

UOS File Operations

Table of contents

UOS File Operations	3
Contents	3
Preface	3
Introduction	3
Control Blocks	3
File Access Block (FAB)	4
Name Block (NAML)	5
Allocation Control XAB (XABALL)	8
Date and Time XAB (XABDAT)	9
File Header XAB (XABFHC)	10
Item List XAB (XABITM)	11
Key Definition XAB (XABKEY)	12
Protection XAB (XABPRO)	13
Revision Date/Time XAB (XABRDT)	14
Summary XAB (XABSUM)	15
Terminal XAB (XABTRM)	16
Record Access Block (RAB)	16

UOS File Operations

UOS File Operations

September 25, 2020

Created with the Personal Edition of HelpNDoc: [Easily create PDF Help documents](#)

Contents

Created with the Personal Edition of HelpNDoc: [Free PDF documentation generator](#)

Preface

Preface

Intended Audience

This manual is intended for application developers writing software to access files on UOS.

Created with the Personal Edition of HelpNDoc: [Free HTML Help documentation generator](#)

Introduction

Introduction

File I/O on UOS is done through system services, such as OPEN, CLOSE, READ, ERASE, etc. A OPEN or CREATE operation creates an 64-bit integer context called a "handle". This handle is then used as a channel of access to the file or device. Each handle has its own position context, access mode, lock context, and error state - even if there are multiple handles associated with the same physical file.

On file systems that support multiple streams per file, such as the UOS native file system, a handle is associated with a specific file stream. For files on file systems without this support, the stream specification is ignored.

Created with the Personal Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

Control Blocks

Control Blocks

Various structures, collectively called "control blocks", are used for file operations. These blocks are described in detail in the following sections.

There are two types of blocks: Record Access Blocks (RABs) for data I/O, and File Access Blocks (FABs) for other operations.

File Access Blocks contain links to other blocks, such as the NAML (name) block, and extended attribute blocks (XAB, pronounced "zab"). There are several types of XABs, each for a specific set of data. XAB blocks can be chained together and each type of XAB is described in the following sections.

Each control block has an ID that indicates what kind of block it is, and a length that indicates the size of

the block. These values must be set properly or the system services that use them will ignore them or return errors.

Created with the Personal Edition of HelpNDoc: [Free EPub producer](#)

File Access Block (FAB)

File Access Block (FAB)

The FAB defines the basic information about a file and contains links to other control blocks. The contents of the FAB, in order in memory are:

Field	Size, in bytes	Description
FAB_B_BID	1	Block identifier. Must be FAB_C_BID
FAB_B_BLN	1	FAB block byte length. Must be FAB_C_BLN.
FAB_B_ACMODES	1	File access modes
FAB_L_ALQ	4	RMS Allocation quantity (blocks), rounded up to clustersize
FAB_B_BKS	1	RMS Bucket size
FAB_W_BLS	4	Magnetic tape block size
FAB_L_CTX	4	Context
FAB_W_DEQ	2	Default RMS file extension quantity in blocks
FAB_L_DEV	4	Device characteristics
FAB_L_DNA	8	Default file specification string address
FAB_B_DNS	2	Default file specification string size
FAB_B_FAC	1	File access control (ignored on open)
FAB_L_FNA	8	File specification string address. If -1, the associated name block (NAML) NAML_L_LONG_FILENAME field points to the file specification string.
FAB_B_FNS	2	File specification string size. If FAB_L_FNA is -1, the associated name block (NAML) NAML_L_LONG_FILENAME_SIZE field contains the file specification string length.
FAB_L_FOP	4	File-processing options: One of the following options must be chosen. Flag Description FAB_V_GET 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_B_FSZ	1	Number of bytes in the RMS fixed-length control field of a VFC record.
FAB_W_GBC	2	Global buffer count
FAB_W_IFI	8	Internal file ID (file-system dependent)
FAB_B_JOURNAL	1	Journal flags status
FAB_L_MRN	4	Maximum record number for relative files

FAB_W_MRS	2	Maximum record size: Defines the maximum record size for all records in the file. Maximum record size refers to the size of all records in a file with fixed-length records, the size of the largest record with variable-length records, or the size of the variable-length portion of VFC records. A value of 0 with variable-length records means that there is no limit on the record size, except for magnetic tape files, for which a value of 0 sets an effective maximum record size equal to the block size minus 4.
FAB_L_NAM	8	Name block (NAML) address
FAB_B_ORG	1	File organization (0 = binary, other = RMS format)
FAB_B_RAT	1	RMS Record attributes
FAB_B_RFM	1	RMS Record format
FAB_B_RTV	1	Retrieval window size
FAB_L_SDC	4	Secondary device characteristics
FAB_B_SHR	1	File sharing
FAB_L_STS	4	Completion status code
FAB_L_STV	4	Status values
FAB_L_XAB	8	Extended attribute block address
FAB_Q_HANDLE	8	Handle for opened file

FAB_B_BID Field

The block identifier (BID) field identifies a control block as a FAB. Once set, this field should not be altered until the FAB is no longer needed. This field must be set to the symbolic value FAB_C_BID.

FAB_B_BLN Field

The block length (BLN) field defines the length of the FAB. Once set, this field should not be altered until the FAB is no longer needed. This field must be set to the symbolic value FAB_C_BLN.

FAB_L_NAM Field

The name block field specifies the address of a name (NAML) block used to invoke a file service, such as an Open or Create. The NAML block is required only in conjunction with services that need to process file specifications. But it can also be used with other services - typically to obtain a file specification string after logical name translation is completed and defaults are applied. A value of 0 indicates no NAML block.

FAB_L_STS Field

The completion status code (STS) field is set with success or failure codes upon return from various file services.

FAB_L_XAB Field

The extended attribute block address (XAB) field specifies the address of a XAB (which may chain to other XABs). A value of 0 indicates no XAB(s) for the file.

For some operations, such as CREATE, you must associate extended attribute blocks with a FAB to convey additional attributes about a file. Other services will write data about the file to those blocks.

FAB_Q_HANDLE Field

The handle associated with an open file. This is written upon OPEN or CREATE, and should not be modified until after the file is closed.

Name Block (NAML)

Name Block (NAML)

The name (NAML) block provides fields for extended file specifications, including parsing and obtaining the

actual file specification. The contents of the NAML block, in the memory order are as follows:

Field	Size, in bytes	Description
NAML_B_BID	1	Block identifier
NAML_B_BLN	1	Block length
NAML_B_NOP	1	Operation flags
NAML_W_FID	8	File-system specific file ID
NAML_L_FNB	8	File name status bits
NAML_L_FILESYS_NAME	8	File system name buffer address.
NAML_L_FILESYS_NAME_ALL OC	4	File system name buffer allocated size, in bytes
NAML_L_FILESYS_NAME_SIZ E	4	File system name length, in bytes
NAML_L_INPUT_FLAGS	4	Additional flags specified as input
NAML_L_LONG_DEFNAME	8	Default file specification string address specified as input (used if FAB\$L_DNA contains -1)
NAML_L_LONG_DEFNAME_SIZ E	4	Default file specification string size specified as input
NAML_L_LONG_DEV	8	Device string address
NAML_L_LONG_DEV_SIZE	4	Device string length
NAML_L_LONG_DIR	8	Path string address
NAML_L_LONG_DIR_SIZE	4	Path string length
NAML_L_LONG_EXPAND	8	Expanded string area address
NAML_L_LONG_EXPAND_ALL OC	4	Expanded string area size
NAML_L_LONG_EXPAND_SIZE	4	Expanded string length
NAML_L_LONG_FILENAME	8	Specification string address
NAML_L_LONG_FILENAME_SIZ E	4	Specification string size
NAML_L_LONG_NAME	8	File name string address
NAML_L_LONG_NAME_SIZE	4	File name string length
NAML_L_LONG_NODE	8	Node name string address
NAML_L_LONG_NODE_SIZE	4	Node name string length
NAML_L_LONG_RESULT	8	Resultant string area address
NAML_L_LONG_RESULT_ALLO C	4	Resultant string area size
NAML_L_LONG_RESULT_SIZE	4	Resultant string length
NAML_L_LONG_TYPE	4	File type string length
NAML_L_LONG_TYPE_SIZE	8	File type string address
NAML_L_LONG_VER	8	File version string address
NAML_L_LONG_VER_SIZE	4	File version string length
NAML_L_OUTPUT_FLAGS	4	Additional status bits passed as output
NAML_Q_USER_CONTEXT	8	User context

The name block is required for certain operations, such as OPEN and CREATE. Depending on the call, the application may need to allocate space to receive result strings.

NAML_B_BID Field

The block identifier (BID) field identifies this control block as a name block. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value `NAML_C_BID`.

NAML_B_BLN Field

The block length (BLN) field defines the length of the NAML block, in bytes. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value `NAML_C_BLN`.

NAML_L_FILESYS_NAME Field

The address of a buffer to receive the fully-qualified file specification.

NAML_L_FILESYS_NAME_ALLOC Field

The size of the buffer whose address is contained in the `NAML_L_FILESYS_NAME` field.

NAML_L_FILESYS_NAME_SIZE Field

Indicate the length, in bytes, of the name string returned in `NAML_L_FILESYS_NAME`.

NAML_L_LONG_DEFNAME Field

This field replaces the `FAB_L_DNA` field in the FAB.

NAML_L_LONG_DEFNAME_SIZE Field

This field replaces `FAB_B_DNS` field in the FAB.

NAML_L_LONG_DEV Field

Points to the start of the device name within the complete file specification in the buffer.

NAML_L_LONG_DEV_SIZE Field

The length, in bytes, of the device name pointed to by `NAML_L_LONG_DEV`, including the ":".

NAML_L_LONG_DIR Field

Points to the directory specification within the complete file specification in the buffer.

NAML_L_LONG_DIR_SIZE Field

The length, in bytes, of the directory pointed to by `NAML_L_LONG_DIR`, including the \ delimiters.

NAML_L_LONG_EXPAND Field

The address of a buffer in the application program to receive the file specification string resulting from the translation of logical names and the defaulting of the file specification. You must specify this field for processing wildcards.

NAML_L_LONG_EXPAND_ALLOC Field

Contains the size of the buffer whose address is contained in the `NAML_L_LONG_EXPAND` field.

NAML_L_LONG_EXPAND_SIZE Field

Contains the length, in bytes, of the file specification string returned in the buffer whose address is in `NAML_L_LONG_EXPAND`.

NAML_L_LONG_FILENAME Field

Address of a buffer containing the filename. If `FAB_L_FNA` contains -1 and `FAB_B_FNS` contains 0, this field is used in place of `FAB_L_FNA`.

NAML_L_LONG_FILENAME_SIZE Field

Contains the length of the filename in the buffer specified by `NAML_L_LONG_FILENAME`. If `FAB_L_FNA` contains -1 and `FAB_B_FNS` contains 0, this field is used in place of `FAB_L_FNS`.

NAML_L_LONG_NAME Field

If `NAML_L_LONG_RESULT_SIZE` is 0, this field points into the buffer specified by

NAML_L_LONG_EXPAND. Otherwise, it points into the buffer specified by NAML_L_LONG_RESULT. The pointer points to the start of the file name within the complete file specification in the buffer.

NAML_L_LONG_NAME_SIZE Field

The length, in bytes, of the file name pointed to by NAML_L_LONG_NAME, up to but not including the type field or the period separating the name from the type - or up to the version if no type is specified.

NAML_L_LONG_NODE Field

If NAML_L_LONG_RESULT_SIZE is 0, this field points into the buffer specified by NAML\$L_LONG_EXPAND. Otherwise, it points into the buffer specified by NAML\$L_LONG_RESULT. The pointer points to the start of the node name within the complete file specification.

NAML_L_LONG_NODE_SIZE Field

The length, in bytes, of the node name pointed to by NAML_L_LONG_NODE, including the :: delimiter.

NAML_L_LONG_RESULT Field

Contains the address of a buffer in the application to receive the resulting file specification string. The NAML_L_LONG_RESULT_ALLOC field must also be specified in order to obtain a file specification. The file specification includes resolution of logical name and defaulting.

NAML_L_LONG_RESULT_ALLOC Field

Size of the buffer, in bytes, whose address is contained in the NAML_L_LONG_RESULT field.

NAML_L_LONG_RESULT_SIZE Field

The resulting string length, in bytes, of the file specification string returned in the buffer whose address is in the NAML_L_LONG_RESULT field.

NAML_L_LONG_TYPE Field

If NAML_L_LONG_RESULT_SIZE is 0, this field points into the buffer specified by NAML_L_LONG_EXPAND. Otherwise, it points into the buffer specified by NAML_L_LONG_RESULT. The pointer points to the start of the file type, including the dot separating it from the name, within the file specification in the buffer.

NAML_L_LONG_TYPE_SIZE Field

The length, in bytes, of the file type pointed to by NAML_L_LONG_TYPE.

NAML_L_LONG_VER Field

If NAML_L_LONG_RESULT_SIZE is 0, this field points into the buffer specified by NAML_L_LONG_EXPAND. Otherwise, it points into the buffer specified by NAML_L_LONG_RESULT. The pointer points to the start of the file version, including the semicolon delimiter, within the file specification in the buffer.

NAML_L_LONG_VER_SIZE Field

The length, in bytes, of the file version pointed to by NAML_L_LONG_VER.

NAML_Q_USER_CONTEXT

This field is reserved for the user. UOS doesn't modify this value.

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

Allocation Control XAB (XABALL)

Allocation Control XAB (XABALL)

The allocation control block provides additional control over file or space allocation on stores in order to optimize performance.

Field	Size,	Description

	in bytes	
XAB_B_COD	1	Type code (must be XAB_C_ALL)
XAB_B_BLN	1	Block length
XAB_L_NXT	8	Next XAB address
XAB_B_AID	1	Area identification number
XAB_B_ALN	1	Alignment boundary type
XAB_L_ALQ	4	Allocation quantity
XAB_B_AOP	1	Allocation options
XAB_B_BKZ	1	Bucket size
XAB_W_DEQ	2	Default extension quantity
XAB_L_LOC	4	Location
XAB_W_RFI	8	Related file identifier or FILE_NAME
XAB_W_VOLL	2	Related volume number

XAB_B_BLN

The block length (BLN) field defines the length of the XABALL, in bytes. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value XAB_C_ALLLEN.

XAB_B_COD

The type code (COD) field identifies this control block as a XABALL. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value XAB_C_ALL.

XAB_L_NXT Field

The next XAB address (NXT) field specifies the address of the next XAB in the XAB chain. A value of 0 indicates that the current XAB is the last, or only, XAB in the chain.

Date and Time XAB (XABDAT)

Date and Time XAB (XABDAT)

The date and time XAB (XABDAT) provides control of date and time attributes of file.

Field	Size, in bytes	Description
XAB_B_COD	1	Type code (must be XAB_C_DAT)
XAB_B_BLN	1	Block length
XAB_L_NXT	8	Next XAB address
XAB_Q_ACC	8	Last access date and time
XAB_Q_BDT	8	Backup date and time
XAB_Q_CDT	8	Creation date and time
XAB_Q_EDT	8	Expiration date and time
XAB_Q_RDT	8	Revision date and time
XAB_W_RVN	2	Revision number

XAB_B_BLN Field

The block length (BLN) field defines the length of the XABDAT in bytes. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value XAB_C_DATLEN.

XAB_B_COD Field

The type code (COD) field identifies this control block as a XABDAT. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value XAB_C_DAT.

XAB_L_NXT Field

The next XAB address (NXT) field contains the address of the next XAB to be used. A value of 0 indicates that the current XAB is the last, or only, XAB in the chain.

Created with the Personal Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

File Header XAB (XABFHC)

File Header XAB (XABFHC)

The file header XAB (XABFHC) contains the header information for the file.

Field	Size, in bytes	Description
XAB_B_COD	1	Type code (must be XAB_C_FHC)
XAB_B_BLN	1	Block length
XAB_L_NXT	8	Next XAB address
XAB_B_ATR	1	Record attributes; equivalent to FAB_B_RAT
XAB_B_BKS	1	Bucket size; equivalent to FAB_B_BKS
XAB_W_DXQ	2	Default file extension quantity; equivalent to FAB_W_DEQ
XAB_Q_EOF	8	End-of-file (logical file length)
XAB_W_GBC	2	Default global buffer count
XAB_L_HBK	8	Highest virtual block in the file; equivalent to FAB_L_ALQ
XAB_B_HSZ	1	Fixed-length control header size; equivalent to FAB_B_FSZ
XAB_W_LRL	2	Longest RMS record length in file
XAB_W_MRS	2	Maximum record size; equivalent to FAB_W_MRS
XAB_B_RFO	1	File organization and record format; combines FAB_B_RFM and FAB_B_ORG
XAB_L_SBN	8	Starting logical block number for the file if it is contiguous; otherwise this field is 0
XAB_W_VERLIM IT	2	Version limit for the file
XAB_Q_SIZ	8	Size on disk
XAB_Q_USZ	8	Uncompressed size
XAB_L_CLS	4	Clustersize
XAB_L_CRE	4	Creator UIC
XAB_W_FLG	2	Flags (not including protection codes)

XAB_B_BLN Field

The block length (BLN) field defines the length of the XABFHC, in bytes. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value

XAB_C_FHCLEN.

XAB_B_COD Field

The type code (COD) field identifies this control block as a XABFHC. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value XAB_C_FHC.

XAB_L_CLS Field

The clustersize (CLS) field defines the cluster size for the file, in bytes.

XAB_L_CRE Field

The creator (CRE) field defines the UIC of the user that created the file.

XAB_Q_EOF Field

The end-of-file (EOF) field defines the logical length of the file, in bytes.

XAB_Q_SIZ Field

The size (SIZ) field defines the physical length of the file, in bytes.

XAB_Q_USZ Field

The uncompressed size (USZ) field defines how large the file would be if uncompressed.

Created with the Personal Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

Item List XAB (XABITM)

Item List XAB (XABITM)

The item list XAB (XABITM) provides additional information for file operations. Each XAB item block points to an item list that includes one or more entries representing either a set or sense function.

Field	Size, in bytes	Description
XAB_B_COD	1	Type code (must be XAB_C_ITM)
XAB_B_BLN	1	Block length
XAB_L_NXT	8	Next XAB address
XAB_L_ITEMLIST	8	Address of item list
XAB_M_MODE	1	Mode (XAB_K_SENSEMODE or XAB_K_SETMODE)

XAB\$B_BLN Field

The block length (BLN) field defines the length of the XABITM, in bytes. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value XAB_C_ITMLEN.

XAB\$B_COD Field

The type code (COD) field is a static field that identifies this control block as a XABITM. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value XAB_C_ITM.

XAB\$L_NXT Field

The next XAB address (NXT) field contains the address of the next XAB to be used. A value of 0 indicates that the current XAB is the last, or only, XAB in the chain.

Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

Key Definition XAB (XABKEY)

Key Definition XAB (XABKEY)

The Key Definition XAB (XABKEY) control block defines a key.

Field	Size, in bytes	Description
XAB_B_COD	1	Type code (must be XAB_C_KEY)
XAB_B_BLN	1	Block length
XAB_L_NXT	8	Next XAB address
XAB_L_COLNAME	4	Collating sequence name
XAB_L_COLSIZE	4	Collating sequence table size
XAB_L_COLTBL	8	Collating sequence table address
XAB_B_DAN	1	Data bucket area number
XAB_B_DBS	1	Data bucket size
XAB_W_DFL	2	Data bucket fill size
XAB_B_DTP	1	Data type of the key
XAB_L_DVB	8	First data bucket virtual block number
XAB_B_FLG	1	Key options flag
XAB_B_IAN	1	Index bucket area number
XAB_B_IBS	1	Index bucket size
XAB_W_IFL	8	Index bucket file size
XAB_L_KNM	8	Key name buffer address
XAB_B_LAN	1	Lowest level of index area number
XAB_B_LVL	1	Level of root bucket
XAB_W_MRL	2	Minimum record length
XAB_B_NS	1	Number of key segments
XAB_B_NUL	1	Null key value
XAB_W_POS0	2	Key position
XAB_W_POS1	2	Key position
XAB_W_POS2	2	Key position
XAB_W_POS3	2	Key position
XAB_W_POS4	2	Key position
XAB_W_POS5	2	Key position
XAB_W_POS6	2	Key position
XAB_W_POS7	2	Key position
XAB_B_PROLOG	1	Prolog level
XAB_B_REF	1	Key of reference
XAB_L_RVB	8	Root bucket virtual block number
XAB_B_SIZ0	1	Key size
XAB_B_SIZ1	1	Key size
XAB_B_SIZ2	1	Key size
XAB_B_SIZ3	1	Key size

XAB_B_SIZ4	1	Key size
XAB_B_SIZ5	1	Key size
XAB_B_SIZ6	1	Key size
XAB_B_SIZ7	1	Key size
XAB_B_TKS	1	Total key field size

XAB_B_BLN Field

The block length (BLN) field defines the length of the XABKEY block, in bytes. Once set, this field must not be altered until the control block is no longer needed. The XAB_B_BLN field must be set to the symbolic value XAB_C_KEYLEN.

XAB_B_COD Field

The type code (COD) field identifies this control block as a XABKEY. Once set, this field must not be altered until the control block is no longer needed. The XAB_B_COD field must be set to the symbolic value XAB_C_KEY.

XAB_L_NXT Field

The next XAB address (NXT) field specifies the address of the next XAB in the XAB chain. A value of 0 indicates that the current XAB is the last, or only, XAB in the chain.

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

Protection XAB (XABPRO)

Protection XAB (XABPRO)

The protection XAB (XABPRO) specifies ownership and protections for a file.

Field	Size, in bytes	Description
XAB_B_COD	1	Type code (must be XAB_C_PRO)
XAB_B_BLN	1	Block length
XAB_L_NXT	8	Next XAB address
XAB_L_ACLBUF	8	Address of buffer that contains ACL
XAB_L_ACLCTX	4	ACL positioning context
XAB_W_ACLLEN	8	Receives the length of an ACL during an Open or Display service
XAB_W_ACLSIZ	8	Length of buffer containing binary ACEs
XAB_L_ACLSTS	4	System error status for ACL processing
XAB_B_MTACC	1	Magnetic tape accessibility
XAB_W_PRO	2	File protection; contains four separate fields denoting protection for system, owner, group, and world
XAB_B_PROT_OPT	1	File protection options
XAB_L_UIC	4	User identification code

XAB_B_BLN Field

The block length (BLN) field defines the length of the XABPRO, in bytes. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value XAB_C_PROLEN.

XAB_B_COD Field

The type code (COD) field identifies this as a XABPRO. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value XAB_C_PRO.

XAB_L_NXT Field

The next XAB address (NXT) field specifies the address of the next XAB in the XAB chain. A value of 0 indicates that the current XAB is the last, or only, XAB in the chain.

XAB_W_PRO Field

The protection (PRO) field specifies the file's protection. There are four classes of users: System, Owner, Group, and World. This field consists of four 4-bit subfields, each of which specifies file access for one of the four user classes. The following flags define the protection codes.

Code	Meaning
PROTECTION_OWNER_READ	The owner can read from the file.
PROTECTION_OWNER_WRITE	The owner can write to the file.
PROTECTION_OWNER_DELET	The owner can delete the file or change it's protection code.
E	
PROTECTION_OWNER_EXECU	The owner can execute the file.
TE	
PROTECTION_GROUP_READ	Users belonging to the file's group can read from the file.
PROTECTION_GROUP_WRITE	Users belonging to the file's group can write to the file.
PROTECTION_GROUP_DELETE	Users belonging to the file's group can delete the file or change it's protection code.
PROTECTION_GROUP_EXECUT	Users belonging to the file's group can execute the file.
E	
PROTECTION_SYSTEM_READ	System users or those with the SYSPRV privilege can read from the file.
PROTECTION_SYSTEM_WRITE	System users or those with the SYSPRV privilege can write to the file.
PROTECTION_SYSTEM_DELET	System users or those with the SYSPRV privilege can delete the file or change it's protection code.
E	
PROTECTION_SYSTEM_EXECU	System users or those with the SYSPRV privilege can execute the file.
TE	
PROTECTION_WORLD_READ	Anyone can read from the file.
PROTECTION_WORLD_WRITE	Anyone can write to the file.
PROTECTION_WORLD_DELET	Anyone can delete the file or change it's protection code.
E	
PROTECTION_WORLD_EXECU	Anyone can execute the file.
TE	

XAB_L_UIC

The UIC of the user who is the current owner of the file.

Created with the Personal Edition of HelpNDoc: [Easily create Qt Help files](#)

Revision Date/Time XAB (XABRDT)

Revision Date/Time XAB (XABRDT)

The revision date/time XAB (XABRDT) provides revision date/time for when the file is closed. This requires Control access to the file. If specified, this overrides the default revision timestamps that UOS normally applies to a file when it is closed.

Field	Size, in bytes	Description

XAB_B_COD	1	Type code (must be XAB_C_RDT)
XAB_B_BLN	1	Block length
XAB_L_NXT	8	Next XAB address
XAB_Q_RDT	8	Revision timestamp
XAB_W_RVN	2	Revision number

XAB\$B_BLN Field

The block length (BLN) field defines the length of the XABRDT, in bytes. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value XAB_C_RDTLEN.

XAB\$B_COD Field

The type code (COD) field is a static field that identifies this control block as a XABRDT. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value XAB_C_RDT.

XAB\$L_NXT Field

The next XAB address (NXT) field contains the symbolic address of the next XAB to be used. A value of 0 indicates that the current XAB is the last, or only, XAB in the chain.

XAB\$Q_RDT Field

The revision date/time (RDT) field contains a 64-bit internal UOS timestamp value expressing the date and time when the file was last opened for modifications.

Summary XAB (XABSUM)

Summary XAB (XABSUM)

The summary XAB (XABSUM) defines the total number of keys and areas defined for RMS files.

Field	Size, in bytes	Description
XAB_B_COD	1	Type code (must be XAB_C_SUM)
XAB_B_BLN	1	Block length
XAB_L_NXT	8	Next XAB address
XAB_B_NOA	1	Number of allocation areas defined for the file
XAB_B_NOK	1	Numbers of keys defined for the file
XAB_W_PVN	2	Prolog version number

XAB_B_BLN Field

The block length (BLN) field defines the length of the XABSUM, in bytes. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value XAB_C_SUM.

XAB_B_COD Field

The type code (COD) field identifies this control block as a XABSUM. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value XAB_C_SUM.

XAB_L_NXT Field

The next XAB address (NXT) field contains the address of the next XAB. A value of 0 indicates that the current XAB is the last, or only, XAB in the chain.

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

Terminal XAB (XABTRM)

Terminal XAB (XABTRM)

The terminal XAB (XABTRM) provides for extended terminal read operations to occur when a read service is used for a terminal device. Instead of being associated with a FAB, this control block is associated with a RAB.

Field	Size, in bytes	Description
XAB_B_COD	1	Type code (must be XAB_C_TRM)
XAB_B_BLN	1	Block length
XAB_L_NXT	8	Next XAB address
XAB_L_ITMLST	8	Item list address
XAB_W_ITMLST_LEN	2	Item list length

XAB_B_BLN Field

The block length (BLN) field defines the length of the XABTRM, in bytes. Once set, this field must not be altered until the control block is no longer needed. This field must be set to the symbolic value XAB_C_TRMLEN.

XAB_B_COD Field

The type code (COD) field identifies this control block as a XABTRM. Once set, this field must not be altered unless the control block is no longer needed. This field must be set to the symbolic value XAB_C_TRM.

XAB_L_NXT Field

The next XAB address (NXT) field contains the address of the next XAB to be used. A value of 0 indicates that the current XAB is the last, or only, XAB in the chain.

Created with the Personal Edition of HelpNDoc: [Full-featured Help generator](#)

Record Access Block (RAB)

Record Access Block (RAB)

The Record Access Block (RAB) allows applications to specify RMS records to access and identify record characteristics.

Field	Size, in bytes	Description

RAB_Size	2	Size of RAB structure, in bytes
RAB_L_BKT	8	Bucket to read (for RMS) or byte offset in file
RAB_W_ISI	8	File handle
RAB_L_KBF	8	Key buffer address (used as input only with random access by relative record number mode)
RAB_B_KSZ	1	Key size (used only if RAB_B_RAC is KEY and the file is a relative file)
RAB_B_RAC	1	Record access mode (Binary, SEQ, KEY)
RAB_Q_RBF	8	Record buffer address
RAB_L_RHB	8	Record header buffer (for variable with fixed control records only)
RAB_W_RSZ	4	Record/Prompt size
RAB_L_ROP	4	Record-processing options
RAB_B_TMO	1	Timeout period: a value of 0 indicates that RMS should not wait to complete a Put service (for mailbox devices only)
RAB_L_UBF	8	User user buffer address.
RAB_W_USZ	4	Size of user buffer
RAB_AST_Err	8	Asynchronous failure handler
RAB_AST_Succe ss	8	Asynchronous successful completion handler
RAB_Data_Strea m	8	Data stream in file
RAB_L_FAB	8	Address of FAB for the file
RAB_L_XAB	8	Address of first XAB for the file.

RAB_L_FAB

Address of the FAB for the file. This is not necessary if the file handle is provided and can be set to 0 in such case.

RAB_L_XAB

Address of the first XAB for this RAB. If 0, there are no XABs associated with the RAB.

RAB_W_ISI Field

The file handle associated with the file.