Converting Exodus Storage Manger Applications from V2 to V3.0

This note explains how to get your E and SM programs running quickly with SM 3.0. It does not explain all the new features. SM 3.0 is very similar to SM 2.2 except in the area of configuration files and how volumes are accessed.

For each volume that an application uses, the SM client library must know its volume ID and the address of the server that manages it. This information is specified using the "mount" option. The mount option is different from others in that is can have multiple values. The example configuration file below explains how to use it.

Volume location and format information is now specified by options for the server and the formatvol program. The example configuration file explains how to use these options.

```
# ---- EXAMPLE .sm_config FILE ----
# This is an example configuration file for SM 3.0
# The client has a 400 page buffer pool.
client*bufpages:        400
#
# The client library finds out what volumes it can mount from the mount
# option.  The first number in the option value is the volume ID.  The
# second number is a port number for the server managing the volume.
# The address after "@" is the machine on which the server is running.
#
client*mount:           3001 9000@caseus.cs.wisc.edu
client*mount:           3002 9000@caseus.cs.wisc.edu



#
# Common server options
server*bufpages:    600
server*verbose:     yes
#
# The server "sm_server" will run on this port.  Notice that 9000
# matches the information in the client's mount option above.
#
server.sm_server.portname:  9000
```

```
#
# Here we tell the server and formatvol about the log and data
# volumes.
# The "[sf]*" prefix to the options makes them visible to
# servers and formatvol.
#
# The logformat option describes the log volume.
# The first part of the option is the name of the file containing
# the volume.  Next is the volume ID, the number of cylinders
# in the volume, the number of tracks/per cylinder, the number
# of pages/track, and finally the size (in K) of log pages.
#
[sf]*logformat: /var/home/zwilling/volume/log.4d 3000: 1: 1: 1000: 4

#
# The dataformat option describes the data volumes.
# It is identical to logformat except there is no log page size
# parameter.  Note that colons or white space can be used to
# delimit the parts of the option value.
#
[sf]*.dataformat:   /var/home/zwilling/volume/vol3001 3001 1 1 1000
[sf]*.dataformat:   /var/home/zwilling/volume/vol3002 3002 1 1 1000

#
# This specifies that the server, "sm_server", uses volume 3000 for a
# log.
#
server.sm_server.logvolume: 3000
```

Below are steps to follow to convert to SM 3.0:

FOR ALL USERS

(1) Use the example configuration file below to update your current configuration file(s).

(2) Run the 3.0 formatvol to reformat your log and data volumes. Here is an example for the volumes in the configuration file below:

```
formatvol -vol 3000
formatvol -vol 3001
```

You can get info on the volumes using:

```
formatvol -dis 3000
formatvol -dis 3001
```

(3) Start `sm_server`.

FOR E USERS

(4) Relink your E programs (with the new E library and the new SM client library).

(5) To specify the volume for data, use the EVOLID environment variable instead of EVOLUME. For example:

```
setenv EVOLID 3001
```

(6) Run your program.

FOR DIRECT SM USERS

(4) Update your programs. Sm_Initialize() no longer takes any parameters. You no longer should call sm_MountVolume() or sm_DismountVolume() Your programs still probably have some "volid" variable to initialize. A common way to do this is to use the value of the EVOLID environment variable. If your code previously did something like:

```
sm_MountVolume(getenv(EVOLUME), &volid);
```

now just use:

```
volid = atoi(getenv(EVOLID));
```

then "setenv EVOLID" to a volume given with the mount option in a .sm_config file

(5) Recompile your programs and link with the new client library (`libsm_client.a`). You must use g++ to compile your programs or else link in the gnu library (at UW this is in `/usr/gnu/lib/gcc-lib/decstation/2.3.3/libgcc.a`).

(6) Run your program.