 3-word OPEN block. Upon return the LH of the first word will have the physical bit set correctly and the RH setup with mode=0 (if the device is a Magtape, the RH will contain the /PARITY and /DENSITY as specified), the second word will have the device name, and the third word will have LH=0 and RH pointing to the buffer header in WILD.
- block+1 RH: location of the extended LOOKUP block. Upon return, WILD will have zeroed the area and set in the length, directory, name, and extension. It will also have stored the directory in the name word+3 for the convenience of programs wishing to do short LOOKUPS.
- block+2 LH: length of one file specification area.
- block+2 RH: length of the extended LOOKUP block.
- block+3 LH: bit 0 -- flag to enable all devices if 1. If 0, only disk devices are handled (others return with DEVCHR in ACL with a non-skip return).  
bits 14-17 -- channel to be used for data file reading if any.

block+3 RH: location of a pointer word. This word must be preset to 0 before the first call to WILD for a given command. It must be preserved from then on. WILD will update it to point to the file specification responsible for the current file. The caller can then examine any other pertinent switches.

### 3.3 DEFAULTS

The only default is that if block+0 RH = 0 or if the location pointed to = 0, then one file specification is assumed.

All future growth will attempt to default to the current behavior if additional arguments are not given. Unused (future) arguments should always be filled with 0 now to ensure maximum long term compatibility.

### 3.4 SECONDARY ENTRY POINTS

Most of these routines take no explicit arguments, but rather use the blocks pointed to by the last call to .LKWLD.

E.DFL will type a LOOKUP failure message based upon the error code in the extended LOOKUP block. WILD has mnemonic messages for all legitimate LOOKUP errors. If the error code is unexpected, it will type the code in octal.

E.DFO will type an OPEN failure message for the device which WILD selected. If available, WILD will include the job number which has the device in use.

E.LKEN will type a LOOKUP/ENTER failure message based upon the error code in an arbitrary extended LOOKUP block. It is called with AC1 pointing at the LOOKUP or ENTER block, AC2 containing the length of the block, and AC3 pointing to the SCAN-style file specification of at least 22(8) words. WILD has mnemonic messages for all legitimate LOOKUP and ENTER errors. If the error code is unrecognized, it will type the code in octal.

.CHKTM checks the /BEFORE and /SINCE switch constraints by using the 15-bit creation date and time in the extended LOOKUP block and the switch values in the current file specification area. If the constraints are satisfied, the skip return is taken; otherwise, the non-skip return.

.CHKTA (for use of DIRECT only) same as .CHKTM except that the compare is always performed. .CHKTM accepts all non-disk files without comparing /BEFORE, /SINCE, etc.

.LKERR (called from E.DFL automatically) will type a mnemonic lookup error message without the file name, etc. The error code must be in AC1 and the protection (or 0) in bits 0-8 of AC3.

.NXDTW skip returns with the next byte in AC1, or non-skip returns at end-of-file. It automatically issues warnings for status errors and clears the condition. The byte size is dependent upon the mode of the OPEN. This routine will allocate space for the buffers automatically on the first call unless the caller has already done this. If done automatically, space will be allocated at .JBFF.

.TFILB is the same as .TFILE except the scan block typed is an arbitrary one. It is called with AC1 pointing to this block, which is assumed to be at least 22(8) words long.

.TFILE types a UFD or filename. The output starts with a space and results in the appropriate line followed by a carriage return, line feed. The messages are:

dev:[pj,pg].UFD,

dev:DIRECTORY,

FILE dev:name.ext[directory], or

DIRECTORY dev:name.SFD[directory]

Only AC1 is altered.

### 3.5 INTERNAL FLAGS

B.DC is the 3-word buffer ring header for the read-byte routine.

.WIFIR and .WILAS point to the first and last of a set of concatenated specifications.

.WILDL is the length of WILD's low core (impure) area.

.WILDZ is the start of WILD's low core (impure) area.

.WLDBF is a 200(8) word block which contains the current directory block.

.WLDFL is 0 if no wild-card on this specification, is +1 if the structure is wild (/STRS), is -1 if the file is wild, and is -2 if both structure and file are wild.

### 3.6 STRUCTURE LIST SUBROUTINES

These routines provide a composite, easily useable extension of the JOBSTR UO. Presented the user device, a list of devices matching this one is returned.

.INSTR is called to initialize the list. The caller supplies the device in AC1 and AC2 containing LB0=1 if /PHYSICAL. .INSTR will not skip if the device is not a disk. If the argument is a disk, it will

skip return with AC1 containing 0 or 1B0=1 if need to scan multiple structures. AC1 should be IORMed into the OPEN block to set the physical-only bit for this case.

.NXSTR has no arguments and returns successive entries in the list of structures. It never skips and returns the structure name in AC1. When the list is exhausted, AC1 is set to 0.

#### 4.0 SECONDARY FILE WILDCARD SUBROUTINE CALL FORMAT

##### 4.1 GENERAL PHILOSOPHY

The principal entry point is named ".SCWLD".

The calling sequence has been designed to allow graceful future expansion in a compatible fashion. The call is with a length and location pointer in AC1. This points to a block of storage. When fetching arguments, WILD assumes that arguments beyond the length are 0, and most arguments take on default behavior when they are 0. If the request cannot be done, the non-skip return is taken with the message already issued. If the call is successful, the skip return is taken with AC1 containing the device characteristics (DEVCHR) and AC2 containing -1 if disk or spooled device, 0 if DECTape, or +1 otherwise (including NUL:). For magtapes, the OPEN block includes the /PARITY and /DENSITY settings; for disks the OPEN block includes the /PROTECTION. For all devices, the file mode is 0 so the caller can set any mode with an IORM instruction, and the /PHYSICAL setting is correct.

The arguments to WILD include the length of the resulting LOOKUP/ENTER block to allow graceful growth with time. The caller also must supply a location containing the default extension.

.SCWLD is called with a set of arguments for each of two files. One set is for the controlling or primary input file. This is typically the result of having called .LKWLD and then performing the requisite OPEN and LOOKUP. The other set is for the secondary file. The user's command is an input argument in the form of a .SCAN filespec and .SCWLD sets as output an OPEN block and a LOOKUP/ENTER block. These can be used by the caller to initialize the secondary file.

Examples of possible use include:

1. A utility such as PIP:
  - \* OUT = IN

IN is the primary spec and OUT is the secondary spec.  
The caller:

- calls .TSCAN
- calls .LKWLD for IN
- OPENS and LOOKUPS IN
- calls .SCWLD for OUT
- OPENS and ENTERS OUT

## 2. A compiler such as FORTRAN:

```
* rel,list = in
```

IN is the primary spec and REL and LIST are both secondary specs. The caller:

```
- calls .TSCAN
- calls .LKWLD for IN
- OPENS and LOOKUPS IN
- calls .SCWLD for REL
- OPENS and ENTERS REL
- calls .SCWLD for LIST
- OPENS and ENTERS LIST
```

## 3. A special utility such as FILCOM:

```
* OUT = IN1,IN2
```

IN1 is the primary spec and IN2 and OUT are both secondary specs. The caller:

```
- calls .TSCAN
- CALLS .LKWLD for IN1
- OPENS and LOOKUPS IN1
- calls .SCWLD for I2
- OPENS and LOOKUPS I2
- CALLS .SCWLD for OUT
- OPENS and ENTERS OUT
```

## 4.2 DETAILED LAYOUT

The argument block is pointed to by AC1. The pointer contains the length in the left half and the location of the argument block in the right half.

block+0 LH: address of primary file specification. The specification is assumed to be at least 22(8) words long.

block+0 RH: address of secondary file specification. The specification is assumed to be at least 22(8) words long.

block+1 LH: address of OPEN block for primary file. This is assumed to be at least 3 words long.

block+1 RH: address of open block for secondary file. Upon return LH of the first word will have the physical bit set correctly and the RH containing parity and density set and mode=0. The second word will have the device name.

block+2 LH: address of extended LOOKUP block for the primary file. The LOOKUP must already have been done using this block. It is in extended format and must be at least six words long.

block+2 RH: address of extended LOOKUP/ENTER block for the secondary file. Upon return, the block will be zeroed and then the name, extension, directory, and protection will be set.

block+3 LH: address of a word containing in the LH the default extension and in the RH the constant 77777(8).

block+3 RH: length of the secondary file LOOKUP/ENTER block.

#### 4.3 DEFAULTS

The default secondary name is the same as the primary file name. If the secondary file has the same owner as the primary file, then the default protection is XYY where X is from the standard monitor file protection and YY is from the primary file.

Typically, the caller would apply additional defaults after the call to .SCWLD. Two common ones are to copy the primary files version (.RVER) and to estimate the secondary size (.RBEST) based on the actual length of the primary file (.RBSIZ).

#### 4.4 SECONDARY ENTRY POINTS

These routines take no explicit argument, but rather use the blocks pointed to by the last call to .SCWLD.

E.SCL will type a LOOKUP/ENTER failure message based upon the error code in the extended LOOKUP/ENTER block. WILD has mnemonic messages for all legitimate LOOKUP and ENTER errors. If the error code is unexpected, it will type the code in octal.

E.SCO will type an OPEN failure message for the device which WILD selected. If available, WILD will include the job number which has the device in use.

### 5.0 REGISTER, CHANNEL, AND CORE USAGE

#### 5.1 REGISTER USAGE

WILD treats ACs 1-4 as temporary registers. It will clobber them on any call. All calls are via PUSHJ on AC17. When WILD needs more than four ACs, it uses ACs 5-7 as preserved ACs. In the interests of future compatibility, none of the arguments should point to ACs 5-10, which are treated as preserved.

## 5.2 CHANNEL USAGE

The only channel used by WILD is channel 0. All directory reading is multiplexed on channel 0. The channel is released after each block is read, so the caller can use channel 0 for its own purposes between calls to WILD.

WILD will use the channel of the caller's choice (even 0) in the read-byte routine.

## 5.3 CORE USAGE

WILD is normally loaded into the high segment (2700(8) words pure) and has a fixed low-segment data base of about 400(8) words which is initially zero. There is also a dummy module, WILDDM, which defines a low segment the same size as the fixed low-segment data base. Within the low-segment is the buffer used for reading the directories. The only dynamic core allocation made by WILD is if the read-byte routine allocates buffers at .JBFF when called after the OPEN. This can be avoided by issuing an INBUF UVO after the OPEN before calling the read-byte routine.

## 6.0 CRITICAL ALGORITHMS

### 6.1 DEVICE SELECTION

The first step is to get the DEVCHR of the specified device. If it is not a disk-type device, then return the DEVCHR to the caller. If the device is a disk, look to see if there are any wild-cards specified. If not, then no directory scanning is needed. In this case, check to see if /STRS has been specified. If so, we must identify the correct device. If not, then we can use the user's device and give the specification to the caller without any further work.

If some wild-cards are indicated, it becomes necessary to perform a wild-card search, which requires that we consider each structure in turn. First, we see if this is a 5-series monitor by checking ST#TDS in the system STATES word (%CNSTS). If this fails, or if this is pre-5-series monitor, then no file structure logic is necessary and only the special case of "SYS:" need be considered.

If the PATH. UVO is implemented, then it will indicate if an ersatz device is specified. If so, we may need to disregard the user's directory and use the monitor's. This will also indicate if a special search list is needed. All monitors which support ersatz devices other than "SYS:" include the PATH. UVO, so this test will always succeed for ersatz devices such as "NEW:", "BAS:", etc.

Otherwise, we make use of the undocumented pre-5.05 behavior of the DSKCHR UVO for ersatz devices. It turns out that in early 5-series monitors, DSKCHR will fail if a ersatz device is indicated. This is a sure sign that the device specified is "SYS:" since ersatz devices

other than SYS: require the PATH. UUO. If DSKCHR succeeds, then it indicates the class of the device: generic disk, structure abbreviation, controller class, controller, or normal. In each case, the appropriate flags are set to select devices from the correct search list. For structure abbreviation, the all search list is used and the names checked for match with the argument. For Controller Class and Controller, the appropriate flags from DSKCHR (DC.CNT and DC.CNN) are used to match the entries in the all search list. For generic disk, the job's search list is used. For "SYS:", the SYS Search List is used in 5.03 and more recent monitors and the All Search List used in 5.01 and 5.02 monitors.

The main (outer-most) loop in WILD is the structure selection loop. It is initialized to 0, and when the appropriate search list returns 0 or -1, the loop is terminated. The procedure is to take the last structure considered and use the JOBSTR, SYSSTR, or GOBSTR UUO for the job, ALL, or SYS search list, respectively. If structure abbreviation is in effect, then the result is checked against as much of the device argument as the user typed. If Controller Class or Controller is indicated, then the result of the search list is checked with DSKCHR for being in the correct set.

## 6.2 DIRECTORY SCAN

Once the structure is known, WILD proceeds to scan the directory hierarchy in its entirety. The general technique is to start with the MFD. When a file is found with extension ".UFD", the current state of reading the MFD is stored (block and word within block) and the scan drops down one level. Except within the MFD, whenever a file is found with extension ".SPD", the current state is pushed, and that directory is read. In the case of an SPD, the SPD is first considered as a candidate for selection, and then is scanned as a directory. This is not necessary for UFDs since they are all in the [1.1].UFD and will be considered from there.

Whenever an end-of-file is encountered while scanning a directory, the stack is popped one level, and the previous directory scan is resumed. This technique causes all files to be considered in a top-to-bottom, left-to-right fashion.

Each file considered is tested against the arguments to see if it matches in name, extension, and directory. Since most requests are not wild in all their parameters, WILD looks ahead to see if each directory can be skipped. If so, then it is not necessary to read that directory. WILD also notices if certain names, extensions, and directories are constant and fills them in without reference to the disk.

## 6.3 ARGUMENTS FETCH AND STORE

When fetching arguments from the caller, WILD carefully observes the length pointer from the caller. When fetching an argument beyond the caller's list, 0 is automatically used. If this is an error, then WILD will bomb out in a consistent and reproducible manner.

When storing in the caller's extended LOOKUP block, WILD is careful to observe the length specified by the caller. This ensures that no wild stores will be done, even if the caller has given a ridiculously short block.

To determine a consistent concatenated file specification, WILD verifies that all components have the same device and directory and setting of the /PHYSICAL switch. If not, then the request is separated into two possibly concatenated file specifications.

#### 6.4 CHANNEL USAGE

WILD uses only two channels. All directory reading is done one block at a time on channel 0. This allows all channels to be used by the caller. Each block is read by OPENing the channel, LOOKUPing the directory, positioning to the block, and reading the block. The channel is then CLOSED and RELEASEd. The CLOSE is issued with the request to delete access tables at end-of-file, and to keep them before end-of-file. The access date is not updated.

.NXDTW uses any channel specified by the caller, even 0. It assumes that the caller takes care of the OPEN, LOOKUP, CLOSE, and RELEASE. When an error is encountered, the status bits are read and listed out as a warning message. Then the error bits are cleared and reading resumes.

#### 6.5 DIRECTORY MATCHING LOGIC

If the user's project number is greater than 7, and the target directory project is less than 10, then if the user typed "\*" for project and not "\*" for programmer number reject the match. This supports the notion that for projects below 10, the programmer number is independent of those above 10. In particular, with libraries, etc., being developed for projects below 10, users looking for their files in all projects should not see these files. If either the project number is explicit or the programmer number is "\*", then the old behavior is indicated. Thus, [\*,\*] will still search all UFDs. Note that [?,x] will match programmer x in all directories below project 10. Note also that the behavior from [1,2] is unaffected.

#### 6.6 SECONDARY WILD-CARD LOGIC

The algorithm is to place both the secondary file specification (input or output) and the primary input file specification into the canonical format device: name.extension [project, programmer, sfd...]. Then these are scanned from left to right for a wild bit (where ? is 3 or 6 wild bits, \* is 18 or 36). For each matching wild bit, the actual value of the input file is copied to the secondary file. (Non-wild bits in the secondary specification leave the value as specified by the user.) Thus, for example,

Primary file	Primary spec.	Secondary spec.	Secondary file
FOO.EXT	*.EXT	*.OUT	FOO.OUT
FOO.EXD	FOO.EX?	FOO?.LST	FOOD.LST
FOO.EXT	*.EXT	XYZ.LST[5,6,*]	XYZ.LST[5,6,FOO]

### 7.0 DEFENSIVE HALTS

In order to guard against bad arguments from the caller or errors from the monitor, several defensive HALTs have been placed in WILD. None of these HALTS should ever happen; they all represent internal errors in either the caller, WILD, or the monitor. A brief summary follows:

1. If the OPEN or LOOKUP block pointers are zero, a HALT is executed.
2. If a failure return is given by JOBSTR, SYSSTR, GOBSTR, or DSKCHR UOs, a HALT is executed.
3. If the LKERR routine determines that error code 0 means no-such directory, a HALT is executed (Internal). This will not happen in practice since the logic will already filter out this case and issue the appropriate message.
4. If the INSTR routine decides that the argument from .LKWLD is not disk, .LKWLD will HALT. This cannot happen in practice because .LKWLD has already filtered out non-disk devices.

### 8.0 ERROR MESSAGES

%WLDDEM DIRECTORY ddd:[xxx] IS EMPTY

?WLDLFO } OPEN FAILURE FOR ddd: IN USE BY JOB jjj  
?WLDSCO }

%WLDIOE ERROR ssssss WHILE READING ddd:nnn.eee[xxx]

?WLDIWC INSUFFICIENT WILD-CARDS IN INPUT FOR OUTPUT dev:file.ext[xxx]

The user specified more wild cards in his secondary or output specification than he did in his primary input specification.

?WLDLKE LOOKUP FAILURE nn ddd:nnn.eee[xxx]

%WLDLKP	NON-EXISTENT NON-EXISTENT UFD FILE BEING MODIFIED RIB OR DIRECTORY READ ERROR NO ROOM WRITE LOCKED CAN'T SUPERSEDE DIRECTORY NON-EXISTENT SFD SEARCH LIST EMPTY SFD TOO DEEP NO CREATE PROTECTION ppp FAILURE	} ddd:nnn.eee[xxx]
---------	--	--------------------

?WLDNDO NULL NAME IN OUTPUT WILD-CARD ddd:nnn.eee[xxx]

?WLDNFS NO FILE NAME specified ddd:nnn.eee[xxx]

The input file specification has a blank file name.

?WLDNNO NULL DIRECTORY IN OUTPUT WILD-CARD ddd:nnn.eee[xxx]

The resulting secondary or output file has a blank name (directory).

%WLDNSD NO SUCH DIRECTORY ddd:[xxx]

%WLDNSF NO SUCH FILES AS ddd:nnn.eee[xxx]

%WLDNXD NONEXISTENT ddd:[xxx]	{ .UFD .SFD
-------------------------------	----------------

%WLDNXU NONEXISTENT ddd:[xxx] .UFD

%WLDLSE SEARCH LIST EMPTY FOR DEVICE ddd:

## 9.0 CODING CONVENTIONS

Coding and commenting conventions of the 5-series monitor apply throughout. Subroutines will be grouped by function. All subroutine calls will be "down the page". All calls will be via PUSHJ 17, or PJRST (i.e., JRST instead of PUSHJ/POPJ). All routines will return (POPJ) or skip return (CPOPJ1). For routines which have a skip return, the non-skip return will indicate the abnormal or error condition. The non-skip return point of such calls and monitor UUs will be indented two spaces. Each subroutine will be preceded by a commentary in a standard format to define its function, its calling sequence, and any abnormal AC usage.

A local symbol is assigned to each error message. The local symbol has the form E\$\$ xxx, where xxx is the last three letters of the error message prefix. The local symbol for the first error message in Paragraph 8.0 is then E\$\$DEM.

## 10.0 SUMMARY OF ROUTINES CALLED

- All but one of the routines described here are defined in SCAN. If that module is loaded, then the only routine which the caller must define is .TNEWL. For most cases, this can be defined as a POPJ 17,.

.CNVDT This routine converts an arbitrary date to the internal format. It is called with AC1=time in millisec and AC2=date in system format. It returns AC1=internal format date,time.

.GTWRD This routine gets the next word from the caller's parameter area. It is called with AC1=address of table, AC2=counter of length to go. It returns with AC1 incremented, AC2 decremented, AC3 containing the contents or 0. AC4 is untouched.

.MKMSK This routine makes a mask corresponding to the non-blanks in a SIXBIT word. It is called with AC3=word. It returns with mask in AC1. Only AC2 is scratched.

.PTWRD Stores one word in a variable length parameter area. It is called with AC1 containing the desired location, AC2 containing the length left in the area, and AC3 containing the data item to store. If the length is positive, the word will be stored. The location will be incremented by one and the length decremented by one.

.SAVE1 This routine preserves AC5 for the caller. It returns with all ACs intact, but with the push-down list two longer.

.SAVE3 This routine preservers AC5-AC7 for the caller. It returns with all ACs intact, but with the pushdown list two longer.

.TCHAR This routine types one ASCII character. It is called with AC1 containing the character. It preserves all ACs.

.TCOLN This routine types a colon (:). It is called with AC1 containing the colon. It preserves all ACs.

.TCRLF This routine types a carriage return and a line feed. It preserves all ACs.

.TDECW This routine types a signed decimal number. It is called with AC1 containing the number. It uses AC1-AC3.

- .TDIRB** This routine types a directory block within "[" and "]". It is called with RH AC1=address of block, LH AC1=2 to indicate SCAN-style biwords. It uses AC1-AC4.
- .TNEWL** This routine positions the typeout to the left margin if necessary. This is needed only if the routine's own typeout is not necessarily at left margin whenever WILD is called. It can use AC1.
- .TOCTW** This routine types a signed octal number. It is called with AC1 containing the number. It uses AC1-AC3.
- .TSIXN** This routine types a SIXBIT word. It is called with AC1 containing the word. It uses AC1-AC2.
- .TSPAC** This routine types a space. It is called with AC1 containing the space. It preserves all ACs.
- .TSTRG** This routine types a string of ASCII text. It is called with AC1 containing the location of the first word of text. It uses AC1.
- .VERBO** This routine uses only AC1 and returns it with bit 35=1 if the six character prefix is to be issued in error messages and with bit 34=1 if the text explanation is to be issued.

**APPENDIX A**  
**SAMPLE PROGRAM**

```
START:      RESET
            call .ISCAN

LOOP:       call .TSCAN
            supply defaults
            pointer <-- 0

GFILE:      call .LKWLD
            if end, go to LOOP
            open/lookup
            call .CHKTM
            if rejected, close/release and go to GFILE

DOIT:       process file using .MDDTW to read it at
            end-of-file, close/release device and go to
            GFILE
```

A detailed example will be supplied later. Refer to RUNOFF V.10 for example usage.

## APPENDIX B

### ASSEMBLY INSTRUCTIONS

| WILD is assembled with C.MAC and SCNMAC.MAC used as UNIVERSALS. C is used to define all the monitor dependent symbols, while SCNMAC defines the file specification area symbols.

## APPENDIX C

### SUGGESTIONS FOR THE FUTURE

Support a switch to control SFD scanning going to LIB: and SYS:. This would override the job's default.

When user specifies "SYS:" and has NEW: enabled, wild-card over both [1,4] and [1,5].

Make the routine which outputs "IN USE BY JOBn" be generally available.

When the user specifies "ALL:", exclude any single-access structures which he does not own.

Extend the error message routines to include all the components of a concatenated specification.

Simulate the CHKACC UO for any monitor which does not support it. This is used to validate that the user has access rights to read a directory in case WILD was called from a JACCT program.

Use super-USETI to simulate the monitor's action of LOOKUPS. This would allow a program to run without assuming that the file structure was mounted or in good condition.

Make low-segment relocatable. This would allow it to float as a function of the caller's core allocation. This would be particularly convenient for multi-segment programs.

Add a switch to specify explicitly which search list to use. This would override the search list implicit in the device. The device would still be used to determine the directory handling. Thus, for example, the specification "SYS:/SEARCH:ALL" would look at [1,4] on all structures currently mounted.

Support iteration loops, and operators for concatenation, and, or, and except. SCAN would have to recognize these and build the data-base correctly. Then WILD needs to extend its recognition logic to handle these constructions.

Supply a standard routine which, given an OPEN and LOOKUP block, checks for it matching this specification or concatenated specification. This would be useful for DIRECT when handling magtapes in FAILSAFE mode.

The notation "[a,b,c,d].SFD" should be treated as "d.SFD[a,b,c]". This is analogous to the handling of UFDs.

The interpretation of "\*" should be more general than "?????". The latter should not match blanks. The former should be allowed following 0-5 characters to indicate any suffix.

WILD

INDEX

Abbreviations, 5, 22  
All Search List, 6, 8, 10, 22

/BEFORE switch, 5, 6, 11 through 13, 16  
Byte-read routine, 15

Caller, 5, 6, 9, 11, 14, 17, 22, 23, 26  
Channel, 9, 20, 21, 23  
CHKACC UWO, 6, C-1  
Coding conventions, 25  
Commenting convention, 25  
Concatenated file specification, 5, 7, 13, 17, 23, C-1  
Controller class name, 7  
Controller name, 7

Date format, 7, 10, 11, 13  
DATE 75, 6, 7, 10  
DECTape, 6, 14  
Default directory, 7  
Device name, 15  
Directory, 5 through 8, 9, 10, 13 through 24  
Directory specification, 7

ENTER, 6, 7  
Ersatz devices, 6, 8, 11, 14, 21  
Extended LOOKUP, 6, 8, 14 through 16, 23

File specification, 8, 11, 15, 23  
also see concatenated file specification  
File specification area, 8, 12, 15 through 17  
File structure, 8, 10, 11  
File structure abbreviations, 5, 8  
Full path, 8

Generic device DSK:, 8

Internal date time-format, 7, 8, 10, 13

JACCT program, 9, C-1  
Job Search List, 9, 10

Logged-in UFD, 9  
LOOKUP, 5, 6, 7, 9 through 11, 15, 16, 23, 24, 25, C-1

Magtapes, 6, 14, 15  
MFD (Master File Directory), 9, 14, 22

Null Extension, 13, 14

/OKNONE, 9, 11, 13  
ONCE-Only Time, 9  
OPEN, 5, 6, 9, 14 through 18, 22, 23, 29

/PHYSICAL, 10, 11, 13  
Physical Unit Name, 10, 11  
Physical UWO's, 10  
Protection check error messages, 8, 9

SCAN, 5 through 10, 11, 13, 26, 27  
Search List, 5, 10, 24, 25, C-1  
SFD (Sub-File Directory), 6, 7, 10, 13, 22, 25  
Short LOOKUP, 10, 14  
/SINCE, 5, 10 through 13, 16  
/STRS, 10, 13, 17  
Structure, 5, 8, 9, 10  
Subroutine call format, 14, 18  
Super-USETI, 10, C-1  
Switches, 6 through 11, 16, C-1  
SYS Search List, 8, 10, 11, 21, 22

Target Directory, 11, 23  
Time format, 11, 13  
Time switches, 11, 14

UFD (User File Directory), 7  
through 11, 14, 22, 23, 25, C-1  
Unit, 11  
User, 11

Wildcard Construction, 11  
Wildcard LOOKUP, 5, 11