

To: Distribution
 From: Susan Barr
 Date: 02/10/77
 Subject: Directory Checking and Salvaging

This MTB describes the syntactic checks on directory structure to be made by the proposed new hierarchy salvager. This is one of two MTBs on the proposed salvager. A later MTB will describe the salvager control structure and general organization. The requirements for salvaging a directory can be broken into two goals:

1. Make the directory usable. All entries in the directory must be fixed or deleted. All checks that the system makes or relies on must be made. For example, backup checks initial ACL's, and append relies on the hash table, so the salvager should check these values for consistency.
2. Compact the directory.

A directory has a fixed header followed by an area. (This area has a special format that is managed by fs_alloc). All information about the directory as well as its entry name hash table is allocated in blocks in this area. The word entry has been used to mean a block of information in the area (ie. ACL entry, name entry) as well as its more widely used meaning to describe an object cataloged in a directory (segment, link, directory). To avoid this conflict the word block will be used to describe the information in the directory area and the word entry will be used for objects cataloged by a directory.

The directory is organized with chains of the same type of block (ie. initial ACL chain). Each chain is doubly threaded with pointers to the beginning and end of the chains being kept in the header. Total counts for the chains are also kept in the header.

Each block of information has the following form:

```

! next_block          ! previous_block  !
! type                ! size          !
!                    ! (information) !
! checksum           !              !

```

Multics Project internal working documentation. Not to be reproduced or distributed outside of the Multics project.

owner	
next_block	relative pointer to the next block in this chain (Zero indicates the end of the chain)
previous_block	relative pointer to the previous block in this chain. (Zero indicates the start of the chain)
type	a number from 1 to 11 indicating the type of block.
information	the actual information used by directory control.
checksum	planned for future implementation
owner	the UID of the owner of this block. For entries, access names, initial ACL's and the hash table the owner is the UID of the directory. For names and ACL's the owner is the UID of its entry.

The syntactic processing of each directory consists of three consecutive passes each of which is dependent on the success of the previous pass. They are:

1. Checking the directory. The salvager "walks" the chains for the different types of information blocks and makes consistency checks.
2. Rebuilding the directory. If an error was detected in the checking mode, then a new copy of the directory is created using only the consistent blocks of information from the old directory. In the worst case of a bad header, this would cause an empty directory. This differs from the old salvager which deleted those directories with bad headers.
3. Recovering lost items. If losses of information were detected during the first pass, then a search is made for blocks that are in the directory, but are not reachable by a chain due to destruction of the forward and backward chain.

CHECKING THE DIRECTORY

The salvager checks the header and walks the chains using information from the header. It duplicates the checks made by directory control, checks data bases that are relied on by directory control and makes some additional consistency tests to make salvaging more reliable.

In addition to checking the consistency of each block, the salvager also keeps track of the space used by all blocks. This is used to detect chains which loop back on themselves and to detect invalid chain threads. These bad threads can be caused by an out of date page of the directory (one that was not updated after a change was made due to a system crash). The salvager checks the chain pointers, type, size and owner. If all these fields are valid and the space is free then the space occupied by the block is marked as used. The first valid block claims the space.

Another check for the validity of the chain threads is made by comparing the block extents against the size of the directory. The number of words used by the directory is given in the area header. This field is assumed correct if it is consistent with the number of pages used by the directory.

The salvager checks and rebuilds directories in a fixed order:

1. header and area header
2. hash table
3. access name chains for person names and project names
4. entry chain (segments, links, directories). The chain of names and the chain of ACL's for each entry are checked as the entry is processed.
5. Initial ACL chains for segments and directories for each ring.

This order is used for convenience in checking and to achieve the correct order in the case of a rebuild. (The hash table is referenced most often and should be at the beginning of the directory for the rebuild case. Initial ACL's should be at the end since they are referenced the least often. The access names have to be checked before the entry chain which will refer back to them.)

The salvager verifies each block of information using two levels of checks. The first set of checks determines the validity of the structure of the block. These checks are the same for all types of blocks. The second set of checks insures that the information within the block is plausible. These checks are specific to the type of block. If an error is detected in

either case, the salvager normally switches from checking mode to rebuild mode. In the following list, the action taken by the salvager if the check fails is given in brackets.

Structure Checks:

1. The block is within the directory.
[the block is ignored and the space is not marked used]
2. The block does not overlap storage already marked as used by previous blocks.
[the block is ignored and the space is not marked used]
3. The owner field matches the one specified.
[the block is ignored and the space is not marked used]
4. The type found is compatible with the type expected.
[item is corrected]
5. The size field is correct for the type.
[item is corrected]
6. The back relative pointer is correct.
[item is corrected]

Information Plausibility Checks:

HEADER

version_number Must be the correct value.
 [item is corrected]

arearpo Must have the value expected.
 [item is corrected]

all totals in the header

As the salvager checks the directory, it keeps a count of the number of blocks found. After every block has been verified, it compares those counts to the totals in the directory header.

segment count
directory count
link count
ACL count
initial ACL counts for segments and directories for rings 1 to 7

The following fields are compared to values supplied by the caller (which gets the information from the branch). If the comparison fails, a rebuild occurs and the supplied values are used. These checks are optional so that the directory checker procedure can be used by the retriever which doesn't have this information available.

access_class	security attributes of directory
ovid	physical volume identifier
sons_lvid	logical volume identifier for segments contained in this directory
vtocx	vtoc entry index of the directory
master_dir_uid	UID of superior master directory
tree_depth	number of levels from root of this directory

AREA HEADER

The area header must contain the expected values for the maximum directory size and the list of sizes permitted for allocation.

HASH TABLE

The salvager loops through the hash table. For each index in the hash table, it follows the chain of all names that hashed to that index. Each address in the hash chain must point to a name block. The hash index within the name block must match the index of the hash table being checked.

ACCESS NAME

ASCII The name must contain only ASCII characters.
[block is deleted]

ACL

pers_rp If a person name relative pointer is
 non-zero, it must specify an access name
 block within the directory.
 [block is deleted]

proj_rp Same as pers_rp for the relative pointer to
 the project name.

tag The two high order bits of the tag must be
 zero.
 [block is deleted]

DIRECTORY BRANCH

bs Branch switch must be on.
 [item is corrected]

dirsw Directory switch must be on.
 [item is corrected]

ovid Must not be zero.

nnames The number of names must be greater than 0.
 [a unique name is supplied]

multiple_class (AIM)
 If the multiple_class switch is set, the the
 access class must be greater or equal to the
 parent. [the security out of service bit is
 set.]

dtd Must be earlier than salvage time.
 [item is set to time salvaged]

dtem Must be earlier than salvage time.
 [item is set to time salvaged]

UID Must be within the expected value. (Since
 UIDs are a function of time only values
 between some fixed early date and the time of
 salvage are valid.)
 [block is deleted]

LINK

bs The branch switch must be off.
 [item is corrected]

pathname_size The pathname_size must fit in the block and be no larger than the maximum size.
[item is corrected]

nnames The number of names must be greater than 0.
[a unique name is supplied]

dtd Must be earlier than salvage time.
[item is set to time salvaged]

dtem Must be earlier than salvage time.
[item is set to time salvaged]

ENTRY NAME

ASCII

The name must contain only ASCII characters
[block is deleted]

SEGMENT BRANCH

bs

The branch switch must be set.
[item is corrected]

dirsw

The directory switch must be off.
[item is corrected]

pvid

Must not be zero.

nnames

The number of names must be greater than 0.
[a unique name is supplied]

multiple_class (AIM)

If the bit is set, the segment must have access greater than the parent. Otherwise the segment must be equal to parent.
[the out of service bit is set.]

dtd

Must be earlier than salvage time.
[item is set to time salvaged]

dtem

Must be earlier than salvage time.
[item is set to time salvaged]

UID

Must be within the expected value.
[block is deleted]

REBUILDING

When an error is detected in the checking pass, the salvager normally switches to the rebuild mode. (This does not happen if the salvager is being run in check only mode.) In the rebuild mode a copy of the directory is created by using the valid blocks of information from the old directory. If an uncorrectable error is found in the structure of a block, then no information from that block will be used including the forward chain thread. In this case the tail pointer for the chain is used to "walk" the chain backward so that the rest of the blocks of information can be recovered.

SEARCH FOR MISSING INFORMATION

This part of the salvager is used only if losses of information are detected during the check/rebuild pass. A loss could be detected in two ways: the value of a total count in the header was not equal to the number of items found; or it was not possible to reach all blocks of information on a chain using both the forward and backward threads.

The salvager has already marked the space that is occupied by valid blocks. It loops through the directory area free chain and marks the blocks that have been released. At this point, in a valid directory, all the space would be marked as used. If there is any space left that is not marked used, the salvager tries to recover the missing blocks within that space.

It loops through the unused space looking for a word that has one of the valid type and size combinations. If the entire block specified by the size is unused, then the salvager applies the same structure and plausibility checks mentioned above. The block must be valid -- no corrections will be made in this pass. If the block is valid, it is threaded onto the proper chain in the rebuilt directory. ACL's and ACL names cannot be recovered in this way since the order and completeness of an ACL is critical to access protection.

OPTIONS

The following options can be specified to modify the default action described above.

- rebuild Rebuild the directory in all cases. This can be used when the caller knows a rebuild will be required. For example, the online salvager would always specify rebuild.
- compact Force a rebuild of the directory, if one or more pages can be freed. The procedure sums the block sizes on the directory area free list to determine the amount of storage that will be freed.
- check Make all checks and log any error messages, but do not rebuild the directory. This can be used for testing purposes.
- debug Cause additional error messages to be logged.