Configuring and Running the Shore Server[1]

The Shore Project Group
Computer Sciences Department
UW-Madison
Madison, WI
Version 1.1.1

October 27, 1997

[1]This research is sponsored by the Advanced Research Project Agency, ARPA order number 018 (formerly 8230), monitored by the U.S. Army Research Laboratory under contract DAAB07-92-C-Q508.

Contents

1 Introduction

This document describes how to start a Shore Shore Value-Added Server (SVAS). We assume that you have installed the software according to the instructions in the *installation manual*. Among other things, that document explains how to get a Shore server running. This document explains the process in somewhat more detail.

As explained in the document *An Overview of Shore*, Shore is a client/server system. The server component has a layered architecture built on a core *storage manager* that facilitates construction of custom-designed *value-added servers*. The distribution comes with a particular value-added server called the *Shore Value-Added Server* (*SVAS*), which includes support for logical object identifiers, a Unix-like namespace, and demand-fetching of objects into a client-side *object cache*. This document explains how to run the SVAS and how to control its execution by setting various options.

2 Configuring a Shore Server

The SVAS is a program that runs as a daemon process. Options control, among other things,

- whether it has a controlling terminal and TCL shell,
- whether it offers a Shore client RPC service,
- whether it offers Network File System (NFS) and NFS mount services,
- what ports it offers these services on,
- whether it uses TCP or shared memory for exchanging data with its clients.

The configuration choices are determined by the values of *configuration options*, which are set by the server at start-up time from a a configuration file containing commands of the form `optionname: value` and from command arguments of the form `-optionname value`. If

both the configuration file and the command line specify a value for a particular option, the command-line setting takes precedence.

The release contains an example configuration file, which you should find in `$SHORE/lib/options`. If you followed the instructions in the *installation manual*, you have already put an edited copy of the configuration file into `.shoreconfig` in your home directory. If not, follow the instructions in the section Running the Shore Server of the installation manual before continuing with these instructions. You can set the environment variable `SHORE_RC` to specify some location or file name other than `/.shoreconfig` for the configuration file.

See the manual page *options(SVAS)* for a description of all the configuration options.

See the manual page *environment(SVAS)* for a description of all the environment variables used by the SVAS.

3 Running a Shore Server

3.1 A Bootstrapping Problem

When the server starts, it locates its configuration file and reads it. The configuration option `svas_root` gives a pathname in the Unix file system of a device or file that is to serve as the root volume for Shore. The default is `./volumes/miniroot 10`, which means that Shore will use the Unix file `./volumes/miniroot` to simulate a disk device. (The "`10`" is an internal volume id to be assigned to this volume by Shore). Using Unix files to simulate disks is convenient for debugging, but for better performance, you should replace this name with the pathname of a real "raw" disk device (`/dev/...`). One or more options `svas_serve` may also be present, indicating other devices containing volumes to be served. When you start from scratch, you have no root directory, and you have to create one. The server is able to run without a root directory, but you can't do much with a server that has no file system, other than use it to format devices and make file systems (which correspond to volumes) on those devices.

After reading the configuration file, the server normally starts a TCL shell. TCL (Tool Command Language) is a package originally written by John Ousterhout at the University of California at Berkeley. It consists of an interpreted language and an extensive support library, including facilities for calling C function from TCL. The main use of TCL in Shore is to provide an interactive command-line interface to its components. The TCL shell started by the SVAS reads a script of start-up commands and then prompts for interactive input.

3.2 TCL command file (option `svas_shellrc`)

One of the configuration options, `svas_tclshell`, determines whether the server will start a TCL shell. If the value of `svas_tclshell` is `yes` or `true`, a shell is started, and the option `svas_shellrc` is inspected. (If the `svas_tclshell` option is not set by the configuration file or a command-line option, its default value is `true`.) If `svas_shellrc` is set, it names a file of *TCL commands*, which are interpreted when the shell starts. The default configuration file included with Shore sets this option to `/shore.rc`. If you followed the installation instructions, you copied `shore.rc` from the installation directory `$SHORE/lib` to your home directory. This file of TCL commands is a convenience, not a requirement. You may modify it or even remove it once you understand what it does.

The first time you run the SVAS, you are doing the Shore counterpart of re-installing Unix. The analogous activity in Unix is to reformat your disks and install Unix from scratch, making file systems with `mkfs(8)` or `newfs(8)`). There are enough steps to this process that you don't want to have to do it by hand each time. The script `shore.rc` automates this task. It determines if the root file system was mounted when the server started running. If not, the TCL commands format the root volume and create a Shore file system on it. The commands also create several directories that are required by the example applications.

Once your system is configured and running, and you are in "production mode", you can dispense with this script. As long as you don't intend to reformat your disks, you can start a server without an TCL commands; all you have to do in this case is to set the SVAS options `svas_root` and `svas_serve` so that the SVAS can mount the root directory and other volumes. On the other hand, it does not hurt anything to run the server with a TCL shell and to have it read the TCL command file each time it starts. If you want to replace the file of TCL commands with your own version, or if you want to avoid reading such a file altogether, edit your options file `.shoreconfig` and change or remove the command that sets the option `svas_shellrc`.

3.3 Running the Shore Server

If you followed the instruction for running the SVAS in Running the Shore Server, you saw several messages looking like this:

```
Warning: no database administrator -- running under userid 1417, groupid 1417
Looking for run command file "~/shore.rc" \ldots
format --./volumes/miniroot-- 20000 true
mkfs ./volumes/miniroot 20000 10
Done serving devices and making filesystems.

root = 0
setroot  10
/ is now 0.0.0.0:10.20007
See if  /  exists in the Shore namespace.
/  does exist
Done putting together the namespace.

Done reading ~/shore.rc.
Shore Server

Version ($Revision:  1.36 $ $Date:  1997/01/24 16:48:16 $)

(0)%
```

You can ignore most of these messages.

`Warning: no database administrator ...` announces that the server is not being run by the super-user, or by the user `shoreadm`.

`Looking for run command file "~/shore.rc" ...` If the server is configured to run a TCL shell (which it was), and if it is given a file to read when the shell starts (`shore.rc` in this case), it reads that file and interprets the commands therein. Among the commands in

`shore.rc` were commands to format the device and volume for the root directory, mount the root directory, and print the full object identifier of the root directory.

`format --./volumes/miniroot ...`

`mkfs ./volumes/miniroot ...` The script checks whether the "device" `./volumes/miniroot` exists. If not it assumes that the device is really a Unix file, which it creates and initializes to look like an empty file system by issuing the commands

```
format ./volumes/miniroot 20000 true
mkfs ./volumes/miniroot 20000 10
```

The number 20000 is the size of the file created, in kilobytes. If you want to use a raw device, you must change both the configuration option `svas_root` in the configuration file `.shoreconfig` and the variable `rootvolumename` in the startup file `shore.rc`. Also the check to see whether the device exists will always succeed, so you will have to run the `format` and `mkfs` commands "by hand" the first time to format the device.

The second and subsequent times you run SVAS, these lines of output will be replaced by

```
serve ./volumes/miniroot
```

meaning that the server has added this volume to the list of volumes for which it will act as a server.

`Shore Server ...` After reading the file of TCL commands, the server's TCL shell prints a welcome message and a prompt for TCL commands.

The SVAS is now running a TCL shell in the window in which you started the server. As you run clients, you can also run TCL shell commands to the SVAS. Unfortunately, this shell has no user guide, but is has some on-line help. Type "commands" or "help".

3.4 Shutting the Server Down

The server shuts down when you type the command `bye`, which is synonymous with `quit, q, exit,` and `x`. You can also run the program `sshutdown` directly from a Unix shell window. This program contacts the server and causes it to shut down. (This is the only way of cleanly shutting down the server if you ran it without a TCL shell).

3.5 Changing Your Environment

If you want to change the location of your logs or the location of your data volumes, you will need to edit `shore.rc` (or whatever file you have chosen as your file of TCL commands). *Warning: If your volumes or log are on a network file system (such as NFS or AFS) rather than a local disk, the server will be sluggish, and you must take care that the volumes and logs have the necessary permissions so that the user who runs the server can create and update the volumes and log files.*

The file of TCL commands does several things:

1. It defines a table of volumes on which to create file systems
2. It formats the volumes if necessary. **Warning: If your volumes are on a Unix raw device, you must do this step by hand.**
3. It determines if the root directory was already mounted when the configuration file was processed, and if not,
4. It creates a file system (with the release, this corresponds to a volume) for the root of the Shore name space ("/") , and mounts it,
5. It mounts any non-root volumes
6. It creates some directories and pools used by the example applications, if they don't already exist.

If you want to change the location or size of your file system, look at the volume table at the beginning of the file. The comments there should be self-explanatory.

4 NFS-Mounting Your File System

The Shore Value-Added Server is, among other things, an NFS daemon and a Mount daemon [SGK+85]. If you configure and run the server as it is distributed, the NFS and Mount services will be available. If you want to use them, you have to mount the Shore file system as an NFS file system on one or more of your workstations.

Take the following steps on each workstation from which you want to access to the Shore file system using NFS.

1. Become super-user by typing "`su`." Most versions of Unix do not allow anybody but the super-user to issue the `mount` command.
2. Identify or create a Unix mount point, which is an empty Unix directory on your workstation.
3. The installed `$SHORE/bin` directory must be in your (super-user's) path.
4. Run the script `mnt`, which takes 2 arguments: the name of a host, and the pathname for the mount point you have chosen. It calls the program `smount` (also in the installed `bin` directory) with a string of arguments, the last two of which are constructed from the host name and the mount point.

   You can use the script `mnt` as it is, or you can edit it if you want to use different arguments to `smount`. You can also just use `smount` directly. See the manual page *smount(SHORE)* for assistance.
5. Check the mount. You don't have to be super-user to do this. Type `mount` or `smount` and see if your mount point shows up in the mount table.

4.1 Instructions for Linux Users

The mount program that comes with Linux already has the features that Shore needs and therefore there is no need for the `smount` program. In the above instructions, replace `smount` with `mount` and `mnt` with `mnt.linux`. See your Linux `mount` manual page for more information.

References

[SGK+85] R. Sandberg, D. Goldberg, S. Kleiman, D.Walsh, and B.Lyon. Design and implementation of the sun network filesystem. In *USENIX Summer Conference Proceedings*, 1985.