

To: Distribution  
From: T. H. Van Vleck  
Date: June 18, 1975  
Subject: SAVE and RESTOR for New Storage System

## PURPOSE

This memorandum is a short note on the differences between the BOS commands SAVE and RESTOR on the current system and the new Multics storage system. It is intended for BOS maintainers and people of similar expertise.

## OVERVIEW

Until the new backup system is written for the new storage system we will want a simple package for saving and restoring the contents of volumes for the new storage system. The SAVE and RESTOR for the current system require redefinition because they depend on the format of the FSDCT and the notion of device ID, both of which are altered in the new system.

The overall function of SAVE remains the same: that is, SAVE is to write the contents of a specified portion of the system's secondary storage onto tape (or onto an IO disk). RESTOR is to be able to reload these tapes (disks) in such a fashion that Multics operation can pick up where it was suspended.

## PROPOSED CHANGES

Four different BOS programs are affected by the changes introduced by the new storage system. They are:

1. save.alm - the SAVE command
2. restor.alm - the RESTOR command

---

Multics Project Internal working documentation. Not to be reproduced or distributed outside the Multics Project.

3. sav\_util.incl.alm - a utility package which manages functions common to both SAVE and RESTOR. This includes the interpretation of the FSDCT bit map and the loop over device ID's.
4. argmul.incl.alm - a utility package which parses storage address ranges typed by the operator into a pair of arrays defining the amount of storage to be dumped or reloaded.

In addition, the format of the tapes produced by SAVE will be different from the current format.

### Changes to ARGMUL

The ARGMUL package now returns its answer in the pair of arrays "amfirst" and "amlast". Each is 8 words long, and indexed by device ID. These are interpreted to specify that all storage addresses from amfirst(i) to amlast(i) for each device ID i are to be dumped or reloaded. ARGMUL is the program in BOS which types the familiar sequence of sixteen numbers when it is invoked by SAVE, RESTOR, TEST, and FMT.

The new version will have to be able to handle more storage, specified differently. In particular, we need to be able to specify that many volumes should be dumped, or that only one physical volume be dumped, or that less than one volume be dumped; or any combination of these. The limited size of the BOS command line (20 words) will be a problem in some cases.

The internal list of addresses to be dumped must also be reformatted, since it may be considerably larger, and since additional configuration flexibility has been introduced into the system with the new CONFIG deck format.

### OUTPUT TABLE

The output from ARGMUL will be a table with the following form:

```
dcl 1 rqst (100) aligned,
    2 volname char (4),
    2 lvname char (4),
    2 frec fixed bin (35),
    2 lrec fixed bin (35),
    2 into fixed bin (35),
    2 dvt_word,
    3 low bit (3) unal,
```

```
3 chn bit (6) unal,  
3 dev bit (9) unal,  
3 typ bit (18) unal;
```

The quantity "into" will always start out equal to "frec"; its use is described below. The addresses "frec", "lrec", and "into" are all Multics addresses -- that is, they are record numbers, not sector numbers.

#### INPUT TO ARGMUL

There are three cases to distinguish here.

#### Requesting all Storage

The commonest form of argument to ARGMUL is the specification of all the storage available to Multics, as when the operator types

```
SAVE
```

In this case, the output table can be built easily by reading the configuration deck.

#### Requesting Specific Volumes or Addresses

The next-most-common use of ARGMUL is to specify a single volume, or a part of a volume. The address forms accepted by ARGBOS are valid here, with an optional TO clause. Examples:

```
SAVE VOL FRED  
SAVE VOL FRED 0 TO 6666.  
SAVE PHY D191 4 100 TO 400  
SAVE PART LOG
```

These address forms specify only one volume, or part of one volume. Therefore, they will generate only one entry in the "rqst" table.

## Requesting Multiple Specific Ranges

The final form, which allows the flexibility of multiple addresses of the forms interpreted by ARGBOS, is difficult to accommodate since the command line is limited in length. Therefore, the keyword LIST in the command line will cause ARG MUL to read multiple input lines and parse each separately until a line containing END is encountered. For example:

```
SAVE LIST
VOL MX01
VOL MX02
VOL MX03
QUIT
```

## Changes to SAVE

The SAVE program must be modified to handle the new forms of ARG MUL output. In addition, the tape format must be changed to correspond with the new system organization.

## TAPE FORMAT

The first record on every physical reel will consist of the "rqst" structure output by ARG MUL. On the second and subsequent reels of a multi-reel SAVE, the "info" items of one or more volumes will be changed. If a volume has been completely processed, the "info" will be equal to the "irec" for that volume.

Identifying information such as date and time saved, and version number of the tape format, will also be stuck into the first record.

The next logical record on the tape will be the free storage map for the physical volume. Whenever SAVE switches to a new volume a new free storage map will be written out. Unless specific record numbers were specified, the usual case will be to dump entire volumes. In particular, the volume label and VTOC for each volume will be dumped as regular data records.

After that, logical records follow which contain the data from the volume. Unless the ALL argument was specified, records marked free in the free storage map will not be dumped. Each logical record is preceded on tape by a header which contains an

ID word. The format of the ID word is as follows:

```
dcl 1 id,
     2 volume_no bit (18) unal,
     2 record_no bit (18) unal;
```

The "volume\_no" item in this header is the index of the volume being dumped in the "rqst" structure. The NTAPE package trims trailing zeroes off the data on output.

### Changes\_to\_RESTOR

The RESTOR command will call ARGNUM, just as SAVE does, to obtain a "rqst" structure defining what is to be reloaded. It will then read the first record of the first tape, and compare the "rqst" table on the tape with the one from the command input source. Since configuration cards may have been re-ordered, the order of the "rqst"s from the operator may have to be changed to agree with the volume numbering specified on the tape. If the operator's request and the tape's contents are not identical, the RESTOR command will either abort, continue without comment, or print a question and ask the operator whether to abort or go on. When the operator asks for more to be restored than is on the tape set, RESTOR will always abort. If, however, the tape set contains more than was requested, this may be reasonable. The command will ask whether to go on, unless the argument ONLY has been specified on the command line.

If, for a given volume, the logical volume name in the CONFIG deck does not agree with the logical volume name on the tape, RESTOR will abort. The operator must correct the CONFIG deck and try again.

If the device model number for a given volume is different between the tape and the command input, RESTOR will print a comment and go on. However, a check will be made to insure that the device being loaded onto has a sufficient address range available to accommodate "irec" before starting to load.

Processing of the input tape is unchanged for data records. The choice of whether RESTOR reloads the volume label or not is determined by a new control argument for RESTOR, the LABEL argument. If LABEL is specified, then the volume label is overwritten by the one from tape. If LABEL is not specified, then the label from tape is checked against the one on disk, and RESTOR will abort if any of the following items disagree:

```
physical volume ID
logical volume ID
physical volume name
```

logical volume name  
manufacturer's serial number

The RESTOR command will update a field in the label which gives the date and time RESTORED in either case.

Another new option for RESTOR is the CHECK option, which will just print the contents directory of a reel of tape (or disk) created by SAVE.