The Shore Release 1.1.1[1]

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

Version 1.1.1

*Copyright ©1994–7
Computer Sciences Department, University of Wisconsin—Madison.
All Rights Reserved.*

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 What is the Shore Release? 1

1.1 Requirements 1

1.2 Copyright and Disclaimer 1

1.3 What is New in Release 1.1.1 2

1.4 What was New in Release 1.1 2

1.4.1 Maintenance Fixes 2

1.4.2 Configuration 3

1.5 What was New in Release 1.0 3

1.5.1 General 3

1.5.2 Applications 3

1.5.3 Object cache 3

1.5.4 Shore server 3

1.5.5 Storage manager 4

2 Release Contents 4

2.1 Documentation Release 4

2.2 Binary Release 6

2.3 Source Release 6

3 Getting Started 6

3.1 Installing the Shore Software 6

3.2 Learning More about the Shore Server 6

3.3 Writing, Compiling, and Running SDL Applications 7

3.4 Building Value-Added Servers using the Storage Manager 7

1 What is the Shore Release?

Release 1.1.1 of the Shore software represents a bug-fix and maintenance release relative to the previous release. The release is targeted for sites that will use Shore as a system for writing applications using the Shore Data-definition Language and for those who want to write their own servers using the Shore Storage Manager. This document describes the release and provides pointers to other documents explaining how to install and use the software.

If you have questions, comments or bug reports, please mail them to

```
shore_support@cs.wisc.edu
```

For information on other mailing lists, see the Shore WWW home page: http://www.cs.wisc.edu/shore/.

1.1 Requirements

The Version 1.1.1 release of Shore is supported for Solaris 2.5 on SPARCstations and Intel X86 (Pentium) workstations and Linux. Shore has been ported to other platforms, but there are no supported ports at this time. We hope to make available, on an "as is" basis, ports contributed by the user community.

You will need a current version of GNU C/C++ compiler *gcc* (with its include files) and the GNU *make* utility. We do not support other compilers or versions of *make*. The current release builds successfully with versions 2.7.2.2 and 2.7.2.3 of the compiler. The pre-compiled (binary) release was built with gcc 2.7.2.3 and libg++ 2.7.2. As newer versions of the compiler are released, we will update the Shore software.

Complete details about required software and hardware are available in the Requirements section of the Installation manual.

1.2 Copyright and Disclaimer

The Shore software is distributed with the following copyright notice and disclaimer, and is subject to its terms:

Copyright ©1994–7
Computer Sciences Department, University of Wisconsin—Madison.
All Rights Reserved.

Permission to use, copy, modify and distribute this software and its documentation is hereby granted, provided that both the copyright notice and this permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof, and that both notices appear in supporting documentation.

THE COMPUTER SCIENCES DEPARTMENT OF THE UNIVERSITY OF WISCONSIN – MADISON ALLOWS FREE USE OF THIS SOFTWARE IN ITS "AS IS" CONDITION. THE DEPARTMENT DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

The Shore Project Group requests users of this software to return any improvements or extensions that they make to the Shore Project Group, Computer Sciences Department, University of Wisconsin, 1210 West Dayton Street, Madison, WI 53706-1685.

In addition, the Shore Project Group requests that users grant the Computer Sciences Department rights to redistribute these changes.

1.3 What is New in Release 1.1.1

The Linux port has been brought up-to-date. Changes were small but rather pervasive. Recent releases have become more and more Solaris-specific. These updates should make Shore easier to port to other platforms.

Numerous other small bugs, mostly concerned with compiling and installing the software, have been repaired.

A rather obscure bug concerning SIX mode locking has been fixed in the lock manager (src/sm/lock.cc).

The binary release was built with the latest GNU compilers: gcc (and g++) version 2.7.2.3 and libg++ 2.7.2.

1.4 What was New in Release 1.1

1.4.1 Maintenance Fixes

Numerous bug fixes and performance improvements have been made throughout Shore, especially in the storage manager. In particular,

- Locks are automatically escalated from record to page to file level.
- Logging has been substantially improved to generate fewer log records.
- Bulk loading facilities have been improved.
- Allocation of space in files has been substantially improved.
- Numerous storage leaks and uninitialized variables discovered by Purify^{TM} have been fixed.
- More kinds of temporary files are available. The semantics have been cleaned up and bugs in logging have been fixed.
- The concurrency control and recovery code for btrees has been extensively revised more closely to follow Mohan's ARIES algorithms.
- Keys of all primitive types are now correctly handled (in a machine-independent manner) for btrees and for sorting.
- The `#include` files in the storage manager have been reorganized to reflect a cleaner layering and to improve compilation speed.
- Error messages from the `diskrw` process have been made more intelligible.
- A great deal of obsolete code has been removed.
- Most of the documents have been substantially rewritten.

1.4.2 Configuration

The system for configuring and building Shore from source has been substantially streamlined. We no longer use `imake` (from the X Window System) to generate Makefiles. Instead, Makefiles are generated from Imakefiles and configuration files in the `config` directory by a Perl script. In addition to eliminating the need to install `imake`, this change makes configuration considerably easier to understand and modify. Imakefiles and substantially shorter, and the generated Makefiles are both *much* shorter (up to 5 times shorter) and formatted more readably. All local configuration settings related to compiling are confined to a single file: `config/config.h`, and mostly to the first 160 or so lines. The only runtime configuration necessary is a one-line change to a single file (`lib/options`, which is normally installed as `~/.shoreconfig`).

1.5 What was New in Release 1.0

1.5.1 General

- Ported to Solaris 2.5.
- The software builds under gcc 2.7.2 with the libg++ 2.7.1 library and include files.
- Numerous bugs were fixed.

1.5.2 Applications

- Applications linked with the no-debug libraries must be recompiled with the no-debug include files, because the two implementations of the return-code class are incompatible.
- Ref¡Pool¿ can be used in applications.
- Enumerations can be used for index key types.
- Reference-counting on modules is enabled and under the control of the option `oc_refcount`. If you use this, use it with caution.

1.5.3 Object cache

No changes.

1.5.4 Shore server

See *Storage manager*.

1.5.5 Storage manager

- The format of logs and volumes has changed; data created under previous releases cannot be read by a Version 1.0 server.
- The SM programming interface is slightly changed to support multi-threaded transactions (in limited circumstances: see *smthread_t(SSM)*),

  external two-phase commit (see *transaction(SSM)*), and faster record creation in a file in append-only context (see *scan_file_i(SSM)*).
- The log can be a raw device. The SM can be configured to use a separate (diskrw) process for the log or to perform the log I/O locally (and synchronously). The SM layer has a new option `sm_reformat_log` to control reformatting the log, in the event that a raw device is used. The meaning of the option `sm_logsize` has changed. (See *options(SVAS)* for details.)
- More statistics are reported.
- Miscellaneous performance improvements were made (logging uninitialized data, append-only file population, store-descriptor cache expands as needed, lock manager simplified).
- The programming interface for threads has changed. The changes are incorporated into the example value-added server, and described in the companion document.

2 Release Contents

The Shore Software distribution consists of documentation, binary, and source releases. These are briefly described below. Additional information is available in the *installation manual*.

2.1 Documentation Release

The documentation release consists of HTML and Postscript versions (in `docs.html/` and `docs.ps/`, respectively) of about 15 documents and over 150 manual pages. It also contains, in the `examples` directory, source code for several example programs.

- *The Shore Release 1.1.1* (this document)
- *Shore Software Installation Manual*
- *How to Submit a Problem Report*
- *An Overview of Shore*
- *Glossary of Shore Terms*
- Documents for Users of SDL and the Shore Server
  - *A list of all the manual pages*
  - *Configuring and Running the Shore Server*
    and related manual pages:
    - *options(SVAS)*
    - *environment(SVAS)*
    - *sshutdown(SVAS)*
    - *mount(SVAS)*
    - *smount(SHORE)*
  - A tutorial on writing Shore applications using the Shore Data Language (SDL) *Getting Started with Shore*
    and miscellaneous helpful documents:
    - *Scanning Directories*
    - *Scanning Pools*
    - *Handling Errors*
  - *Shore Data Language Reference Manual*
    and related manual pages:
    - *Manual pages* for using the SDL compiler and the SDL C++ language binding,
    - *Manual pages* for the Shore Object Cache and Unix-compatibility classes
- Documents for Value-Added Server Writers (using the Storage Manager)
  - *A Roadmap to the Shore Releases*
  - *The Shore Storage Manager Programming Interface*
  - Manual Pages for the Storage Manager
    - *Storage Manager proper*
    - *Foundation classes*
    - *Thread package*

        - *Common utility classes*
    - *Writing Value-Added Servers with the Shore Storage Manager* a tutorial on using the Storage Manager

The example programs in the examples directory include

stree examples to go with the tutorial *Getting Started with Shore*,

pscan an example to go with the tutorial *Scanning Pools*,

shls an example to go with the tutorial *Scanning Directories*,

unixfile some examples that use the type `SdlUnixFile`, which is a registered object with one attribute, of type `text` (the object looks like a Unix file),

vas/hello a trivial “value-added server,”

vas/grid a much more extensive value-added server, described in the tutorial *Writing Value-Added Servers with the Shore Storage Manager*, and

oo7 an implementation of the OO7 Benchmark.

2.2 Binary Release

A binary release consists of the SDL compiler, a run-time library to be linked with applications, the Shore Server executable, and utilities for mounting the Shore File System as an NFS file system. It also includes the libraries and header files for the Shore Storage Manager. The binary release is available with and without internal debugging support. The differences between the two are as follows:

- Debugging release:
    - Compiled with debugging symbols (`-g`)
    - Compiled with extra auditing/debugging (`-DDEBUG`)
    - Compiled with simple optimization (`-O`)
    - Executables are considerably larger and slower than their no-debugging counterparts.
- No-Debugging release:
    - Compiled without debugging symbols
    - Compiled without extra auditing/debugging (`-UDEBUG`)
    - Compiled with more optimization (`-O2`)

2.3 Source Release

The source release contains the source tree and tools for building everything in the binary release.

3 Getting Started

The following sections explain how to get started using Shore to write applications.

3.1 Installing the Shore Software

The first step in using Shore is to install the documentation and software. The *Shore Software Installation Manual* explains how to install the documentation and binary releases and how to use the source release to compile new binaries.

The installation manual contains a section on testing the installation. This is a step-by-step guide to starting a Shore Server, compiling and running an application and mounting the Shore file system.

3.2 Learning More about the Shore Server

After you have installed the Shore software and tested your installation, you can find more information about using the Shore Server and file system in the document *Configuring and Running a Shore Server.*

3.3 Writing, Compiling, and Running SDL Applications

Several example programs are included in documentation release. The document, *Getting Started with Shore* uses these examples to explain how to write, compile and run applications.

To write your own Shore application, you will need to be familiar with the Shore Data-definition language, SDL. See the document *An SDL Reference Manual*, and the man pages sdl(SDL) and sdlcxx(SDL) for more information.

3.4 Building Value-Added Servers using the Storage Manager

This release includes support for using the Shore Storage Manager to build your own servers. The Shore Server is an example of such a server. The tutorial *Writing Value-Added Servers with the Shore Storage Manager* uses a simple example to illustrate the creation of a customized server. For details on the Storage Manager programming interface, see *The Shore Storage Manager Programming Interface.*