


GemStone/S 64 BitTM **Installation Guide**

For Red Hat-compatible Linux on
x86_64, using RPM

Version 3.7

September 2023



INTELLECTUAL PROPERTY OWNERSHIP

This documentation is furnished for informational use only and is subject to change without notice. GemTalk Systems LLC assumes no responsibility or liability for any errors or inaccuracies that may appear in this documentation.

Warning: This computer program and its documentation are protected by copyright law and international treaties. Any unauthorized copying or distribution of this program, its documentation, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted under the maximum extent possible under the law.

The software installed in accordance with this documentation is copyrighted and licensed by GemTalk Systems under separate license agreement. This software may only be used pursuant to the terms and conditions of such license agreement. Any other use may be a violation of law.

Use, duplication, or disclosure by the Government is subject to restrictions set forth in the Commercial Software - Restricted Rights clause at 52.227-19 of the Federal Acquisitions Regulations (48 CFR 52.227-19) except that the government agency shall not have the right to disclose this software to support service contractors or their subcontractors without the prior written consent of GemTalk Systems.

This software is provided by GemTalk Systems LLC and contributors “as is” and any expressed or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall GemTalk Systems LLC or any contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

COPYRIGHTS

This software product, its documentation, and its user interface © 1986-2023 GemTalk Systems LLC. All rights reserved by GemTalk Systems.

PATENTS

GemStone software has been covered by U.S. Patent Number 6,256,637 “Transactional virtual machine architecture”, Patent Number 6,360,219 “Object queues with concurrent updating”, Patent Number 6,567,905 “Generational garbage collector with persistent object cache”, and Patent Number 6,681,226 “Selective pessimistic locking for a concurrently updateable database”. GemStone software may also be covered by one or more pending United States patent applications.

TRADEMARKS

GemTalk, **GemStone**, **GemBuilder**, **GemConnect**, and the GemStone and GemTalk logos are trademarks or registered trademarks of GemTalk Systems LLC, or of VMware, Inc., previously of GemStone Systems, Inc.

UNIX is a registered trademark of The Open Group.

Intel is a registered trademarks of Intel Corporation.

Microsoft, **Windows**, **Windows Server**, and **Azure** are registered trademarks of Microsoft Corporation.

Linux is a registered trademark of Linus Torvalds and others.

Red Hat, **Red Hat Enterprise Linux**, **RHEL**, and **CentOS** are trademarks or registered trademarks of Red Hat, Inc.

AlmaLinux is a trademark or registered trademark of AlmaLinux OS Foundation.

Rocky Linux is a trademark or registered trademark of Rocky Enterprise Software Foundation.

Ubuntu is a registered trademark of Canonical Ltd., Inc.

AIX, **Power**, **POWER**, **Power8**, **Power9**, and **VisualAge** are trademarks or registered trademarks of International Business Machines Corporation.

Apple, **Mac**, **macOS**, and **Macintosh** are trademarks of Apple Inc.

Instantiations is a registered trademarks of Instantiations, Inc.

CINCOM, **Cincom Smalltalk**, and **VisualWorks** are trademarks or registered trademarks of Cincom Systems, Inc.

Raspberry Pi is a trademark of the Raspberry Pi Foundation.

RabbitMQ is a trademark of VMware, Inc.

Prometheus is a registered trademark of The Linux Foundation.

Grafana is a registered trademark of Raintank, Inc. dba Grafana Labs.

Other company or product names mentioned herein may be trademarks or registered trademarks of their respective owners. Trademark specifications are subject to change without notice. GemTalk Systems cannot attest to the accuracy of all trademark information. Use of a term in this documentation should not be regarded as affecting the validity of any trademark or service mark.

GemTalk Systems

15220 NW Greenbrier Parkway
Suite 240
Beaverton, OR 97006

Preface

About This Documentation

This document explains how to install GemStone/S 64 Bit version 3.7 on a workstation running Red-Hat compatible Linux on x86_64, using RPM installation; and how to upgrade from previous GemStone/S 64 Bit versions.

If you are not installing GemStone on a Red-Hat compatible workstation using Red Hat Package Manager (RPM), see the separate *Installation Guide for Linux*.

Terminology Conventions

The term “GemStone” is used to refer to the server products GemStone/S 64 Bit and GemStone/S, and the GemStone family of products; the GemStone Smalltalk programming language; and may also be used to refer to the company.

Technical Support

Support Website

gemtalksystems.com

GemTalk’s website provides a variety of resources to help you use GemTalk products:

- ▶ **Documentation** for the current and for previous released versions of all GemTalk products, in PDF and HTML.
- ▶ **Product download** for the current and selected recent versions of GemTalk software.
- ▶ **Bugnotes**, identifying performance issues or error conditions that you may encounter when using a GemTalk product.
- ▶ **Supplemental Documentation** and **TechTips**, providing information and instructions that are not in the regular documentation.

- ▶ **Compatibility matrices**, listing supported platforms for GemTalk product versions.

We recommend checking this site on a regular basis for the latest updates.

Help Requests

GemTalk Technical Support is limited to customers with current support contracts. Requests for technical assistance may be submitted online (including by email), or by telephone. We recommend you use telephone contact only for urgent requests that require immediate evaluation, such as a production system down. The support website is the preferred way to contact Technical Support.

Website: techsupport.gemtalksystems.com

Email: techsupport@gemtalksystems.com

Telephone: (800) 243-4772 or (503) 766-4702

Please include the following, in addition to a description of the issue:

- ▶ The versions of GemStone/S 64 Bit and of all related GemTalk products, and of any other related products, such as client Smalltalk products, and the operating system and version you are using.
- ▶ Exact error message received, if any, including log files and statmonitor data if appropriate.

Technical Support is available from 8am to 5pm Pacific Time, Monday through Friday, excluding GemTalk holidays.

24x7 Emergency Technical Support

GemTalk offers, at an additional charge, 24x7 emergency technical support. This support entitles customers to contact us 24 hours a day, 7 days a week, 365 days a year, for issues impacting a production system. For more details, contact GemTalk Support Renewals.

Training and Consulting

GemTalk Professional Services provide consulting to help you succeed with GemStone products. Training for GemStone/S is available at your location, and training courses are offered periodically at our offices in Beaverton, Oregon. Contact GemTalk Professional Services for more details or to obtain consulting services.



Table of Contents

Chapter 1. Installing GemStone/S 64 Bit Version 3.7 using RPM

Review the Installation Procedure	7
Check the System Requirements	8
Configure the Operating System	9
Install the GemStone Server.	17
Install the GemStone Product	17
Set the GemStone Key File.	17
Configure the NetLDI	18
Set the Environment	19
Create a GemStone repository	19
Setup Extents and Configuration	19
Complete GemStone Configuration	20
Further Configuration and Administration	21

Chapter 2. Upgrading from GemStone/S 64 Bit 3.4.x or later

Keyfiles	23
Upgrade Procedure	24
Prior to Upgrade in existing application.	24
Prepare for Upgrade	24
Perform the Upgrade.	26
GsDevKit Upgrade.	28
Post-upgrade Application Code Modifications	28
Make Backup	28
Configure GCI clients and GBS.	28

Chapter 3. Upgrading GLASS/GsDevKit Applications

Upgrade Procedure	29
1. Ensure that GemStone 3.7 is installed and your repository upgraded	30
2. If necessary, customize the upgrade instructions	30
3. Perform the Upgrade.	31
4. Load your Application Code	32
5. Complete the Upgrade Process	32

Chapter 4. Configuring GBS for GemStone/S 64 Bit

Supported GBS client Smalltalk Platforms with Linux	34
GBS Setup or Upgrade Procedure.	34
Upgrade GBS to a version that supports v3.7	35
Update GBS to reference v3.7 libraries	35

Installing GemStone/S 64 Bit Version 3.7 using RPM

This chapter describes the procedure for installing GemStone/S 64 Bit™ version 3.7, using the RPM (Red Hat Package Manager) installation process.

For new users, we recommend that you set up GemStone initially on a single machine, to ensure that all the pieces work together. Further setup to run a distributed system are described in the *System Administration Guide*. You will need to adjust the installation process to meet your specific needs.

If you are upgrading to this release from a previous version of GemStone/S 64 Bit, follow the instructions in the appropriate later chapter of this Installation Guide.

When upgrading, note that if your existing application relies on files in `$GEMSTONE/data/`, including the `system.conf` file, or use `$GEMSTONE/ua1ib` for User Action libraries, you must make changes in your configuration for the RPM environment. In an RPM installation, GemStone repositories should not write to the `$GEMSTONE` directory tree. You will need to define a new directory, move any files to this location, and replace references in configuration files and application code.

Installation using RPM is a recent capability for GemStone software. The specific installation details and recommendations in this document may not be final. Contact GemTalk Technical Support if you have issues or questions.

Review the Installation Procedure

The following list summarizes the steps to install GemStone/S 64 Bit.

- ▶ Check the System Requirements. 8
- ▶ Install the GemStone Server 17
- ▶ Set the Environment 19
- ▶ Setup Extents and Configuration 19
- ▶ Complete GemStone Configuration. 20
- ▶ Further Configuration and Administration 21

Check the System Requirements

Before you install GemStone/S 64 Bit, ensure that the following system requirements are satisfied. Additional system resources may be necessary to support large applications.

Platform

- ▶ Intel x86_64 CPU with architecture Sandy Bridge or newer (excluding Atom CPUs)
- ▶ AMD x86_64 CPUs with architecture Bulldozer version 1 or newer

GemStone/S 64 Bit v3.7 will not run on x86_64 CPUs older than listed. GemStone/S 64 Bit v3.7 was built using compilation features that are not available on older CPUs; v3.7 cannot run on these CPUs, regardless of the Linux OS version.

RAM and Swap space

- ▶ While small installations can run on systems with only a few GB of physical RAM, increasing RAM is important for GemStone performance.

Total swap space should be at least equal to the amount of RAM. Due to the way GemStone uses memory, systems with insufficient swap space allocated have a risk of memory errors even if there is available RAM.

Disk space

- ▶ Space for the installed distribution files – you need approximately for GemStone/S 64 Bit, and additional space for other products.
- ▶ Additional disk space as required for your repository.
- ▶ ext4, XFS, and ZFS are supported; XFS or ZFS filesystem should be used in preference to ext4. Extent files cannot be accessed via CIFS, and NFS requires special configuration.
- ▶ The repository files should be located on a disk drive that does not contain swap space. Use of multiple disk drives is advisable for servers.

Operating system

- ▶ Red Hat-compatible Linux ES 9.2 on **x86_64**
kernel version 5.14.0-284.11.1.el9_2.x86_64 and glibc-2.34-60.el9.x86_64
- ▶ Red Hat-compatible Linux ES 8.7 on **x86_64**
kernel version 4.18.0-425.3.1.el8.x86_64 and glibc-2.28-211.el8.x86_64
- ▶ Red Hat-compatible Linux ES 7.9 on **x86_64**
kernel version 3.10.0-1160.71.1.el7.x86_64 and glibc-2.17-326.el7_9.x86_64

GemStone performs testing on a mixture of Red Hat, CentOS, Rocky and Alma servers; these are all considered fully certified platforms. Any reference to Red Hat applies to any Red Hat-compatible distribution.

Note that you must also have a supported CPU; GemStone/S 64 Bit v3.7 will not run on CPUs older than listed under “Platform”.

Debugger

A C debugger allows C-level stack traces when a GemStone error occurs, or when using the `pstack` command. While not required for GemStone execution, it is strongly recommended that a debugger be installed, so diagnostic error stacks will be available.

- ▶ Red Hat-compatible Linux ES 9.2: gdb 10.2
- ▶ Red Hat-compatible Linux ES 8.7: gdb 9.1
- ▶ Red Hat-compatible Linux ES 7.9: gdb 9.1

C/C++ Compiler

GemStone requires a C/C++ compiler only if you are developing C or C++ code for user actions or for a C or C++ application; as described in the *GemBuilder for C* manual. This compiler is required only for development work, not for execution.

- ▶ Red Hat-compatible Linux ES 9.2: gcc/g++ 11.3.1
- ▶ Red Hat-compatible Linux ES 8.7: gcc/g++ 8.5
- ▶ Red Hat-compatible Linux ES 7.9: gcc/g++ 4.8.5

X Windows

An X Windows server allows you to use GemStone's graphical VSD application on Linux. Alternatively, GemStone statistical data may be viewed on Windows, by transferring the data files, or by mounting the file system on Windows. X Windows is not required for GemStone execution.

Configure the Operating System

Modern Linux installations default to settings that support most GemStone configurations; there is no longer a need to set `kernel.shmall`, `kernel.shmmax`, `kernel.sem`, or `fs.file-max`. The default limits on modern Linux kernels are sufficiently large even for very large GemStone configurations.

1. Locking the Shared Page Cache in memory

It is recommended that you have sufficient memory to avoid the need to swap to disk. If there is the risk of swapping, you can avoid problems with GemStone performance (at the expense of other processes running on this machine) by locking the shared page cache into memory. Locking the cache into memory is configured by the parameter `SHR_PAGE_CACHE_LOCKED`.

If you are using huge pages (see page 15), the cache is inherently locked, so there is no further action needed.

Locking the cache into memory requires that the user starting GemStone has permission to lock the amount of memory required by the cache. There are several ways to do this.

- ❑ Edit `/etc/security/limits.conf` to give the administrative user permission to lock larger memory segments:

```
gsAdminUser soft memlock sizeInKB
gsAdminUser hard memlock sizeInKB
```

Note that after editing this file, you do not need to reboot, but you will need a new shell.

- ❑ While not recommended for security reasons, you may give the Shared Page Cache Monitor executable the Linux capability `CAP_IPC_LOCK`.

```
os$os$ setcap cap_ipc_lock=pe $GEMSTONE/sys/shrpcmonitor
```

Note that setting this capability will prevent `statmonitor` and `gdb` from attaching to `shrpcmonitor`; `statmonitor` will not be able to record certain process statistics, and `pstack` cannot get C stack traces. See “`statmonitor` additional permission” on page 16.

2. OOM Killer

If your system runs low on memory, the Linux OOM killer may select GemStone processes to terminate. To protect the shared page cache and other critical GemStone processes, the process’s `oom_score_adj`, which is used to select processes to terminate, is reduced, making them less likely to be selected and killed.

Reducing the `oom_score_adj` requires configuring the process to have that capability. To set the capability on the required executables:

```
os$ setcap cap_sys_resource=pe $GEMSTONE/sys/stoned
os$ setcap cap_sys_resource=pe $GEMSTONE/sys/pgsvrmain
```

Only the `Stone` and `pgsvrmain` executables need the capability; other critical processes spawned by the `Stone`, such as the Shared Page Cache Monitor, inherit the `Stone`’s protection and adjust their `oom_score_adj`, and therefore also are protected. The capability is dropped as soon as the `oom_score_adj` has been updated.

Likewise, the `pgsvrmain`, which is responsible for spawning critical processes for distributed configurations with remote caches, protects processes such as the remote cache.

To examine the `oom_score_adj` for a process, check the file:

```
os$ cat /proc/pid/oom_score_adj
```

3. Transparent Huge Memory Pages

The default size for memory pages on Linux/x86 is 4KB. Linux/x86 supports 2MB and 1GB Huge Pages; this document, and GemStone code, refers to these page sizes and all larger-than-default memory pages using the generic term large pages. Using large pages will improve performance for large repositories with large shared page caches.

The Linux kernel allows you to explicitly reserve 2MB or 1GB large pages for a specific application (such as GemStone), and also allows you to take advantage of 2MB transparent large pages, which are managed automatically without application level configuration.

Specifically allocated large memory pages are described in the next section, “Consult your System Administrators for more information on how authentication is handled on your system.” on page 15.

Transparent huge pages are limited to anonymous memory regions. If they are enabled (they are enabled by default), transparent huge pages are managed by the kernel without application level configuration.

In general we recommend settings of "advise" or "always", which allow GemStone to use transparent huge pages. Settings of "always" will allow GemStone to use transparent huge pages, but may affect overall performance either positively or negatively.

Linux systems should ensure:

- ▶ `/sys/kernel/mm/transparent_hugepage/enabled` should be set to "advise".
- ▶ `/sys/kernel/mm/transparent_hugepage/defrag` should be set to "advise".
- ▶ `/sys/kernel/mm/transparent_hugepage/shmem_enabled` (on all except some older OS versions) should be set to "advise".

To confirm these settings are correct in your Linux environment, **cat** the file; the enabled parameter is within the square brackets. E.g.,

```
os$ cat /sys/kernel/mm/transparent_hugepage/enabled
always [advise] never
```

To confirm that GemStone is using transparent huge pages, after GemStone has been started up, execute:

```
os$ cat /proc/pid/smaps
```

and look for the /SYSV line, and check that the subsequent value for ShmemPmdMapped is nonzero.

4. Huge Memory Pages

Linux supports explicit configuration for 2MB and 1GB large pages that are specifically reserved for an application, as well as transparent large pages (as described in the previous section).

A Linux installation may be configured to use either 2MB and 1GB large pages, or both. Using large pages may improve performance for large repositories with large shared page caches; by explicitly configuring large pages, they are allocated on boot and the GemStone executable can be explicitly configured to use them.

To use explicitly configured large pages, you must determine the page size and the number of pages needed, configure Linux to allocate the required number of pages, and configure GemStone to use these pages.

To configure the use of large pages:

a. Determine the Linux Page Size

From the page sizes that are available on your Linux system, you must decide on the size you intend to use. Smaller repositories may use 2MB pages, but

repositories with multi-GB shared page cache sizes may benefit more from 1GB pages.

To determine the default huge page size, execute:

```
os$ grep -e '^Huge' /proc/meminfo
```

which reports, for example,

```
HugePages_Total:    5000
HugePages_Free:     2756
HugePages_Rsvd:     0
HugePages_Surp:     0
Hugepagesize:       2048 kB
```

To determine if you have another page size available, execute:

```
os$ ls /sys/kernel/mm/hugepages/
```

If you have 2MB pages, it will return:

```
hugepages-2048kB
```

If you have both 2MB and 1GB pages, it will report:

```
hugepages-1048576kB hugepages-2048kB
```

b. Determine the required number of Linux Pages

Calculate the number of huge pages that will be needed, based on your GemStone configuration. This is calculated by the utility **largememorypages**. This utility needs several details to compute the number of required pages: the shared page cache size, the maximum number of GemStone processes, and the number of shared counters. These can be provided by arguments or read from an existing configuration file. When using a configuration file, the file must have large memory pages (SHR_PAGE_CACHE_LARGE_MEMORY_PAGE_POLICY) enabled, and page size (SHR_PAGE_CACHE_LARGE_MEMORY_PAGE_SIZE_MB) set, if the calculated default page size is not correct.

The **largememorypages** computation by default is for the Stone's cache. When configuring large pages for remote cache, the values are somewhat smaller. Use the **-r** option to compute large page requirements for a remote cache.

You may specify the required values using an existing configuration file, or by passing in the relevant values as argument. A configuration file is parsed as a Stone configuration file and the necessary values extracted; the configuration file must have large memory pages enabled.

If a configuration file is not specified, then **-M** or **-F**, and **-p**, **-P** and **-C** are required. These arguments can be used in addition to the configuration file, in which case they override values set or computed in the configuration file:

```
largememorypages [-e path ][-z path ][-r ][-F cacheFrames | -M cacheKB ]
                 [-P maxProcesses ][-C maxSharedCounters ][-p largeMemoryPageSize ]
```

-e path

specifies an executable configuration file (same as startstone -e).

-z path

specifies a system configuration file (same as startstone -z).

-F *cacheFrames*
 shared cache size expressed in 16 KB frames

-M *cacheKB*
 shared cache size with a optional units suffix.

-P *maxProcesses*
 setting for SHR_PAGE_CACHE_NUM_PROCS

-C *maxSharedCounters*
 setting for SHR_PAGE_CACHE_NUM_SHARED_COUNTERS.

-p *largeMemoryPageSize*
 setting for SHR_PAGE_CACHE_LARGE_MEMORY_PAGE_SIZE_MB.
 Options are 2MB and 1GB (or equivalent using appropriate units).

The following example is for an (approximately) 20GB shared page cache, 200 processes, and 1900 shared counters, with 1 GB memory pages:

```
os$ largememorypages -p 1GB -M 20GB -C 1900 -P 200

Cache kind is: Primary SPC (stone shared page cache)

Cache config is 1373248 pages = 21457 MB, total is 22528
MB, overhead 4% of configured size
  HashTableSize 2097152 TotalPces 3483008 freeListPces
1385856
Large page size requested is: 1024 MB.
Large page overhead: 0.09 MB
[Info]: Increasing number of pages from 1310720 to 1373248
to reduce large memory page overhead.
For 1373248 pages, 200 processes and 1900 shared counters, 0
pusherThreads
  minimum sizing for cache shmmax 23622320128, shmall
5767168.
Number of 1024 MB large pages required: 22
<with further update commands>
```

c. Allow large memory pages to be allocated and accessed by GemStone processes

To ensure processes have permission to use large memory pages, you need to do one of the following. You must execute these commands as root.

- Define a huge pages group.

Select or create a group that includes the Linux user that will be starting up the shared page cache, and all users who will be accessing the shared page cache

Then create a file `/etc/sysctl.d/64-gemstone-local.conf`, to set this group's numeric id; add the line:

```
vm.hugetlb_shm_group = numericGidOfGroup
```

To determine the group id based on the group name, execute

```
os$ getent group nameOfGroup
```

This will take effect on reboot; you can apply immediately by executing:

```
os$ sysctl --system
```

To verify the current value, cat `/proc/sys/vm/hugetlb_shm_group`, which will show the group id. To enable large pages transiently without

rebooting, you can set the gid of the group directly in `/proc/sys/vm/hugetlb_shm_group`. This will be reset on reboot if not configured as by `/etc/sysctl.d/64-gemstone-local.conf`.

- ❑ While not recommended for security reasons, you may instead give the Shared Page Cache Monitor the `cap_ipc_lock` capability:

```
os$ setcap cap_ipc_lock=pe $GEMSTONE/sys/shrpcmonitor
```

Setting this capability will prevent statmonitor and gdb from attaching to that process; statmonitor will not be able to record certain process statistics, and pstack cannot get C stack traces. See “statmonitor additional permission” on page 16.

d. Configure large pages on Linux

Linux must be configured according to the instructions for your Linux distribution. You must execute these commands as root.

(1) To enable hugepages, the file `/etc/default/grub` needs to contain an entry for `GRUB_CMDLINE_LINUX_DEFAULT` that defines the number of 2MB and 1 GB huge pages.

Edit the file `/etc/default/grub`, to append a line defining the number of each size of pages to the `GRUB_CMDLINE_LINUX_DEFAULT` entry, for example to configure with 1GB pages to support the 20GB cache in the above example:

```
"hugepagesz=1G hugepages=22 hugepagesz=2M hugepages=0
default_hugepagesz=1G"
```

(2) Run the command to update the kernel config.

On Ubuntu:

```
os$ /usr/sbin/update-grub
```

On Red Hat and CentOS:

```
os$ /usr/sbin/grub2-mkconfig -o /boot/grub2/grub.cfg
```

On Red Hat and CentOS UEFI systems:

```
os$ /usr/sbin/grub2-mkconfig -o /boot/efi/EFI/centos/grub
.cfg
```

(3) Reboot the system:

```
os$ shutdown -r now
```

e. Configure GemStone to use large pages

To configure GemStone to request large pages, set the configuration option `SHR_PAGE_CACHE_LARGE_MEMORY_PAGE_POLICY`. This can be set to 1 or 2; with a setting of 1, the cache will be started anyway if the request for large pages is denied, while a setting of 2 indicates that startup should fail if large pages cannot be allocated.

Setting `SHR_PAGE_CACHE_LARGE_MEMORY_PAGE_SIZE_MBto2`(for 2MB) or 1024 (for 1GB) is recommended; otherwise, the default is based on the current configuration of 1 GB, 2 MB, and/or the Linux default huge page size.

5. PAM

If you are using UNIX authentication for GemStone logins, or if you run NetLDI as root with `setuid` (i.e. not in guest mode), you must have PAM (Pluggable Authentication Module) configured on the server. You may include a specific GemStone authorization service name, or allow the default “other” authentication definitions to be used.

PAM authentication definitions are in files under the directory `/etc/pam.d`. Alternatively, they can be lines in the configuration file `/etc/pam.conf`, but this usage is deprecated on many distributions. On these distributions, the presence of the `/etc/pam.d` directory will cause `/etc/pam.conf` to be ignored.

The specific GemStone service file names are `gemstone.gem` for user authentication, and `gemstone.netldi` for a NetLDI running with authentication.

The libraries that are specified in the stack depend on how you are configuring PAM to perform the authentication. The examples below are for PAM configured to invoke LDAP for authentication.

For GemStone UNIX authentication, which uses PAM, to authenticate via LDAP, create a file named `/etc/pam.d/gemstone.gem` with the following contents:

```
auth                required                pam_ldap.so
```

For NetLDI authentication, again using LDAP, create a file named `/etc/pam.d/gemstone.netldi` with the following contents:

```
auth                required                pam_ldap.so
```

Red Hat, by default, installs a file `/etc/pam.d/other` which disables “other” authentication. On Ubuntu, it is enabled by default. You can allow the “other” authentication stack to be used for GemStone authentication by ensuring that the file `/etc/pam.d/other` has the following contents:

```
auth                required                pam_ldap.so
```

Consult your System Administrators for more information on how authentication is handled on your system.

6. System clock

The system clock must be set to the correct time. When GemStone opens the repository at startup, it compares the current system time with the recorded checkpoint times; a system time earlier than the last checkpoint time may be taken as an indication of corrupted data and prevent GemStone from starting. The time comparisons use GMT.

7. TCP keepalive option

GemStone processes ordinarily use the `TCP keepalive` option to determine how long they will wait after communications activity ceases unexpectedly. This setting can be useful for reaping stale RPC Gems, but the OS default may not be appropriate for your system. For further information, refer to your operating system documentation.

8. `/etc/systemd/logind.conf` must disable `RemoveIPC`

To avoid Stone shutdown when the session that started the Stone logs out, ensure that

```
/etc/systemd/logind.conf
```

includes the line

```
RemoveIPC=no
```

With the default value of yes, when the session that started the Stone logs out, the Stone's shared semaphore array is deleted, causing fatal errors.

9. statmonitor additional permission

If you are running with the NetLDI owned as root with the s bit set, then **statmonitor** running as an ordinary user, including as the administrative user, will not be able to collect certain memory statistics for Gems started by other users.

- ❑ You can enable statmonitor to still collect these statistics by giving it `cap_sys_ptrace`.

```
os$ setcap cap_sys_ptrace=pe $GEMSTONE/bin/statmonitor
```

- ❑ Alternatively, statmonitor can be run as root with s bit set:

```
os$ cd $GEMSTONE/sys
os$ chown root $GEMSTONE/bin/statmonitor
os$ chmod u+s $GEMSTONE/bin/statmonitor
```

If you have set the Linux capacity `cap_ipc_lock` for `shrpcmonitor`, for huge pages or locking (this is permitted but not recommended), statmonitor will not be able to collect memory statistics for the shared page cache monitor, and the above workarounds may be applied. The `cap_sys_resource` capability, for OOM protection, is dropped after the process has adjusted its `oom_score_adj`, and is not an issue.

10. Unset LD_BIND_NOW

On some Linux distributions, setting the environment variable `LD_BIND_NOW` may result in process startup failures due to loading incorrect shared libraries.

Install the GemStone Server

Installing GemStone using RPM installs under `/usr/lib64`, as root user. This installation differs from the historic installation of GemStone from zipped distributions.

The defaults that GemStone provides for configuration and utilities, and many examples provided in documentation, assume a writable `$GEMSTONE/data` directory, which contains the system-wide configuration file and process log files. The default installation also places the repository extent files and transaction logs in this location. While convenient for new users and small examples, this has never been a recommended configuration for enterprise-level GemStone production environments.

These default assumptions are not valid for an RPM installation, since this installs GemStone in a location that is expected to be shared and write-only, and the Stone's primary configuration file must be writable.

Instead, after installing the product tree, you will need to define directories to contain your extents, configuration files, and other important GemStone files.

Install the GemStone Product

1. Log in as root.
2. Perform the installation.

```
os$ rpm -i GemStone64-3.7.0-1.x86_64.rpm
```

The RPM will install the GemStone product tree to the following locations:

```
/usr/lib64/gemstone/3.7.0
```

3. The GemStone server product is now installed in `/usr/lib64/gemstone/3.7.0`.

In addition to subdirectories, this directory also contains the text files `version.txt`, which identifies this particular product and release of GemStone, and `externals.sha.txt`, which lists shas for external projects included in this version.

Note that with an RPM installation, do **not** run the `installgs` executable. The options in that script are not appropriate for an RPM installation.

Set the GemStone Key File

To run GemStone, you must have a key file for the correct version of GemStone/S 64 Bit and for the appropriate platform. The keyfile must be located where GemStone can find it on startup:

- ▶ A file specified by the `KEYFILE` configuration parameter in the configuration file used by the stone. This is not set by default, but may be defined to read a keyfile with any name in any location.
- ▶ `$GEMSTONE/sys/gemstone.key`
- ▶ `$GEMSTONE/sys/community.starter.key`

Licensed Customer key file

Keyfiles for GemStone version 3.6.x and earlier are not valid with v3.7. Licenced customers can email keyfiles@gemtalksystems.com or contact GemTalk Technical Support to request a keyfile for version 3.7 for their platform or platforms.

Community key file

The GemStone distribution includes a community key file, `community.starter.key`, with product and system limits per the Community and Web Edition License. See <https://gemtalksystems.com/licensing> for details on the license terms.

If you do not install a custom keyfile, this starter keyfile will be used instead.

Installing a keyfile

To specify the location and name of the keyfile using the `KEYFILE` configuration parameter, edit the configuration file that will be used by the v3.7 stone to include the location and name of the keyfile.

You may also put the keyfile in the default location, `/usr/lib64/gemstone/3.7.0/sys/gemstone.key`. This requires modifying the write permissions of the `$GEMSTONE/sys` directory; ensure you change this back to not writable, after this update.

The keyfile should have read-only permissions.

Configure the NetLDI

The NetLDI permits remote processes to interact with the repository. A NetLDI is required for some local and all remote sessions to log into GemStone.

There are two ways to set up a NetLDI so that it can provide services to all GemStone users: it can run as root, or it can run in guest mode with a captive account.

If you are running a development system without strict security requirements, or for new users, we recommend running in guest mode with captive account. This requires no further configuration of your GemStone installation.

If you will be running the NetLDI with authentication, and if multiple UNIX userIDs will be logging into GemStone, then set the `s` bit for the `$GEMSTONE/sys/netldid` executable. This allows an administrative UNIX userID to start a NetLDI process that will run as root.

```
os$ chmod u+s $GEMSTONE/sys/netldid
```

Named NetLDI Service

The NetLDI can be specified by port number, or you may configure a named NetLDI service. Named NetLDIs must be defined in your system services database to resolve to a port number, on all nodes that will be using GemStone.

Using a port avoids this configuration, however you must select a port that will consistently be available.

Configuring a named NetLDI service

1. Determine whether a service for the NetLDI name, such as `gs64ldi`, is already defined. How to do this will depend on how your system is set up. The GemStone distribution includes an executable that will allow you to do this:

```
os$ $GEMSTONE/install/getservbyname gs64ldi  
s_name=gs64ldi s_port = 50377 s_proto = tcp
```

2. If your desired NetLDI name is not defined, add an entry similar to the following to the system services database:

```
gs64ldi 50377/tcp #GemStone/S 64 Bit 3.7
```

Choose a port number that is not being used by another service. The port number should be in the range 49152 <= port <= 65535, to conform to IANA standards (<http://www.iana.org/assignments/port-numbers>).

3. If several machines will be running GemStone, have the UNIX system administrator update the system services database for each machine. This includes Windows client machines as well as UNIX nodes. Note that the port number must be the same for every machine.

Set the Environment

Perform the following steps to properly configure the operating environment.

1. Set the environment variable GEMSTONE to the full pathname (starting with a slash) of your new GemStone installation directory. For example:

```
os$ export GEMSTONE=/usr/lib64/gemstone/3.7.0
```

If GemStone was previously installed, or more than one installation GemStone exists on this machine, check for existing GemStone environment variables:

```
os$ env | grep GEM
```

If any environment variables exist and are not appropriate for the new installation, you should specifically unset each one.

Create a GemStone repository

If you are upgrading from a previous version, you may return to the appropriate chapter of this installation guide for upgrade instructions.

The following steps apply if you are setting up a new GemStone repository.

Setup Extents and Configuration

Create a new directory or directories, or determine an existing directory or directories, for your repository extent files, transaction logs, and system log files.

In a simple configuration, this may be a single directory. For non-trivial applications, multiple directories are recommended. The following instructions provide the simplest possible setup. The *System Administration Guide* provides much more guidance on setting up a scalable, multi-user GemStone configuration.

Simple Installation

1. Create a directory to hold your extent files, *stoneDataDirectory*.
2. Copy `$GEMSTONE/bin/extent0.dbf` and `$GEMSTONE/bin/gemstone_data.conf` to *stoneDataDirectory*:

```
os$ cp $GEMSTONE/bin/extent.dbf stoneDataDirectory
os$ cp $GEMSTONE/bin/gemstone_data.conf stoneDataDirectory
```

3. `chmod` both files to have user and group write.

```
os$ chmod ug+w stoneDataDirectory/*
```
4. Start the stone, using `-E` to pass in the configuration file, `stoneDataDirectory/gemstone_data.conf`.

```
os$ startstone -E stoneDataDirectory/gemstone_data.conf
```

This will start the stone with the default name **gs64stone**.
5. Start the NetLDI. This example starts the NetLDI in guest mode with captive account.

```
os$ startnetldi -E stoneDataDirectory/gemstone_data.conf -g  
-a adminUserName
```

This will start the NetLDI with the default name **gs64ldi**. The `-E` argument to `startnetldi` is optional; it allows RPC Gems started by the NetLDI to use the same configuration file, but while Gems may use a configuration file, it is not required.

Your GemStone repository is now running and ready for users to log in.

Complete GemStone Configuration

The following should be done by an administrative user, not as root.

Change Passwords for Administrative Accounts

GemStone comes with a number of built-in System user accounts, which are needed to perform administrative operations (such as adding application user accounts).

- ▶ The **DataCurator** account is used to perform system administration tasks.
- ▶ The **SystemUser** account ordinarily is used only for performing GemStone system upgrades.
- ▶ The **GcUser** account is used by the garbage collection task, which runs automatically as a separate login.

The initial password for these administrative accounts is `swordfish`.

Access to each of these accounts should be restricted; you should always change the passwords for these accounts, to provide basic security for your application.

The chapter entitled “User Accounts and Security” in the *System Administration Guide* tells you how to change the passwords.

Install the default TimeZone

GemStone/S 64 Bit is shipped with a default time zone of US Pacific. If you are in another Time Zone, edit the file `installtimezone.txt` in the GemStone upgrade directory, then file it in as `SystemUser`.

Add GemStone User Accounts

While it is possible for users to login as DataCurator, DataCurator has broad permissions and is not recommended. In a multi-user system, each users should have a GemStone accounts, which involves creating an individual UserProfile in GemStone.

The chapter entitled “User Accounts and Security” in the *System Administration Guide* provides information on how create accounts for GemStone users, and the options for authentication. This task can be done by executing Smalltalk code, or using GemBuilder for Smalltalk tools. See the *GemBuilder for Smalltalk Users’s Guide* for information on the GUI tools in GemBuilder.

Each user should:

- ▶ Set the environment variable GEMSTONE to the *full pathname* (starting with a slash) of the GemStone/S 64 Bit 3.7 directory.
- ▶ update their path to include the \$GEMSTONE/bin directory.
- ▶ Optionally, update the man path (MANPATH variable) to include the \$GEMSTONE/doc directory. GemStone provides man pages for utility functions.

These last two steps can be done using scripts that are part of the GemStone distribution. The directory \$GEMSTONE/bin contains the files `gemsetup.sh` and `gemsetup.csh`, which define the GemStone environment for users by modifying the PATH and MANPATH variables to include \$GEMSTONE/bin and \$GEMSTONE/doc, respectively.

For example:

```
os$ export GEMSTONE=installdir
os$ export PATH=$GEMSTONE/bin:$PATH
os$ export MANPATH=$MANPATH:$GEMSTONE/doc
```

If the user will use GemStone frequently, consider adding these steps to the login shell initialization file.

Further Configuration and Administration

This chapter has guided you through installation of GemStone/S 64 Bit 3.7, with the objective of getting a simple, default configuration up and running.

The next chapters explain the process of upgrading a previous version of GemStone/S 64 Bit to version 3.7.

For more information and details on customizing your GemStone object server, Gem client processes, and setting up distributed configurations, see the *System Administration Guide*.

Upgrading from GemStone/S 64 Bit 3.4.x or later

This chapter describes how to upgrade an existing GemStone/S 64 Bit 3.4.x, 3.5.x, or 3.6.x installation to GemStone/S 64 Bit version 3.7.

To upgrade from GemStone/S 64 Bit version 3.2.x and earlier, you must first upgrade to a more recent version, and then upgrade to v3.7. Upgrade from 3.3.x to 3.7 is not disallowed, but is not tested or supported.

If you are using GemBuilder for Smalltalk (GBS), you also need to upgrade the client libraries that are used by GBS. You may also need to upgrade your version of GBS; versions of GBS earlier than 8.7 and 5.4.7 may be usable, but are not fully supported with v3.7. See Chapter 4 for details.

Keyfiles

Keyfiles for GemStone/S 64 Bit 3.6.x or earlier will not work with v3.7.

To acquire a keyfile for version 3.7, email keyfiles@gemtalksystems.com, or contact GemStone Technical Support, preferably providing your existing keyfile.

Keyfiles also manage access to GemConnect, GemBuilder for Java, and the X509-Secured GemStone feature. If you are using these add-on products, you must use a keyfile with the appropriate permissions.

Upgrade Procedure

The following list summarizes the steps necessary to perform the upgrade to GemStone/S 64 Bit v3.7.

- ▶ Prior to Upgrade in existing application. 24
- ▶ Prepare for Upgrade 24
- ▶ Perform the Upgrade 26
- ▶ Post-upgrade Application Code Modifications. 28
- ▶ Make Backup 28
- ▶ Configure GCI clients and GBS. 28

NOTE

The following instructions use the version number 3.6.6 to refer to the version you are upgrading from, and version number 3.7 indicate the target version you are upgrading to.

Prior to Upgrade in existing application

1. Check for use of deprecated methods

Verify that your application does not invoke any methods that were deprecated in previous releases, by enabling error or logging on deprecation in your existing repository. Deprecated methods are subject to removal in major releases; finding them before upgrading allows the deprecation messages to provide replacement instructions.

For details on finding deprecated methods, refer to the *Programming Guide for GemStone/S 64 Bit*.

2. File out modifications to GemStone classes

File out any modifications or additions you made to GemStone/S 64 Bit kernel class methods. For more information about fileout, see the *GemStone/S 64 Bit Topaz Programming Environment*.

You will need to carefully compare these changes with GemStone/S 64 Bit 3.7 kernel methods, and refer to the *Release Notes* for version 3.7 to determine whether your changes are still necessary or appropriate.

CAUTION

Any changes that you have made to the GemStone/S 64 Bit kernel classes will be lost during upgrade; you MUST file these out in order to preserve the changes in version 3.7.

Prepare for Upgrade

1. Install and configure GemStone/S 64 Bit 3.7

Install GemStone/S 64 Bit 3.7 to a new installation directory, separate from the installation directory for version 3.6.6, as described in Chapter 1, starting on page 7.

Configure GemStone/S 64 Bit 3.7 the way you expect to use it — that is, with the appropriate extent locations and sizes.

If you copy the configuration files from your previous version to the version 3.7, be sure to review any changes in configuration parameters to determine if changes are needed.

You should ensure that adequate space is available for extents, transaction logs, and a backup during the upgrade. You must provide space for the extents and transaction logs for both repositories, the old and the new.

2. Reset SystemUser password

Log in to the version 3.6.6 system as a user with OtherPassword privilege, such as DataCurator, and reset the SystemUser password to 'swordfish':

```
topaz 1> printit
(AllUsers userWithId: #SystemUser) password: 'swordfish' .
System commitTransaction.
%
```

The upgrade script logs in with the SystemUser account and the default password, and resets the password for DataCurator and GcUser.

3. Stop user activity

Log in to the version 3.6.6 system as a user with SessionAccess and SystemControl privileges, such as DataCurator, and halt all user activity on the repository.

```
topaz 1> printit
System stopUserSessions.
%
```

4. Shut down the repository

You may now shut down the Stone. At the UNIX command line:

```
stopstone stone366
```

where *stone366* is the name of the version 3.6.6 stone on this machine. The repository must be cleanly shut down to avoid needing recovery when it is restarted with the new version's executables.

5. Set up the version 3.7 environment.

Set the environment variables required for the upgrade. For example:

```
export GEMSTONE=InstallDir37
export PATH=$GEMSTONE/bin:$PATH
export upgradeLogDir=tempDir
```

where *InstallDir37* is the GemStone/S 64 Bit version 3.7 installation and *tempDir* is a temporary directory for which you have write permission.

6. Copy extent files

Copy your version 3.6.6 extent files into the location specified by the v3.7 configuration file option DBF_EXTENT_NAMES:

- a. Identify the complete set of extent files that are used by your 3.6.6 stone. This can be found by examining the configuration file for the version 3.6.6 repository, looking for the last entry for DBF_EXTENT_NAMES.

- b. The target location is the setting for `DBF_EXTENT_NAMES` in the version 3.7 installation. Copy each of these extent files to the target location.

For example:

```
cp dataDirFor366/extent0.dbf dataDirFor37
cp dataDirFor366/extent1.dbf dataDirFor37
cp dataDirFor366/extent2.dbf dataDirFor37/
```

Before upgrading, ensure that there are no transaction logs from a previous version of GemStone/S 64 Bit in any of the transaction log locations specified in the configuration file that will be used by version 3.7.

Perform the Upgrade

1. Start the Stone

Start the 3.7 Stone on the 3.6.6 extents you just copied:

```
startstone stoneName37
```

2. Upgrade image

Ensure you are in a directory to which you have write permission, and run the upgrade script.

The upgrade is performed by the script `upgradeImage`. This script has optional switches to specify the stone name and to set to size of the `GEM_TEMPOBJ_CACHE_SIZE` used for the upgrade process.

```
upgradeImage [-h] [-c cacheSize] [-s stoneName]
```

-h prints this usage information.

-c *cacheSize* sets the size of the `GEM_TEMPOBJ_CACHE_SIZE`; if this is not used, the script will default to use a value of 100000.

-s *stoneName* sets the name of the running stone to upgrade; if this option is not used, the script will default to **gs64stone**.

For example,

```
upgradeImage -s stoneName37
```

The script will prompt you to press the return key to begin.

The script invokes subordinate scripts to complete the upgrade. The upgrade process will take some time. You can examine the progress, if desired, by examining the file `$upgradeLogDir/upgradeImage*.out`.

The script should complete with the message:

```
Upgrade completed. No errors detected.
```

If not, please preserve the Stone log file and the contents of `$upgradeLogDir`. Contact your internal GemStone support person or GemStone Technical Support.

3. Restore System Account passwords

Log in to GemStone/S 64 Bit version 3.7 as DataCurator or SystemUser, and change the password for SystemUser, DataCurator, and GcUser to a secure password, such as the passwords used for these accounts in v3.6.6. For example:

```
topaz 1> run
(AllUsers userWithId: 'SystemUser') password: '366Password'.
(AllUsers userWithId: 'GcUser') password: '366Password'.
(AllUsers userWithId: 'DataCurator') password: '366Password'.
System commitTransaction
%
```

where *366Password* is the account password used in version 3.6.6.

4. Recompile methods referencing instance of reimplemented class definitions

GsHostProcess and JsonParser have new implementations in v3.7 that affect some applications.

GsHostProcess

If you have methods that encode a reference to the GsHostProcess class (and you have not recompiled these methods after a previous upgrade to 3.6.4 or later), these methods must be recompiled. After upgrade, the compiled method will refer to the class ObsoleteGsHostProcess, which does not support new GsHostProcess primitives.

Affected instances can be found using an expression such as:

```
(ClassOrganizer newExcludingGlobals) referencesToObject:
  (ObsoleteClasses at: #ObsoleteGsHostProcess)
```

This returns instances of GsNMethod. Once these have been examined, recompile can be done using an expression of the form:

```
aGsNMethod recompileFromSource
```

Note that this recompiles using the SymbolList of the current session, which may affect other class references in the method.

If you refer to the GsHostProcess class by name or symbol, the correct class will be found in the SymbolList, and no further action is needed.

JsonParser

The JsonParser class has been replaced in v3.7 with a non-PetitParser based implementation, which is smaller and faster. The basic `parse:` API is the same, but PetitParser-specific methods are not present in the new implementation.

If you are using JsonParser to simply parse JSON, you do not need to do anything. The methods reference the older class (now JsonPetitParser); if you want faster JSON parsing, you can recompile methods with class references. Affected instances can be found using an expression such as:

```
(ClassOrganizer newExcludingGlobals) referencesToObject:
  (Globals at: #JsonPetitParser)
```

If you are using JsonParser as part of a PetitParser customization, references to the class by name must be updated to refer to 'JsonPetitParser' instead of 'JsonParser'.

References from compiled methods to the class will continue to work, but it is recommended to edit the source code, so that a future recompile does not

inadvertently introduce a reference to the new implementation, or fail due to a reference to an inherited instance variable.

GsDevKit Upgrade

If you are using the open-source Development Kit for GemStone/S 64 Bit (GsDevKit, also referred to as Seaside or GLASS), you will need to perform another step to upgrade your GsDevKit image. This step upgrades the GsDevKit base code, and you will also need to reload your application code.

For details, see Chapter 3, starting on page 29.

When you have completed the GsDevKit upgrade, continue with the upgrade process and perform the following steps.

Post-upgrade Application Code Modifications

1. Reinstall any other GemStone products that modify kernel classes.

If you use GemConnect or GemBuilder for Java, you must reinstall the appropriate version of these products into your repository at this time.

To install, use the procedure in the *Installation Guide* for that product.

2. File in Kernel class changes

If you have modified any kernel class methods of the previous version or if you have added methods to kernel classes, carefully compare your changes with the changes in version 3.7 to see whether your changes are still necessary or appropriate. Carefully review the *Release Notes* for each intervening version, as well as examining code in the image. If the kernel class changes are still applicable, file in the changes, verify that errorcount is 0, and commit.

Make Backup

1. Make backup

At this point, you should create a full backup of the upgraded repository.

Configure GCI clients and GBS

1. Recompile User Actions

It is recommended to recompile and relink User Action libraries.

2. Configure GBS

If you are using GBS clients, ensure you are running a supported version of GBS and client Smalltalk. Versions of GBS earlier than 8.7 for VW and GBS 5.4.7 for VA may be usable but are not fully supported with GemStone/S 64 Bit v3.7.

Configure GBS to use the version 3.7 client libraries. Depending on the GBS version you are upgrading from, the required libraries, library naming conventions, and the process GBS uses to identify the correct library to load may have changed.

See Chapter 4, 'Configuring GBS for GemStone/S 64 Bit' for details. If your GBS clients run on a different platform than your GemStone server, refer to the *Installation Guide* for that platform.

Upgrading GLASS/GsDevKit Applications

This chapter describes the additional upgrade step that applies when upgrading an application that is using variants of the open-source Development Kit for GemStone/S 64 Bit (GsDevKit, referred to also as also as GLASS or Seaside) to GemStone/S 64 Bit v3.7. The term GsDevKit is used to collectively refer to any of these environments.

The process described here can be used to upgrade repositories using GLASS, GLASS1, the older GsDevKit environment, and tODE. There are a number of possible configurations and there may be additional setup required for some environments. For more background on these environments, see

https://github.com/GsDevKit/GsDevKit_upgrade/blob/master/README.md#upgrading-glassgsdevkit-applications-to-gemstone-350

If you are using the most recent version, from github.com/GsDevKit/GsDevKit_home, then you may use the upgrade scripts provided there to perform the entire upgrade, rather than using the instructions in this *Installation Guide*.

The complete process for upgrading includes the GemStone standard upgrade, followed by additional GsDevKit upgrade. Upgrading GemStone is described in earlier chapters of this *Installation Guide*; Chapter 2. You will need to follow the steps in that chapter, which will note the point at which the GsDevKit upgrade takes place.

Upgrade Procedure

The GsDevKit upgrade occurs partway through a standard GemStone upgrade.

To upgrade a GsDevKit/GLASS application:

- First, install the new version of GemStone, and upgrade your repository, according to the instructions in Chapter 2.
- After the **upgradeImage** step, you will upgrade the GsDevKit application as described in this chapter.
- After the GsDevKit upgrade completes, continue with the remaining steps of the GemStone upgrade in Chapter 2.

1. Ensure that GemStone 3.7 is installed and your repository upgraded

You must first follow install version 3.7 and follow the instructions in Chapter 2, before you can upgrade your GsDevKit application. These instructions will let you know at which point you perform the GsDevKit upgrade.

You should also have confirmed that your application code has been updated as required. We recommend that you install your application code in a test GemStone/S 64 Bit v3.7 repository, and verify that your code is working correctly, making changes as necessary. Do this prior to upgrading the data in your application, to ensure the upgrade process will go smoothly.

2. If necessary, customize the upgrade instructions

There are many ways a GsDevKit or GLASS application may be built, and a variety of packages that can be loaded. The **upgradeSeasideImage** script will upgrade a hypothetical standard installation, but there may be customizations required in specific cases.

The upgrade is performed by an upgrade script. By default, this is `$GEMSTONE/upgrade/createGsDevKit_upgrade.topaz`.

The default upgrade script is a file containing topaz commands for the upgrade.

Example 3.1 Default upgrade instructions in `createGsDevKit_upgrade.topaz`

```
run
  UserGlobals
    at: #GsDevKit_Image_Upgrade
      put: (GsuAbstractGsDevKitUpgrade
        upgradeUserName: SeasideUpgradeUser).
  System commitTransaction
%
```

In this script, `SeasideUpgradeUser` is an internal global that by default resolves to `DataCurator`.

Customizing upgrade

To create customized upgrade instructions, make a copy of this file, edit the copy, and pass the path to your customized upgrade script file as an argument to the **upgradeSeasideImage** script.

The following example shows a customized file.

Example 3.2 Example customized upgrade

```

run
UserGlobals
  at: #GsDevKit_Image_Upgrade
  put: ((GsuAbstractGsDevKitUpgrade
        upgradeUserName: 'DataCurator'
        upgradeSymbolDictName: #UserGlobals)
        bootstrapApplicationLoadSpecs: {
          { 'Metacello' .
            'github://dalehenrich/metacello-work:master/repository' } .
          { 'GLASS1' .
            'github://glassdb/glass:master/repository' .
            #( 'default' 'Base' 'Announcements') } .
          { 'Seaside3' .
            'github://SeasideSt/Seaside:master/repository' .
            #( 'CI' ) }
        } ).
System commitTransaction
%
```

Further information about LoadSpecs is provided in the comment for `$GEMSTONE/upgrade/createGsDevKit_upgrade.topaz`.

You may also refer to the example scripts on github, at github.com/GsDevKit/GsDevKit_upgrade/tree/master/bin.

3. Perform the Upgrade

The GsDevKit upgrade is performed by the script **upgradeSeasideImage**, which is located in the `$GEMSTONE/seaside/bin` subdirectory.

Prior to executing **upgradeSeasideImage**, if you need to do a customized upgrade, you should have set up the upgrade script as described above. This is provided a argument to this script.

```
upgradeSeasideImage [ -c tempObjCacheSize ] [ -s stoneName ] [ -u gemstoneUser ]
  [ -p password ] [ -P pathToUpgradeScript ] [ -W ] [ -E ]
```

-c tempObjCacheSize

set the size of the `GEM_TEMPOBJ_CACHE_SIZE`; if omitted, default is 100000.

-E enable isNil optimization (leave the isNil selector optimized, and not recompilable). Base GemStone **upgradeImage** enables isNil optimization by default, but there are potential problems with isNil optimization in some Seaside projects. **upgradeSeasideImage** disables isNil optimization, unless the *-E* option is specified.

See https://github.com/GsDevKit/GsDevKit_upgrade/issues/30 for more information.

-u gemstoneUser

specify the GemStone user name, if seaside was installed as a user other than DataCurator. If omitted, defaults to DataCurator.

-p password

specify the password for *gemstoneUser*. If omitted, defaults to swordfish.

- P** *pathToUpgradeScript*
path to customized to GsDevKit_upgrade instance creation script. If omitted, `$GEMSTONE/upgrade/createGsDevKit_upgrade.topaz`.
- s** *stoneName*
set the name of the running stone to upgrade; if omitted, default is **gs64stone**.
- W** enable `GEM_LISTEN_FOR_DEBUG`, and set up to allow debugging via **debuggem**.

For example,

```
os$ $GEMSTONE/seaside/bin/upgradeSeasideImage -s stoneName37
```

The script will prompt you to press the return key to begin.

The script should complete with the message:

```
Seaside Upgrade completed. No errors detected.
```

If you encounter errors in **upgradeSeasideImage**, and you are experienced at using **topaz**, you may use the **-W** argument, and follow the instructions in the *Topaz User's Guide* for v3.6 on how to use **debuggem** to attach to the upgrade process and debug the error.

4. Load your Application Code

After upgrade has successfully completed, reload your application code.

5. Complete the Upgrade Process

To complete the upgrade, return to Chapter 2 and complete the remaining upgrade steps.

Configuring GBS for GemStone/S 64 Bit

This chapter describes how to configure or update your client Smalltalk application using GemBuilder for Smalltalk (GBS) on Linux to run with GemStone/S 64 Bit version 3.7.

This chapter describes Linux clients only; for GBS clients running on Windows, see the *GemStone/S 64 Bit Windows Client Installation Guide*. GBS clients are not supported on AIX or Macintosh. For a table of all supported GBS and client Smalltalk platforms, see the *GemStone/S 64 Bit Release Notes* for v3.7.

The GemStone/S 64 Bit v3.7 server requires a compatible version of GBS; versions of GBS/VW earlier than 8.7 or GBS/VA earlier than v5.4.7 are not supported with GemStone/S 64 Bit v3.7 or later.

In addition to using the appropriate version of GBS, you must use GemStone/S 64 Bit 3.7 client libraries with your GBS client application, to be able to log in to the v3.7 server. These libraries are specific to the GemStone/S 64 Bit server version and to the client platform.

GemStone/S 64 Bit provides both 64-bit libraries and 32-bit libraries.

- ▶ With 64-bit VisualWorks environments, you must use the 64-bit GemStone client libraries, and it is possible to login either RPC or linked.
- ▶ With 32-bit VisualWorks Smalltalk environments, you must use 32-bit libraries, and can only login in RPC. 32-bit processes cannot load 64-bit libraries.

For instructions on installing and configuring GBS, see the *GemBuilder for Smalltalk Installation Guide* for the appropriate version of GBS.

Supported GBS client Smalltalk Platforms with Linux

GBS version 8.7 or later is required for use with GemStone/S 64 Bit v3.7 or later servers.

The following VisualWorks client Smalltalk versions are supported:

GemBuilder for Smalltalk v8.7

■ **VisualWorks 9.1.1 (32-bit and 64-bit)**

- ▶ RedHat Linux 9.2, 8.7 and 7.9; Ubuntu 22.04 and 20.04

For supported client Smalltalk versions on Windows, refer to the *GemStone/S 64 Bit Windows Client Installation Guide*.

GBS Setup or Upgrade Procedure

Shared Libraries for GBS Client Node

The GBS client requires a set of shared libraries (.so files) that are provided as part of the GemStone server product distribution. When these are loaded into the VisualWorks image in which GBS code is installed, the GBS client can log into the GemStone server.

The shared libraries must be the same version as the GemStone server. Since they are loaded into the client smalltalk VM, they must be appropriate for the client platform and client executable bit size (32-bit or 64-bit).

If your GBS client is on a different platform than your GemStone/server, you will need to download the version-specific libraries for the platform that the GBS client is running on.

You can either install the full GemStone/S 64 Bit Server on your GBS client node, or copy just the specific shared libraries you need.

Install full GemStone/S 64 Bit on client

If your clients run on the same machine as the server, you have no need to do anything further – you can use the libraries in their locations in the existing server installation.

Otherwise, you may find it useful to install the full GemStone/S 64 Bit Server on the client.

If you will run linked sessions on the GBS client, or other configurations in which the gem is on the same node as the GBS client, you will need much of the GemStone server installation. The GemStone/S 64 Bit installation also includes tools such as topaz, gslis, and VSD, that may be useful to run on your client.

Install GemStone/S 64 Bit on the client machine following the instructions in the Installation Guide for the client platform. You do not need to configure an extent or perform similar server tasks.

Copy only specific client libraries

If you will only be running RPC sessions, and do not require tools such as gslis, vsd, or topaz on the client, you do not need to install the full GemStone/S 64 Bit Server on the client node. You may copy only the small set of library files that GBS requires. If you want to run linked sessions as well as RPC, you will need a full server installation on the client.

The files that are required depend on the bit size of the Client smalltalk application.

32-bit VisualWorks clients, RPC only

With 32-bit VisualWorks, the following files are needed:

```
$GEMSTONE/lib32/libgcirpc-3.7.0-32.so
$GEMSTONE/lib32/libssl-3.7.0-32.so
$GEMSTONE/lib32/libkrb5-3.7.0-32.so
```

64 bit VisualWorks clients, RPC only

With 64-bit VisualWorks, the following files are needed:

```
$GEMSTONE/lib/libgcirpc-3.7.0-64.so
$GEMSTONE/lib/libssl-3.7.0-64.so
$GEMSTONE/lib32/libkrb5-3.7.0-64.so
```

GBS provides a number of options as to where on the client machine to place the shared libraries. Refer to the *GemBuilder for Smalltalk Installation Guide* for details on these options

Upgrade GBS to a version that supports v3.7

Versions of GBS/VW earlier than 8.7 are not supported with GemStone/S 64 Bit v3.7.

If you are currently running with an older version of GBS, you should upgrade the GBS client, and possibly also upgrade VisualWorks, to a more recent version for v3.7.

See the *GemBuilder for Smalltalk Installation Guide* for installation instructions, and the *GemBuilder for Smalltalk Release Notes* for details on the changes in these versions.

Update GBS to reference v3.7 libraries

Once you have installed the GemStone server on the GBS client machine, or copied the appropriate shared libraries, you need to ensure that the client Smalltalk executable – a VisualWorks application – will load the v3.7 libraries.

Determining library name to specify

Whether or not you have a full server installation on the client or have copied a few libraries, there is a specific library name you will specify to have GBS load using the `libraryName:` parameter.

32-bit VisualWorks clients, RPC logins only:

```
libgcirpc-3.7.0-32.so
```

64-bit VisualWorks clients, RPC logins only:

```
libgcirpc-3.7.0-64.so
```

64-bit VisualWorks clients, Linked and RPC logins:

```
libgbslnk-3.7.0-64.so
```

Setup GBS to load the new libraries

- If you have set the GBS configuration parameter `libraryName:`, update this to the new library name, and save your image.
- If you have set the GBS configuration parameter `libraryName:` to an empty string, ensure that no other client libraries of the same name are in the current working directory or the bin directory or subdirectory of your VisualWorks image's VISUALWORKS directory.

- For a new GBS application, refer to the *GemBuilder for Smalltalk Installation Guide* for details on the library loading setup options.

Stop and restart the client VM

GBS loads the client libraries into the client Smalltalk VM the first time a GemStone server call is made after each startup of the VM.