**Paper 070-2010**

# Why Aren't You Using MEMLIB?

Erik Dilts, INC Research, Raleigh, NC

## ABSTRACT

In 2002, I presented a poster at SESUG titled "Using a RAM Disk to Speed up your Windows 2000 SAS Session." At that time, you had to purchase a third party RAM disk program, install it, and set it up correctly in order to use the technology. Due to this difficulty, the cost, and 21 CFR part 11 compliance issues, many companies were unwilling or unable to utilize it. Seven years later, SAS version 9 has changed all of that by including a de facto RAM disk program in Base SAS. It can be configured as a system option, setting up the WORK directory in memory, or as a LIBREF option within a SAS session. This paper will explain how to use MEMLIB (short for memory library) to set up a RAM disk. It will also demonstrate that for most SAS programmers, the savings from decreased program run time will quickly offset the $25-$50 cost for additional memory – which may not even be needed. It is written toward any skill level and applicable to SAS version 9 and up, running on Windows XP Pro/Vista.

## INTRODUCTION

The concept of a RAM disk is simple: set aside an area of the computer's memory (RAM - Random Access Memory) to serve as a simulated hard disk. While hard drive speeds have increased tremendously since RAM disks fell out of use, they are still **much** slower than RAM, both in data transfer speed and access time. The newest generation workstation disk drives have a maximum data transfer speed near 70 Mb/second and an access time of about 10 milliseconds. The latest RAM, on the other hand, transfers data at up to 6400 Mb/second (91 times faster) and an access time of 9 nanoseconds (roughly 1,000,000 times faster). Any application that reads and writes large amounts of data to the disk will benefit from these speed improvements. In SAS, they add up very quickly to provide a very substantial decrease in total running time.

## RAM DISK/MEMORY LIBRARY SETUP

```
%let dsets =sashelp.zipcode sashelp.zipcode sashelp.zipcode
            sashelp.zipcode sashelp.zipcode sashelp.zipcode
            sashelp.zipcode sashelp.zipcode sashelp.zipcode
            sashelp.zipcode sashelp.zipcode sashelp.zipcode
            sashelp.zipcode sashelp.zipcode sashelp.zipcode;


data TEST_ML;
  set &dsets;
run;

libname test1 "c:\temp1";
data test1.TEST_ML;
  set &dsets;
run;

libname test2 "c:\temp2" memlib;
data test2.TEST_ML;
  set &dsets;
run;
```

Testing was performed on a fairly standard setup: Lenovo/IBM T60 laptop with 1.5 MB of RAM. Laptop drives are generally slower than desktop drives so the benefits should be even greater when using that setup. The numbers shown are the means from five testing cycles.

| Startup | Setup | Results | | |
|---|---|---|---|---|
| | | total time | user CPU time | system CPU time |
| invoked* | (invoked) | 0.504 | 0.290 | 0.200 |
| invoked | standard | 22.308 | 0.370 | 0.652 |
| | | | | |
| standard | libname | 0.548 | 0.296 | 0.234 |
| standard | standard | 21.774 | 0.162 | 0.752 |

* When startup is “invoked”, SAS is started with the following shortcut parameters: -memmaxsz 1024M -memlib -memcache 4. This sets up a memory library 1024 MB (1 GB) in size, and specifies that SAS use it as a file cache.

The table shows the advantages of using a memory library. Stepping though the table, we see that line 1 only takes half of a second to write 116 MB to a dataset stored in memory. Writing the same dataset to a normal disk library takes 22.3 seconds, 44 times longer. While 22 seconds doesn’t sound like much of a time savings, if your program has ten data steps of this size and you have to run that program 10 times, you will save 36 minutes! In line 3 MEMLIB is used as a LIBNAME option and used for the data step. The time goes up slightly to 0.548 seconds; essentially the same speed gain. There are also gains to be had in most procedures:

```
proc sort data=TEST_ML out=TEST_ML2;
  by city countynm statename;
run;
```

In a memory library, this proc takes 9.18 seconds; in a normal library it takes 22.40 seconds.

```
proc sql noprint;
  create table TEST_ML2 as select distinct city, countynm,
    statename, count(distinct(zip))
  from TEST_ML group by city,countynm,statename
               order by city,countynm,statename;
quit;
```

In a memory library, this proc takes 2.38 seconds; in a normal library it takes 11.04 seconds.

```
proc means data =TEST_ML noprint;
  class city countynm statename;
  var county;
  output out=outstats
    n=n mean=mean median=median min=min q1=q1 q3=q3 max=max;
quit;
```

In a memory library, this proc takes 1.28 seconds; in a normal library it takes 1.38 seconds. There is essentially no gain here as the time it takes to run is limited by the CPU.

The reason benefits are realized with some procedures is twofold. First, the output dataset is written to the memory library so it gains the same benefits as the data step example. Second, SAS creates temporary work files in the WORK directory “behind the scenes.” These files are read and written much faster when WORK is a memory library. For these gains, using MEMLIB as a libname option does not offer any improvement – the WORK library can only be set up in memory by using the MEMLIB option at invocation.

## CONCLUSION

Memory libraries were considered nearly essential for time-efficient database processing in the 1980's but technology outgrew them as hard drives sped up at a greater rate than CPUs. Now, due to the extreme drop in the price of large memory modules and SAS Institute's MEMLIB option, they have regained their potential as a very easy and inexpensive way to more than halve the program run time for many disk-intensive procedures. Computers being sold today usually have 2 or more GB of memory installed, so setting up a 1 GB memory library requires no additional cost or equipment. If 1 GB of memory is installed, then a 512 MB memory library can be set up, but it has an increased chance of filling up, which usually causes SAS to generate an error and halt. If this is the case, another 1 GB memory module can be had for under $25, and the performance gains offered from this simple upgrade are far greater than those offered by buying a whole new computer setup with twice the raw CPU power of your current system.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Erik Dilts
Enterprise: INC Research, Inc.
Address: 4700 Falls of Neuse Road, Suite 400
City, State ZIP: Raleigh, NC 27609
Work Phone: 919-334-3532
Fax: 919-876-9360
E-mail: edilts@incresearch.com
Web: http://www.incresearch.com