GemStone/S 64 Bit™ Installation Guide

For Red Hat-compatible Linux on x86_64, using RPM

Version 3.7.2

December 2024

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

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".

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.
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**, m**acOS**, 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 3.7.2 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*.

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 4. Upgrading GLASS/GsDevKit Applications

Chapter 5. Configuring GBS for GemStone/S 64 Bit

Chapter

1 Installing GemStone/S 64 Bit Version 3.7.2

This chapter describes the procedure for installing GemStone/S 64 Bit™ version 3.7.2, using the RPM (Red Hat Package Manager) installation process, and for additional OS configurations that can optimize your GemStone application.

If you are upgrading to this release from a previous version of GemStone/S 64 Bit, after installing v3.7.2, follow the instructions in the appropriate later chapter of this Installation Guide. These upgrade instructions will provide details on steps that need to be taken before and after the installation described here.

The RPM installation differs from the historic installation of GemStone from zipped distributions. The historic GemStone installation defaulted to using GemStone distribution subdirectories for configuration, repository extent, transaction logs, and user action libraries. 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 the RPM installation installs GemStone in a location that is expected to be shared and read-only.

If you are upgrading from a non-RPM installation, note that if your existing application relies on files in `$GEMSTONE/data/`, including the `system.conf` file, or uses `$GEMSTONE/ualib` 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.

Review the Installation Procedure

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

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 and later will not run on x86_64 CPUs older than listed. These versions are built using machine instructions that are not available on older CPUs. v3.7.2 cannot run on these CPUs, regardless of the Linux OS version.

RAM and space

- While small installations can run on systems with only a few GB of physical RAM, increasing RAM, and so allowing a larger Shared Page Cache, is important for GemStone performance.
- Total swap space should be at least 20-25% of the amount of RAM. On memory-constrained systems, more swap may be useful, but does affect performance.

Disk space

- Space for the installed distribution files—you need approximately 1.3GB for GemStone/S 64 Bit, and additional space for other products.
- Additional space as required for your repository.
- ext4, XFS, and ZFS are supported; XFS is recommended; for large systems, XFS or ZFS should be used in preference to ext4.

  Extent files cannot be accessed via CIFS, and NFS requires special configuration.

Operating system

- Red Hat-compatible Linux ES 9.4 on **x86_64**
  kernel version 5.14.0-362.13.1.el9_3.x86_64 and glibc-2.34-100.el9.x86_64
- Red Hat-compatible Linux ES 8.9 on **x86_64**
  kernel version 4.18.0-513.24.1.el8_9.x86_64 and glibc-2.28-236.el8_9.13.x86_64
- Red Hat-compatible Linux ES 7.9 on **x86_64**
  kernel version 3.10.0-1160.119.1.el7.x86_64 and glibc-2.17-326.el7_9.x86_64

  GemStone performs testing on a mixture of Red Hat, CentOS, and Rocky 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 v3.7.2 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 Linux ES 9.4: gdb 11.4.1
- Red Hat Linux ES 8.9: gdb 9.1
- Red Hat Linux ES 7.9: gdb 9.1

C/C++ Compiler

GemStone requires a C/C++ compiler if you are developing C or C++ code for user actions or a C or C++ application; as described in the *GemBuilder for C* manual.

If you have committed your FFI generated code, or compiled your user actions or application, the C/C++ compiler is not needed for application execution.

The compiler is required for application execution only when using GsTsExternalSession to access remote Stones running a different version of the GemStone server.

- Red Hat Linux ES 9.4: gcc/g++ 11.4.1
- Red Hat Linux ES 8.9: gcc/g++ 8.5
- Red Hat 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.

For further OS configuration and optimizations, see "Additional Configuration of the Operating System" on page 12

Install GemStone v3.7.2 RPM distribution

The GemStone distribution must be installed as the root user.

Note that starting GemStone and GemStone administration should be done as a regular user, not as root. For GemStone administration, select an existing user, or create a specific user to be the GemStone administrative user. This administrative user is referred to in these instructions as *gsAdminUser*.

This installation process install the distribution in a directory under `/usr/lib64`, owned by root and read-only.

1. Log in as root.
2. The GemStone/S 64 Bit RPM distribution is provided as a file **`GemStone64-3.7.2-1.x86_64.rpm`**. Move this file to an accessible location.
3. Perform the installation. The installation command depends on the version of Red Hat.

   On Red Hat 8x or 9.x:

   ```
   os$ dnf install GemStone64-3.7.2-1.x86_64.rpm
   ```

   On Red Hat 7x:

   ```
   os$ yum install GemStone64-3.7.2-1.x86_64.rpm
   ```

   Note: Do not use `dnf`/`yum` on older versions of GemStone, which did not include the version number in the package name.
4. The GemStone server product is now installed in `/usr/lib64/gemstone/3.7.2`.
5. Log out root user.

The GemStone distribution is now installed.

In addition to subdirectories, the installed GemStone distribution 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 the RPM installation process does not install the VSD product. VSD is not provided as an RPM. To use VSD, download and install manually from https://gemtalksystems.com/products/vsd/, following instructions in the *VSD User's Guide*.

Configure the GemStone Installation

After installing the GemStone distribution, you need to set or reset environment variables, ensure you have the appropriate keyfile installed, and that your NetLDI user is configured correctly.

Set Environment Variables

To use GemStone, you will need to define the GEMSTONE environment variable, and in most cases, set the OS search path.

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

   ```
   os$ export GEMSTONE=/usr/lib64/gemstone/3.7.2
   ```
2. Set your OS search path to include the $GEMSTONE/bin directory.

   ```
   os$ export PATH=$GEMSTONE/bin:$PATH
   ```
3. If this machine has run previous versions of GemStone, or will run multiple versions, it is important to review all GemStone-related environment variables for ones that are

incorrect for the new version, such as GEMSTONE_NRS_ALL, GEMSTONE_EXE_CONF, and GEMSTONE_SYS_CONF.

Check for existing GemStone environment variables:

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

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

Set the GemStone Keyfile

The GemStone/S distribution includes a community keyfile, which will be used if a keyfile is not specified.

You may use a licenced customer keyfile (that is, a GemTalk-provided keyfile for your license) of the correct platform from any v3.7.x version with v3.7.2; keyfiles from 3.6.x and earlier are not valid with v3.7.2.

A keyfile must be located where GemStone can find it on startup, in order:

- 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 keyfile

You may continue to use your v3.7.x keyfile with version 3.7.2. If you are upgrading from v3.6.x or earlier and need a new keyfile, or you have questions about your keyfile or license limits, email keyfiles@gemtalksystems.com, or contact GemTalk Technical Support.

Community keyfile

The GemStone distribution includes a community keyfile, `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.

Installing a keyfile

To specify the location and name of a customized keyfile using the KEYFILE configuration parameter, edit the configuration file that will be used by the v3.7.2 stone to include the location and name of the keyfile.

You may also put the keyfile in the default location, `$GEMSTONE/sys/gemstone.key`. This requires adding write permission to the `$GEMSTONE/sys` directory; ensure that you remove write permission after the update.

Configure Services for a Named 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. Logins can specify a NetLDI by name or by port number.

Defining a NetLDI service name is only necessary if you will use named NetLDIs. If you start a NetLDI using a port number, then no configuration of the system services database is needed. The login parameters can use the NetLDI's port rather than a name.

If you are using named NetLDIs, and your configuration includes multiple machines, (including clients on Windows or other platforms), you must update the services database on each node with an entry mapping the NetLDI name to the same port number.

To configure a named NetLDI:

1. Determine how the services file is setup on your system. /etc/services is a common location for this configuration. Otherwise, consult your system administrators.
2. Determine whether a service for the NetLDI name, such as `gs64ldi`, is already defined. The GemStone distribution includes an executable that will allow you to check for a definition:

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

3. 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 3.7.2
   ```

   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).

If several machines will be running GemStone, the system services database must be updated for each machine, including Windows clients as well as UNIX nodes. The port number must be the same for every machine.

GemStone/S 64 Bit v3.7.2 is now installed and configured.

To create a test repository, see the following section, "Additional Configuration of the Operating System".

For configuration of your Operating System to optimize GemStone, see "Additional Configuration of the Operating System" on page 12.

If you are upgrading from an earlier version, continue with the instructions in Chapter 3.

Additional Configuration of the Operating System

There are a number of Operating System-level configurations that can affect your GemStone environment. This includes a number of settings that are important for very large applications, but are not needed for simple test installations.

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. 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 contents of the file:

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

2. Transparent Huge Memory Pages

The default size for memory pages on Linux is 4KB. Linux supports 2MB and 1GB Huge Pages; this document, and GemStone code, also 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.

Reserved huge memory pages are described in the next section, "Reserved Huge Memory Pages" on page 14. These are recommended for the Shared page Cache. Transparent huge memory pages improve performance for Gems with large temporary object memory.

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 "`madvise`" or "`advise`", 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 "`madvise`".
- `/sys/kernel/mm/transparent_hugepage/defrag`
  should be set to "`madvise`".
- `/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 [madvise] never
```

3. Reserved Huge Memory Pages

Linux supports explicit configuration for 2MB and 1GB huge pages that are specifically reserved for an application, as well as transparent huge 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 improves 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, which are available on all Linux distriutions.
- Repositories with multi-GB shared page cache sizes benefit more from 1GB pages, which are available on some CPUs.

  To determine if your CPU supports 1 GB huge pages, execute:

  ```
  os$ cat /proc/cpuinfo | grep -o pdpe1gb | head -1
  ```

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 18.

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 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_MB to 2 (for 2 MB) 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.

4. Locking the Shared Page Cache in memory

It is recommended that you have sufficient memory to avoid the need to swap portions of the shared page cache 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 14), 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$ 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 18.

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. /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
```

On some platforms/versions, the default value is yes. This means that when the session that started the Stone logs out, the Stone's shared semaphore array is deleted, causing fatal errors. This excluded processes started by sessions running as root or as UNIX users with uids lower than 1000, which are defined as system users.

8. 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 ba 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.

Chapter

2 Setting up a new Repository

This chapter describes how to setup a new GemStone repository.

For new users, we recommend that you create a GemStone repository initially on a single machine, to ensure that all the pieces work together. Chapter 1 of the *System Administration Guide* provides an introduction to GemStone, and this manual provides details on setting up configurations to meet specific application requirements.

Set up a new Repository

There are many ways to set up a GemStone repository, depending on your system requirements.

The repository setup instructions in this Chapter describes setting up a basic repository, suitable for a single user, or a small system with low security requirements.

Multi-user systems and system with stringent security requirements will need special configuration to allow multiple users to login and access the database while ensuring unauthorized users cannot access or modify data. Large or high-volume repositories require separation of the extents holding data, the transaction logs holding the records of changes, and the log files.

The options for setting up a GemStone repository that is optimized for larger, higher-volume, highly trackable and maintainable, and for more highly secure repositories are described in the *System Administration Guide*. Please consult GemTalk Technical Support for additional assistance.

Simple Data-conf repository setup using createNewGemStoneRepository

A Data-conf installation is a repository configuration in which all the repository-specific files are located in a single directory (one that is not within the GemStone distribution directory), and a specific configuration file, `gemstone_data.conf`, is used as the **-E** argument to GemStone processes.

While convenient for new users and small examples, this is not a recommended configuration for enterprise-level GemStone production environments. The *System*

Administration Guide provides much more guidance on setting up a scalable, multi-user GemStone configuration.

The special configuration file `gemstone_data_conf`, makes setup of this environment easy. The Stone, NetLDI, and linked topaz startup commands use **-E** to specify this configuration file.

To set up a Data-conf repository, use the script **createNewGemStoneRepository**, passing in an argument of the path to your desired repository root directory. This directory must not exist or be empty, and the path must be writable.

For example, to create a new repository directory under the existing directory `/gshost/`, with the directory name `testGS`, do the following:

1. Create the repository using the script:

   ```
   os$ $GEMSTONE/install/createNewGemStoneRepository
       /gshost/testGS
   ```

   This will printout commands to start the Stone and NetLDI.

2. Start the stone, passing in the configuration file using the **-E** argument.

   ```
   os$ startstone -E /gshost/testGS/gemstone_data.conf
   ```

   This will start the stone with the default name **gs64stone**.

3. Start the NetLDI. This example starts the NetLDI in guest mode with captive account.

   ```
   os$ startnetldi -g -a gsAdminUser
   ```

   or

   ```
   os$ startnetldi -E /gshost/testGS/gemstone_data.conf -g -a
       gsAdminUser
   ```

   This will start the NetLDI with the default name **gs64ldi**. The **-E** argument to **startnetldi** is optional; it is only useful if you have other configuration file customizations that apply to Gem sessions.

Using a NetLDI by Port number

Without the *nameOrPortNumber* argument, the NetLDI will be started with the default name **gs64ldi** and a randomly selected port. If you have not defined gs64ldi (see "Configure Services for a Named NetLDI" on page 11), you may use a port number; this requires an additional argument.

```
os$ startnetldi -g -a gsAsminUser  netldiNameOrPort
```

For example,

```
os$ startnetldi -g -a gsAsminUser 54321
```

Verifying your new repository by logging in

Verify your Stone and NetLDI by logging in with topaz, the command line environment. This verification is done on the same host as the stone, with the gemstone environment setup. You will need to set login parameters, and execute the topaz **login** command.

You should ensure that your envionment is setup as described on page 10, that is, with the GEMSTONE environment and path set up.

For example:

```
os$ export GEMSTONE=/usr/lib64/gemstone/3.7.2
os$ export PATH=$GEMSTONE/bin:$PATH

os$> topaz
<startup headers>
topaz> set stone gs64stone user DataCurator password swordfish
topaz> login
```

Configure and set up users in new repository

Users log into GemStone using a GemStone user account (UserProfile), which is distinct from the UNIX user account. Logging into GemStone normally requires first authentication for your unix userId, and then login to GemStone with the GemStone userId and password.

Note that there are ways to configure a streamlined authentication, as described in the The chapter entitled "User Accounts and Security" in the *System Administration Guide*.

Change Passwords for Administrative Accounts

GemStone comes with built-in administrative accounts, referred to as System accounts or System userProfiles. These accounts are used to perform system administration and maintenance operations.

- The **DataCurator** account is used to perform system administration tasks.
- The **SystemUser** account is ordinarily 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

The GemStone/S 64 Bit extents come with the time zone of 'America/Los_Angeles' installed as the default.

If you are in another time zone, login to GemStone as SystemUser and execute `TimeZone installOsTimeZone` to install the time zone configured in the OS, or `TimeZone installNamedZone:` to install another time zone by the zoneinfo (Olson) name.

You may also edit the file `installtimezone.txt` in the `$GEMSTONE/upgrade` directory, and file it in as SystemUser.

GemStone User Accounts

All users who log into GemStone should have a GemStone user account. While possible, it is not recommended to do development as DataCurator.

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 using topaz, or tools included the GemBuilder for Smalltalk (GBS) product. See the *GemBuilder for Smalltalk Users's Guide* for information on the GUI tools in GBS.

Before login, Unix users should:

- Set the environment variable `GEMSTONE` to the *full pathname* (starting with a slash) of the GemStone/S 64 Bit 3.7.2 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.

For example:

```
os$ export GEMSTONE=/usr/lib64/gemstone/3.7.2
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 user's login shell initialization file.

Chapter

3 Upgrading from GemStone/S 64 Bit 3.3.x or later

This chapter describes how to upgrade an existing GemStone/S 64 Bit 3.3.x, 3.4.x, 3.5.x, or 3.6.x installation to GemStone/S 64 Bit version 3.7.2.

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 v3.7.2.

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 8.8.1 and 5.4.7 may be usable, but are not fully supported with v3.7.2. See Chapter 5 for details.

Keyfiles

You may use a keyfile from 3.7 or any v3.7.x version with v3.7.2. Keyfiles from 3.6.x and earlier are not valid with v3.7.2. For more on keyfiles, see “Set the GemStone Keyfile” on page 11.

To acquire a keyfile for version 3.7.2, 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 v3.7.2.

NOTE

The following instructions use the version number **3.7.1** *to refer to the version you are upgrading from, and version number* **3.7.2** *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 upgrade 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 3.7.2 kernel methods, and refer to the *Release Notes* for version 3.7.2 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 3.7.2.

Prepare for Upgrade

1. Install and configure GemStone/S 64 Bit 3.7.2

Install GemStone/S 64 Bit 3.7.2 to a new installation directory, separate from the installation directory for version 3.7.1, as described in Chapter 1, starting on page 7.

Configure GemStone/S 64 Bit 3.7.2 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 3.7.2, 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 3.7.1 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 3.7.1 system as a user with SessionAccess and SystemControl privileges, such as DataCurator, and halt all user activity on the repository.

```
topaz 1> exec System stopUserSessions %
```

4. Shut down the repository

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

stopstone *stone371*

where *stone371* is the name of the version 3.7.1 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 3.7.2 environment.

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

export GEMSTONE=*InstallDir372*
export PATH=$GEMSTONE/bin:$PATH
export upgradeLogDir=*tempDir*

where *InstallDir372* is the GemStone/S 64 Bit version 3.7.2 installation and *tempDir* is a temporary directory for which you have write permission.

6. Copy extent files

Copy your version 3.7.1 extent files into the location specified by the v3.7.2 configuration file option DBF_EXTENT_NAMES:

a. Identify the complete set of extent files that are used by your 3.7.1 stone. This can be found by examining the configuration file for the version 3.7.1 repository, looking for the last entry for DBF_EXTENT_NAMES.

b. The target location is the setting for `DBF_EXTENT_NAMES` in the version 3.7.2 installation. Copy each of these extent files to the target location.

For example:

```
cp dataDirFor371/extent0.dbf dataDirFor372
cp dataDirFor371/extent1.dbf dataDirFor372
cp dataDirFor371/extent2.dbf dataDirFor372/
```

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 3.7.2.

Perform the Upgrade

1. Start the Stone

Start the 3.7.2 Stone on the 3.7.1 extents you just copied:

```
startstone stoneName372
```

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 stoneName372
```

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 3.7.2 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 v3.7.1.

For example:

```
topaz 1> run
(AllUsers userWithId: 'SystemUser') password: '371Password'.
(AllUsers userWithId: 'GcUser') password: '371Password'.
(AllUsers userWithId: 'DataCurator') password: '371Password'.
System commitTransaction
%
```

where *371Password* is the account password used in version 3.7.1.

4. If you are upgrading from a 3.6.x version, recompile methods referencing instance of reimplemented class definitions

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

This step only applies if you are upgrading from 3.6.x or earlier.

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. This upgrade process is only tested and supported for upgrades from 3.5.3 and later.

For details, see Chapter 4, starting on page 31.

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 (GBJ), you must reinstall the appropriate version of these products into your repository at this time. Note that version 3.7.2 requires GBJ v3.2 or above, so GBJ upgrade is required.

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 3.7.2 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. Some older versions of GBS may be usable but are not fully supported with GemStone/S 64 Bit v3.7.2.

Chapter 5, 'Configuring GBS for GemStone/S 64 Bit' provides the supported versions of GBS for use on this platform. If your GBS clients run on a different platform than your GemStone server, refer to the *Installation Guide* for that platform.

Follow the instructions in that chapter for configuring the correct library for GBS.

Chapter

4 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 v3.7.2. 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 3. 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 3.
- 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 3.

1. Ensure that GemStone 3.7.2 is installed and your repository upgraded

You must first follow install version 3.7.2 and follow the instructions in Chapter 3, 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 v3.7.2 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 4.1 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 4.2 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 stoneName372
```

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 3 and complete the remaining upgrade steps.

Chapter

5 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 3.7.2.

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 v3.7.2.

In addition to using the appropriate version of GBS, you must use GemStone/S 64 Bit 3.7.2 client libraries with your GBS client application, to be able to log in to the v3.7.2 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

The following GBS versions are certified with v3.7.2 on Linux.

GemBuilder for Smalltalk v8.8.1

- **VisualWorks 9.4 (64-bit)**
  - RedHat Linux 9.4, 8.9 and 7.9; Ubuntu 24.04, 22.04 and 20.04
- **VisualWorks 9.3.1 (64-bit)**
  - RedHat Linux 9.4, 8.9 and 7.9; Ubuntu 24.04, 22.04 and 20.04

- **VisualWorks 9.1.1 (64-bit)**
  - RedHat Linux 9.4, 8.9 and 7.9; Ubuntu 24.04, 22.04 and 20.04
- **VisualWorks 8.7.2 (64-bit)**
  - RedHat Linux 9.4, 8.9 and 7.9; Ubuntu 24.04, 22.04 and 20.04

GemBuilder for Smalltalk v8.7.1

- **VisualWorks 9.1.1 (64-bit)**
  - RedHat Linux 9.4, 8.9 and 7.9; Ubuntu 24.04, 22.04 and 20.04

Earlier 8.7.x and 8.8.x versions may be compatible but have not been tested.

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, gslist, 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 gslist, 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.

Note that GemStone/S 64 bit v3.7.2 and later do not support 32-bit clients on Linux.

64 bit VisualWorks clients, RPC only

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

```
$GEMSTONE/lib/libgcirpc-3.7.2-64.so
$GEMSTONE/lib/libssl-3.7.2-64.so
$GEMSTONE/lib/libkrb5-3.7.2-64.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

Install or upgrade client smalltalk and GBS

To use GBS to login to GemStone/S 64 Bit v3.7.2, you must use a supported version of GBS, which must be loaded into a version of the client Smalltalk (VisualWorks) environment that supports this version of GBS. The certified configurations are listed on page 35.

See the *GemBuilder for Smalltalk Installation Guide* for v8.8.1 or v8.7.1 for installation instructions.

If you are upgrading, see the *GemBuilder for Smalltalk Release Notes* for the appropriate versions, for details on the changes.

Configure GBS to reference v3.7.2 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 v3.7.2 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.

64-bit VisualWorks clients, RPC logins only:

```
libgcirpc-3.7.2-64.so
```

64-bit VisualWorks clients, Linked and RPC logins:

```
libgbslnk-3.7.2-64.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.