
GemStone®

GemStone/S 64 Bit Installation Guide

For Linux on x86_64 Compatible Systems

Version 3.0.1

November 2011

vmware®

GEMSTONE[™]
.....S[™]64

INTELLECTUAL PROPERTY OWNERSHIP

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

This documentation, or any part of it, may not be reproduced, displayed, photocopied, transmitted, or otherwise copied in any form or by any means now known or later developed, such as electronic, optical, or mechanical means, without express written authorization from VMware, Inc.

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 VMware, Inc. 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 VMware, Inc.

This software is provided by VMware, Inc. 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 VMware, Inc. 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 © 1986-2011 VMware, Inc., and GemStone Systems, Inc. All rights reserved by VMware, Inc.

PATENTS

GemStone software is 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

VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions.

GemStone, **GemBuilder**, **GemConnect**, and the GemStone logos are trademarks or registered trademarks of VMware, Inc., previously of GemStone Systems, Inc., in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Sun, **Sun Microsystems**, and **Solaris** are trademarks or registered trademarks of Oracle and/or its affiliates. **SPARC** is a registered trademark of SPARC International, Inc.

HP, **HP Integrity**, and **HP-UX** are registered trademarks of Hewlett Packard Company.

Intel, **Pentium**, and **Itanium** are registered trademarks of Intel Corporation in the United States and other countries.

Microsoft, **MS**, **Windows**, **Windows XP**, **Windows 2003**, **Windows 7** and **Windows Vista** are registered trademarks of Microsoft Corporation in the United States and other countries.

Linux is a registered trademark of Linus Torvalds and others.

Red Hat and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.

SUSE is a registered trademark of Novell, Inc. in the United States and other countries.

AIX, **POWER5**, and **POWER6** are trademarks or registered trademarks of International Business Machines Corporation.

Apple, **Mac**, **Mac OS**, **Macintosh**, and **Snow Leopard** are trademarks of Apple Inc., in the United States and other countries.

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. All terms mentioned in this documentation that are known to be trademarks or service marks have been appropriately capitalized to the best of our knowledge; however, VMware 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.

VMware, Inc.
15220 NW Greenbrier Parkway
Suite 150
Beaverton, OR 97006

Preface

About This Documentation

This document explains how to install GemStone/S 64 Bit version 3.0.1, and how to upgrade from previous GemStone/S 64 Bit versions. These release notes are also available on the GemStone Technical Support website.

For information regarding new and modified features, please refer to the *GemStone/S 64 Bit Release Notes*.

Terminology Conventions

The term “GemStone” is used to refer to the server products GemStone/S 64 Bit and GemStone/S; the GemStone Smalltalk programming language; and may also be used to refer to the company, previously GemStone Systems, Inc., now a division of VMware, Inc.

Technical Support

GemStone Website

<http://support.gemstone.com>

GemStone’s Technical Support website provides a variety of resources to help you use GemStone products:

- ▶ **Documentation** for released versions of all GemStone products, in PDF form.
- ▶ **Downloads and Patches**, including past and current versions of GemBuilder for Smalltalk.
- ▶ **Bugnotes**, identifying performance issues or error conditions you should be aware of.
- ▶ **TechTips**, providing information and instructions that are not otherwise included in the documentation.
- ▶ **Compatibility matrices**, listing supported platforms for GemStone product versions.

This material is updated regularly; we recommend checking this site on a regular basis.

Help Requests

You may need to contact Technical Support directly, if your questions are not answered in the documentation or by other material on the Technical Support site. Technical Support is available to customers with current support contracts.

Requests for technical support should be submitted online or by telephone. We recommend you use telephone contact only for serious requests that require immediate attention, such as a production system down. The support website is the preferred way to contact Technical Support.

Website: <http://techsupport.gemstone.com>

Email: techsupport@gemstone.com

Telephone: (800) 243-4772 or (503) 533-3503

When submitting a request, please include the following information:

- ▶ Your name, company name, and GemStone server license number.
- ▶ The versions of all related GemStone products, and of any other related products, such as client Smalltalk products.
- ▶ The operating system and version you are using.
- ▶ A description of the problem or request.
- ▶ Exact error message(s) received, if any, including log files if appropriate.

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

24x7 Emergency Technical Support

GemStone Technical Support 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 your GemStone account manager.

Training and Consulting

Consulting is available to help you succeed with GemStone products. Training for GemStone software is available at your location, and training courses are offered periodically at our offices in Beaverton, Oregon. Contact your GemStone account representative for more details or to obtain consulting services.

Contents

Chapter 1. Installing GemStone/S 64 Bit Version 3.0.1	7
Review the Installation Procedure	7
Check the System Requirements	8
Prepare for Installation	10
Set the Environment	11
Create the GemStone Key File	12
Verify TCP/IP	12
Define the NetLDI Service	12
Run the Installation Script.	13
Decisions to Make During Installation.	14
Change System Passwords	16
Install the default TimeZone	16
Configure Users	16
Have Users Execute gemsetup	16
What Next?	17
Chapter 2. Upgrading from previous GemStone/S 64 Bit 3.x versions	19
Upgrade Strategy.	19
Upgrade Procedure	19
Prepare for Upgrade	20
Perform the Upgrade.	21
Restore Your Site-Specific Settings and Back Up the Repository	22

Chapter 3. Converting from previous GemStone/S 64 Bit 2.4.x versions 25

Upgrade Strategy	26
Upgrade Procedure	26
Prior to Upgrade in existing application	26
Prepare for Upgrade	27
Perform the Upgrade	29
Restore Your Site-Specific Settings	29
Post-upgrade Application Code Modifications	30
Backup repository and configure GBS	32

Chapter 4. Configuring GBS for GemStone/S 64 Bit 33

VisualWorks versions and platforms	34
GemStone/S 64 Bit 3.0.1 Windows Client Installation	34
Installing GemStone/S 64 Bit 3.0.1 Libraries onto Windows Clients	35
Installing GemStone/S 64 Bit 3.0.1 Libraries onto Unix Clients	36

Installing GemStone/S 64 Bit Version 3.0.1

This chapter describes the procedure for installing GemStone/S 64 Bit version 3.0.1 on a single machine. We recommend that you set up GemStone this way initially to ensure that all the pieces work together. At the end of this chapter, we suggest refinements you might want to make, such as running GemStone in a network configuration.

NOTE

If you are upgrading to this release from a previous version of GemStone/S 64 Bit, follow the instructions in the appropriate chapter, either Chapter 2, 'Upgrading from previous GemStone/S 64 Bit 3.x versions', or Chapter 3, 'Converting from previous GemStone/S 64 Bit 2.4.x versions'.

Adjust the installation to meet your specific needs. The topic "What Next?" on page 17 provides references to procedures and related information in the *System Administration Guide for GemStone/S 64 Bit*.

Review the Installation Procedure

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

- ▶ Check the System Requirements 8
- ▶ Prepare for Installation 10
- ▶ Set the Environment 11
- ▶ Create the GemStone Key File 12
- ▶ Verify TCP/IP 12
- ▶ Define the NetLDI Service 12
- ▶ Run the Installation Script 13
- ▶ Change System Passwords. 16
- ▶ Install the default TimeZone. 16

- ▶ Configure Users 16

Check the System Requirements

Before you install GemStone/S 64 Bit, ensure that the following system requirements are satisfied. Systems meeting these requirements are suitable for installing GemStone/S 64 Bit and beginning development, but additional system resources may be necessary to support large applications.

Platform

- ▶ System with an x86_64-compatible processor.

RAM

- ▶ At least 1 GB Physical RAM installed.
1 GB is sufficient for only for very small systems. In most cases you will need much more, depending on the size of the cache and the number of Gem sessions.

Swap space

- ▶ Total swap space should be at least twice the RAM installed.

Disk space

- ▶ Space for the installed distribution files—you need approximately 450 MB for GemStone/S 64 Bit, and additional space for other products.
- ▶ Additional disk space as required for your repository.
- ▶ 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

- ▶ SuSE Linux ES 10 SP1 (kernel version 2.6.16.46-0.12-smp and glibc-2.4-31.30), Red Hat Linux ES 5.0 (kernel version 2.6.18-8.el5 and glibc-2.5-12), or Red Hat Linux ES 5.5 (kernel version 2.6.18-194.el5 and glibc-2.5-49)
- ▶ The kernel must be configured to support shared memory and semaphores. See your operating system documentation for further information. These requirements apply both to server nodes and to client nodes.

- a. The maximum shared memory segment should be set to a value larger than your desired Shared Page Cache size, and not more than 75% of your real memory size.

The shared memory sizes are configured in pages, with a base page size of 4KB. Note that the results of `ipcs` may be reported in kbytes.

For example, if you have 8192 MB of real memory:

$$8192 \text{ MB} * .75 = 6144 \text{ MB}$$

$$6144 \text{ MB} * 2^{20} = 6442450944 \text{ bytes}$$

$$6442450944 / 4K = 1572864$$

To set shared memory, you would append the following text to the `/etc/sysctl.conf` file. The setting of the shared memory size is read from this file during the boot process.

```
# Shared Memory setting for GemStone
kernel.shmall = 1572864
kernel.shmmax = 1572864
```

For more details, consult your Linux operating system documentation.

- b. You may need to increase the settings for semaphores. `semmsl` sets the maximum number of semaphores per id (per semaphore set). This parameter limits the number of GemStone sessions that can log in to a particular Stone and connect to its shared page cache. (Note that `semmsl` ends with a lowercase L, not a digit.)

On the Stone's node, this parameter must provide **two** semaphores for each user who will log in to that Stone from any node plus an overhead of **four**. In distributed systems, nodes that have only user sessions must provide **two** semaphore for each user session on that node plus an overhead of **one**.

The number of semaphores actually requested for a particular shared page cache depends on the GemStone configuration file read by the process that starts the cache and is $(SHR_PAGE_CACHE_NUM_PROCS * 2) + 1$.

- c. `semms` sets the total number of semaphores in the system. This parameter limits the total number of GemStone sessions on a node. Keep this setting \geq `semmsl` (above).
- d. A file descriptor limit of 1024 is adequate for up to about 500 GemStone users. Each user session requires two file descriptors, and others are needed for extents, transaction logs, and overhead. Consequently, you may need to set the `nfile` parameter to a larger value, such as 2048. *Use caution and increase the default setting only when necessary because doing so can have system side effects.*

GemStone/S 64 Bit includes configuration parameters that may be modified from the default values depending on the requirements of your system. For more information about these configuration options, see Appendix A of the *System Administration Guide for GemStone/S 64 Bit*.

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 as part of a consistency check. A system time earlier than the time at which the last checkpoint was written may be taken as an indication of corrupted data and prevent GemStone from starting. The time comparisons use GMT.

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 operating system default may not be appropriate for this purpose. For further information, refer to your operating system documentation.

C/C++ Compiler

- ▶ g++ (GCC) 4.1.2 20070115 (prerelease)

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. This compiler is required only for development work, not for execution.

Debugger

- ▶ GNU gbd 6.6

A C debugger can be useful to allow problem analysis by GemStone consulting or Technical Support. It also may allow you to debug your C user actions. It is not required for GemStone execution.

Prepare for Installation

Perform the following steps to prepare the machine to receive the GemStone/S 64 Bit software. Although most steps require root login, we recommend that you perform the initial step as the GemStone administrator.

These are the portions of the system that are affected by the installation of GemStone:

`/dev/raw`

Optional raw partitions for repository extents and transaction logs.

`/etc/services`

Internet services database, for NetLDI name lookup.

`/InstallDir/GemStone64Bit3.0.1-x86_64.Linux`

Location of the object server software.

`/opt/gemstone`

Default location for server lock files, host name id file, and log files for GemStone network servers (NetLDIs). See the *System Administration Guide* for more information.

`/usr/gemstone`

Alternative location for lock and log files, for compatibility with previous products; `/opt/gemstone` is created unless `/usr/gemstone` already exists. See the *System Administration Guide* for more information.

1. As the GemStone administrator, log in to a machine that has adequate resources to run GemStone and that owns the disk on which you are going to install the GemStone files.

NOTE

Do not copy the files as root. The ownerships that were in effect when the distribution media was created are preserved, and this might result in file permission errors for users at your site.

2. Determine that adequate swap space is available:

```
% cat /proc/swaps
```

3. Check the free disk space and determine the disk drive and partition on which you will install the GemStone software.

To list all disk partitions, along with the amount of free space in each partition:

```
% df
```

We recommend that you avoid choosing either an NFS-mounted partition or one containing UNIX swap space for the initial installation. Mounted partitions can result in executables running on the wrong machine and in file permission problems. Existence of swap space on the same drive can dramatically slow GemStone disk accesses.

4. GemStone/S 64 Bit is provided as a zipped archive file with a name similar to `GemStone64Bit3.0.1-x86_64.Linux.zip`.

Select an installation directory, *InstallDir*, and move the distribution file to the directory location in which GemStone will be installed. Make this directory the current working directory.

5. Unzip the distribution file using `unzip`. For example:

```
% unzip GemStone64Bit3.0.1-x86_64.Linux.zip
```

The *InstallDir* now contains a GemStone directory with a name similar to `GemStone64Bit3.0.1-x86_64.Linux`.

In addition to several subdirectories, this directory also contains two text files: `PACKING`, which lists all of the GemStone files, and `version.txt`, which identifies this particular product and release of GemStone.

6. Log in as root.

Although you can complete the installation as a non-root user, we do not recommend this. During installation, GemStone system security is established through file permissions and process attributes. To ensure that the installation is successful, **you must install as root**. If you later decide to change the security of your GemStone system, see Chapter 1 of the System Administration Guide for GemStone/S 64 Bit, which explains the concept of GemStone server file permissions and how to change them.

Set the Environment

Perform the following steps to properly configure the operating environment.

1. Set the environment variable `GEMSTONE`.
 - a. If more than one installation of any GemStone/S product resides on this machine, check for existing GemStone environment variables:

```
% env | grep GEM
```

All GemStone environment variables are displayed.

- b. If any environment variables exist, you must specifically unset each one.

C shell:

```
% unsetenv GEMSTONE GEMSTONE_SYS_CONF \
  GEMSTONE_EXE_CONF GEMSTONE_LOG GEMSTONE_LANG
```

Bourne or Korn shell:

```
$ unset GEMSTONE GEMSTONE_SYS_CONF GEMSTONE_EXE_CONF \
  GEMSTONE_LOG GEMSTONE_LANG
```

- c. Set the environment variable GEMSTONE to the *full pathname* (starting with a slash) of your new GemStone installation directory.

C shell:

```
% setenv GEMSTONE InstallDir/GemStone64Bit3.0.1-x86_64.Linux
```

Bourne or Korn shell:

```
$ GEMSTONE=InstallDir/GemStone64Bit3.0.1-x86_64.Linux
```

```
$ export GEMSTONE
```

Create the GemStone Key File

To run GemStone, you must have a key file. Keyfile information is provided by GemStone Contract Administration. Contact your GemStone account representative or keyfiles@gemstone.com if you have questions about your keyfile, or if you need an evaluation keyfile.

1. Change the permissions on the directory `$GEMSTONE/sys` so that you can create the file:

```
% cd $GEMSTONE/sys
```

```
% chmod 755 .
```

2. Using a text editor, create the key file, pasting in the provided keyfile information:

```
$GEMSTONE/sys/gemstone.key.
```

3. Change the file and directory permissions so that they are no longer writable:

```
% chmod 555 gemstone.key
```

```
% chmod 555 .
```

Verify TCP/IP

To run GemStone, TCP/IP must be functioning, even if your machine is not connected to a network.

1. Verify that TCP/IP networking software is functioning (1 is the number 1):

```
% /bin/ping hostname
```

where *hostname* is the name of your machine. If **ping** responds with statistics, TCP/IP is functioning.

Define the NetLDI Service

The NetLDI service, by default `gs64ldi`, should be defined in your TCP/IP network database. This is required for certain kinds of local sessions, or any client on a remote machine, to log into GemStone. For clients on remote machines, the same NetLDI service name and port number must be defined on the remote machine.

NOTE

If you are upgrading from a previous version, you might need to keep the NetLDI for that version running. You may need to select a distinct name and port for the NetLDI for GemStone/S 64 Bit 3.0.1, rather than using the default `gs64ldi`.

1. Determine which TCP/IP network database (local or NIS) is in use:

```
% ypwhich
```

If the program is missing or you see an error message when you run it, you can assume that your machine is using a local copy of the TCP/IP network database instead of a copy provided by NIS.

If NIS is running, have your UNIX system administrator perform the following steps.

2. Determine whether the `gs64ldi` service is already defined.

If you are using NIS:

```
% ypcat services | grep gs64ldi
```

If your machine is using a local copy of the TCP/IP network database:

```
% grep gs64ldi /etc/services
```

If `gs64ldi` is defined, skip the rest of this procedure and continue with the installation at “Run the Installation Script” on page 13.

If it is not defined, continue performing this procedure.

3. Add an entry similar to the following to the network database:

```
gs64ldi 50377/tcp #GemStone/S 64 Bit 3.0.1
```

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>).

4. If NIS is running, propagate the change to the network database to the rest of the network.
5. If NIS is not running, but several machines will be running GemStone, have the UNIX system administrator update the network database for each machine. Note that the port number must be the same for every machine.

Run the Installation Script

Invoke the installation script from the `install` subdirectory:

```
% cd $GEMSTONE/install  
% ./installgs
```

`installgs` is an interactive script that analyzes your system configuration and makes suggestions to guide you through installing GemStone on your machine.

NOTE

You can usually terminate execution of the installation script with `Ctrl-C` without risk to your files. When it is not safe to do so, the message `Please do not interrupt` appears on the screen. If this happens, wait for the message now it is OK to interrupt before you interrupt the script. You can run the script again from the beginning as many times as necessary.

Decisions to Make During Installation

During installation, you are asked several questions. The entire installation dialog is not reproduced here, but the main points are addressed. Some questions may not be asked, depending on answers to previous questions.

Whenever you are asked to answer “yes” or “no,” answer with **y** or **n**. When the script offers a default answer in square brackets (such as “`[y]`”), press Enter to accept the default.

Do you want the installation script to set up directories for server lock files and NetLDI logs?

The default location for server lock files and NetLDI log files is `/opt/gemstone`, although for compatibility with earlier products `/usr/gemstone` is used only if it exists. If the environment variable `GEMSTONE_GLOBAL_DIR` is defined to point to a valid directory, this overrides the default server lock files and log file location; however, all Gemstone processes that will interact on this machine must have this environment variable set to the same directory.

If these directories do not exist, the installation script offers to create `/opt/gemstone` and the subdirectories `locks` and `log`. Then, the script offers to set access (`770`) to these directories.

If you answer **no** to creating the directories, you must create them (or provide a symbolic link) before starting the server.

Do you want the installation script to set the owner and group for all the files in the GemStone distribution?

If you answer **yes**, the script will prompt you for the owner and group you want to use. Refer to Chapter 1 of the *System Administration Guide for GemStone/S 64 Bit* for more information about setting owner and group permissions.

If you answer **no**, the permissions will remain the same as when the files were extracted from the distribution media.

Do you want the installation script to protect the repository file?

The default, which we recommend, gives only the owner read and write access (`600`) through ordinary UNIX commands. Other users can read and write the repository through a GemStone session. If you choose not to protect the repository, the `setuid` bit is cleared from all executables, which causes them to run under ownership of the user who invokes them.

Default: Set the repository permission to `600`, and leave the `setuid` bit applied.

Allow NetLDI to Run as Root?

Do you want the installation script to allow non-root users to start a NetLDI that runs as root?

The NetLDI is a network server that permits remote processes to interact with the repository. 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.

- ▶ To run NetLDIs as root, accept the default “yes” response. Ownership of the NetLDI executable is changed to root, and the setuid bit is set. Any GemStone user will be able to start a NetLDI process that is accessible to all GemStone users because it will always run as root. For certain services, users will need to authenticate themselves by supplying a password. Alternatively, answer “no” but log in as root before starting the NetLDI.

If the NetLDI uses a port number less than 1024, it must run as root.

- ▶ To run NetLDIs in guest mode with a captive account, answer “no” to the prompt, because those modes are not permitted if the NetLDI runs as root. “Guest mode” means that GemStone users do not have to supply a UNIX password to use NetLDI services. The “captive account” is an account that owns all processes the NetLDI starts; typically, it is the GemStone administrative account that owns the files. You must start the NetLDI while logged in as that account.

Default: Change ownership of the `netldi` executable to root, and set its setuid bit.

Set up an Extent?

Do you want the installation script to set up an extent now?

GemStone is distributed with a read-only copy of the initial repository in `$(GEMSTONE)/bin/extent0.dbf`. Before you can start GemStone, this file must be copied to a suitable location and made writable. The script offers to copy the file to its default location of `$(GEMSTONE)/data`.

If you are a new GemStone user, we recommend that you answer **y**. If you are an existing GemStone user, you might prefer to answer **n**, then copy the extent to a different location yourself. (If you choose a location other than the default, you must edit your configuration file before starting GemStone. For information, see the *System Administration Guide for GemStone/S 64 Bit*.)

Default: Place a writable copy of `extent0.dbf` in `$(GEMSTONE)/data`.

Start a NetLDI?

Do you want the installation script to start a NetLDI?

If you prefer, you can start these processes manually at any time.

Almost every host needs a NetLDI. You must start a NetLDI when the Stone repository monitor or Gem session processes will run on this machine.

You can start a NetLDI that runs as root by answering **yes** to this prompt and the confirmation that follows. However, if you want to start the NetLDI in guest mode with a captive account, you must do that after completing the installation. For more information

about guest mode with captive account, see Chapter 3 of the *System Administration Guide for GemStone/S 64 Bit*.

Default: Do not start a NetLDI at this time.

Start an Object Server?

As root, you cannot start an object server, but the script offers to start one as another user. You will start the server later in the installation, so answer **no**.

Default: Do not start an object server at this time.

Log out as root

Log out as user root. The rest of the installation is done as the GemStone administrative user.

Change System Passwords

After installing GemStone/S 64 Bit, you must change the passwords for the three administrative users: DataCurator, SystemUser, and GcUser. (The initial password for each is `swordfish`.) 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. Access to each of these accounts should be restricted.

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.

Configure Users

Chapter 6 of the *System Administration Guide for GemStone/S 64 Bit* tells you how to change the passwords and set up accounts within GemStone for other GemStone users.

In addition to a GemStone user account, user sessions must define their environment.

Have Users Execute gemsetup

The directory `$GEMSTONE/bin` contains two files, `gemsetup.sh` and `gemsetup.csh`, to help set a user's environment. These files define the GemStone environment for users by modifying the `PATH` and `MANPATH` variables to include `$GEMSTONE/bin` and `$GEMSTONE/doc`, respectively.

After GemStone/S 64 Bit 3.0.1 has been installed, you should notify each GemStone user of the installation and explain how to use the `gemsetup` files.

NOTE

This procedure applies to users ONLY. Each user must perform this procedure before running GemStone.

1. Set the environment variable `GEMSTONE` to the *full pathname* (starting with a slash) of the GemStone/S 64 Bit 3.0.1 directory.

C shell:

```
% setenv GEMSTONE InstallDir/GemStone64Bit3.0.1-x86_64.Linux
```

Bourne or Korn shell:

```
$ GEMSTONE=InstallDir/GemStone64Bit3.0.1-x86_64.Linux  
$ export GEMSTONE
```

2. Invoke the script `gemsetup`.

C shell:

```
% source $GEMSTONE/bin/gemsetup.csh
```

Bourne or Korn shell:

```
$ . $GEMSTONE/bin/gemsetup.sh
```

3. If you will use GemStone frequently, consider adding to your login shell's initialization file (`.cshrc` or `.profile`) the environment variable `GEMSTONE` and the command `gemsetup`. This way, the GemStone environment is automatically configured every time you log in or create a login shell. It overrides any other GemStone path settings.

NOTE

If you use the Korn shell and your `.profile` contains commands that are not POSIX-compliant, you might encounter errors when a shell is initialized. To remedy this situation, place the non-compliant commands within a conditional, such as the following:

```
hash -r 2>/dev/null  
status=$?  
if [ $status -ne 0 ]; then  
    # Place Korn-shell-specific initialization here  
fi
```

What Next?

This chapter has guided you through installation of GemStone/S 64 Bit 3.0.1 in an initial configuration that is sufficient to create a basic repository and begin setting up user accounts. The objective has been to get a simple, default configuration up and running.

You might consider performing the following tasks:

- ▶ To modify the initial object server configuration to one that is more efficient for your particular needs, refer to Chapter 1 of the *System Administration Guide for GemStone/S 64 Bit*. This chapter contains sample configurations, from small to large, and also contains detailed information about how to tailor these configurations to your own system.

- ▶ To modify the configuration of Gem session processes and to ensure that users have the necessary permissions to access the shared page cache and the extents, refer to Chapter 2 of the *System Administration Guide for GemStone/S 64 Bit*.
- ▶ If you are going to operate in a network environment, Chapter 3 of the *System Administration Guide for GemStone/S 64 Bit* has additional information about the GemStone network server (NetLDI), how to handle user authentication, how to share software over the network, and how to set up some common configurations.
- ▶ To start and stop the GemStone object server, refer to instructions in Chapter 4 of the *System Administration Guide for GemStone/S 64 Bit*.

Upgrading from previous GemStone/S 64 Bit 3.x versions

This chapter describes how to upgrade an existing GemStone/S 64 Bit 3.0 installation to GemStone/S 64 Bit version 3.0.1.

To upgrade from GemStone/S 64 Bit version 2.4.x, which requires conversion, see Chapter 3, 'Converting from previous GemStone/S 64 Bit 2.4.x versions'. To upgrade from earlier versions of GemStone/S 64 Bit, or from 32-bit GemStone/S, you must first upgrade or convert to GemStone/S 64 Bit version 2.4.x.

For applications using GemConnect or Gembuilder for Java, these products will need to be reinstalled following the upgrade process.

Upgrade Strategy

We recommend that you perform the upgrade twice: first a pilot upgrade and then the production upgrade. With this strategy, you can keep your production system running while you familiarize yourself with the upgrade process.

Upgrade Procedure

The following list summarizes the steps necessary to perform the upgrade to GemStone/S 64 Bit version 3.0.1.

- ▶ Prepare for Upgrade 20
- ▶ Perform the Upgrade 21
- ▶ Restore Your Site-Specific Settings and Back Up the Repository 22

Prepare for Upgrade

Perform the following steps to prepare for the upgrade.

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

You will need to carefully compare these changes with GemStone/S 64 Bit 3.0.1 kernel methods to determine whether your changes are still necessary or appropriate.

2. Install GemStone/S 64 Bit 3.0.1 to a new installation directory, separate from the installation directory for version 3.0, as described in Chapter 1 of this Installation Guide.
3. Configure GemStone/S 64 Bit 3.0.1 the way you expect to use it – that is, with the appropriate extent locations and sizes.
4. Ensure that adequate space is available for extents, transaction logs, and a backup during the upgrade. You must provide space for the following:
 - ▶ Your version 3.0 extents and transaction logs.
 - ▶ Your version 3.0.1 extents and transaction logs.
5. Log in to the version 3.0 system 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.

6. Halt all user activity on the repository you are going to upgrade:
 - a. Log in to Topaz as DataCurator.
 - b. Force all other users off the system:

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

CAUTION

You MUST file out any changes that you have made to the GemStone/S 64 Bit kernel classes in order to preserve these changes in version 3.0.1. Also, consider saving important modified files, such as configuration files, that will be overwritten during the upgrade.

7. File out any modifications or additions you made to GemStone/S 64 Bit version 3.0 kernel class methods by using the Topaz command **fileout**. For more information about fileout, see the *GemStone/S 64 Bit Topaz Programming Environment*.
8. You must now shut down the Stone:

```
% stopstone stone30
```

where *stone30* is the name of the version 3.0 stone on this machine. The repository must be cleanly shut down before it can be restarted under version 3.0.1

9. Set up the version 3.0.1 environment.

Set the environment variables required for the upgrade.

C shell:

```
% setenv GEMSTONE InstallDir301
% set path = ($GEMSTONE/bin $path)
% setenv upgradeLogDir tempDir
```

Bourne or Korn shell:

```
$ GEMSTONE=InstallDir301
$ export GEMSTONE
$ export PATH=$GEMSTONE/bin:$PATH
$ upgradeLogDir=tempDir
$ export upgradeLogDir
```

where *InstallDir301* is the GemStone/S 64 Bit version 3.0.1 installation and *tempDir* is a temporary directory for which you have write permission.

NOTE

Use a separate log directory for each repository you convert. A repository may contain multiple extents.

10. Copy your version 3.0 extent files into the location specified by the configuration file option `DBF_EXTENT_NAMES`:
 - a. Using a text editor, open the configuration file that the version 3.0 repository uses.
 - b. Locate the last occurrence of the option `DBF_EXTENT_NAMES`, and note its value, a list of `.dbf` files.
 - c. Copy each `.dbf` file to the noted location in the version 3.0.1 installation. For example:

```
% cp InstallDir30/data/extent0.dbf 301location
% cp InstallDir30/data/extent1.dbf 301location
% cp InstallDir30/data/extent2.dbf 301location
```

where *301location* is the location specified by `DBF_EXTENT_NAMES` in the configuration file that will be used in version 3.0.1.

11. 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 3.0.1. Transaction logs from earlier versions are not compatible with version 3.0.1. If the transaction log directories will be reused for version 3.0.1, any transaction logs should be deleted or copied elsewhere.

Perform the Upgrade

1. Start the 3.0.1 Stone on the 3.0 extents you just copied:


```
% startstone stoneName301
```
2. 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 stoneName301
```

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

3. 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 `$GEMSTONE/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.

Restore Your Site-Specific Settings and Back Up the Repository

1. Reinstall any other GemStone products that modify kernel classes.
If you use GemConnect, you must install it again at this time. Use the procedure in the installation guide for this product.
2. Log in to GemStone/S 64 Bit version 3.0.1 as DataCurator.
3. Change the password for SystemUser, which you changed to `swordfish` prior to the conversion, back to its previous value. Also, change the passwords for GcUser and DataCurator, which were reset by the conversion process, back to the version 3.0 value:

```
topaz 1> printit
(AllUsers userWithId: 'SystemUser') password: '30Password'.
(AllUsers userWithId: 'GcUser') password: '30Password'.
(AllUsers userWithId: 'DataCurator') password: '30Password'.
System commitTransaction
%
```

where `30Password` is the account password used in GemStone/S 64 Bit version 3.0.

The upgraded repository is now usable. Other users can log in to assist with the following steps.

4. If you have modified any kernel class methods of the previous version, carefully compare your changes with version 3.0.1 kernel methods to see whether your changes are still necessary or appropriate. If so, file in the changes and commit.

5. Create a full backup of the upgraded repository. For details, see the *System Administration Guide for GemStone/S 64 Bit*.
6. If you are using GBS clients, configure these to use the version 3.0.1 client libraries. See Chapter 4, 'Configuring GBS for GemStone/S 64 Bit' for details.

Converting from previous GemStone/S 64 Bit 2.4.x versions

This chapter describes how to upgrade an existing GemStone/S 64 Bit 2.4.x installation to GemStone/S 64 Bit version 3.0.1. If you are upgrading from version 3.0, which does not require conversion, see the instructions in Chapter 2, 'Upgrading from previous GemStone/S 64 Bit 3.x versions'.

GemStone/S 64 Bit version 3.0.1 supports upgrade from GemStone/S 64 Bit versions 2.4.4 and later only. If you are upgrading from an earlier version of GemStone/S 64 Bit, or from GemStone/S, you will first need to upgrade to version 2.4.4.x, following the instructions in the Installation Guide for v2.4.

Upgrade of Seaside/GLASS repositories is not yet available with for v3.0.1.

Version 3.0 and later have changes from v2.x to compiled methods and all new bytecodes to support native code. As a result, the upgrade process requires that you file in all application source code so it can be recompiled. There are a number of changes you may have to make to application source code and to persistent instances; please review the upgrade process carefully.

New keyfiles are required with GemStone/S 64 Bit version 3.0 and later. Keyfiles for GemStone/S 64 Bit 2.x will not work with version 3.0. If you need a keyfile for version 3.0.1, contact GemStone Technical Support, or email keyfiles@gemstone.com.

If you are using GemBuilder for Smalltalk (GBS), you should also upgrade GBS to version 7.4, and update the client libraries that are used by GBS. See Chapter 4 on page 33 for using GBS with GemStone/S 64 Bit 3.0.1, and instructions on installing updated client libraries.

For applications using GemConnect or Gembuilder for Java, updated versions of these products will also need to be reinstalled following the upgrade process.

Upgrade Strategy

We recommend that you perform the upgrade twice: first a pilot upgrade and then the production upgrade. With this strategy, you can keep your production system running while you familiarize yourself with the upgrade process.

Upgrade Procedure

The following list summarizes the steps necessary to perform the upgrade to GemStone/S 64 Bit version 3.0.1.

- ▶ Prepare for Upgrade. 27
- ▶ Perform the Upgrade 29
- ▶ Restore Your Site-Specific Settings. 29

NOTE

The following instructions use the version number 2.4.4.7 to represent any version from which upgrade/conversion is supported. The procedure is the same regardless of which version you have.

Prior to Upgrade in existing application

1. If you do not already have source code for your application stored externally to the GemStone repository in a code management system, file out all application source code using the Topaz command **fileout**. Filein of application code is required to recompile methods.
2. File out any modifications or additions you made to GemStone/S 64 Bit kernel class methods. You will need to carefully compare these changes with GemStone/S 64 Bit 3.0.1 kernel methods to determine whether your changes are appropriate. GemStone kernel classes have extensive changes for version 3.0. See the Release Notes for version 3.0 for details of the changes, and read these carefully before filing 2.x code modifications into your version 3.0.1 repository.
3. If your application uses the ScaledDecimal class, determine if you wish to continue to use this class under its new name FixedPoint, or if you want to use the new implementation of ScaledDecimal. If you wish to continue to use the old ScaledDecimal with its new name, you will need to modify your source code fileout so that all references to ScaledDecimal are changed to refer to FixedPoint, and all uses of the s ScaledDecimal literal notation are to the FixedPoint p literal notation.

Note that you will also have to manually updated stored instances of ScaledDecimal, which is done following upgrade.

- If you have indexes that include instance of any of the following classes, these indexes will need to be removed and rebuilt in version 3.0 to update information within the indexing structures.

```

DoubleByteString
DoubleByteSymbol
Float
LargeNegativeInteger
LargePositiveInteger
QuadByteString
ScaledDecimal (if you will be using the new ScaledDecimal implementation)
SmallFloat

```

If unsure, it is safer to remove the index rather than to risk incorrect indexed query results. You may remove these indexes prior to upgrade, or as part of application filein after upgrade.

- If you use the Class description variable, or if you have defined class comments using GBS browsers, which store the class comment in the description field, you will need to save the contents elsewhere. Due to structural changes in classes, the conversion process will remove any data stored here. Following upgrade, you can restore the description using `Class>>description:`. Note that class comments are now returned as Strings by the class method `comment`.

Prepare for Upgrade

Perform the following steps to prepare for the upgrade.

- Install GemStone/S 64 Bit 3.0.1 to a new installation directory, separate from the installation directory for version 2.4.4.7, as described in Chapter 1 of this Installation Guide.
- Configure GemStone/S 64 Bit 3.0.1 the way you expect to use it – that is, with the appropriate extent locations and sizes.
- Ensure that adequate space is available for extents, transaction logs, and a backup during the upgrade. You must provide space for the following:
 - ▶ Your version 2.4.4.7 extents and transaction logs.
 - ▶ Your version 3.0.1 extents and transaction logs.
- Log in to the version 2.4.4.7 system 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.

5. Halt all user activity on the repository you are going to upgrade:

- a. Log in to Topaz as DataCurator.
- b. Force all other users off the system:

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

6. You may now shut down the Stone:

```
% stopstone stone2447
```

where *stone2447* is the name of the version 2.4.4.7 stone on this machine. It is preferred, but not required, that the repository be cleanly shut down before it is restarted under version 3.0.1.

7. Set up the version 3.0.1 environment.

Set the environment variables required for the upgrade.

C shell:

```
% setenv GEMSTONE InstallDir301
% set path = ($GEMSTONE/bin $path)
% setenv upgradeLogDir tempDir
```

Bourne or Korn shell:

```
$ GEMSTONE=InstallDir301
$ export GEMSTONE
$ export PATH=$GEMSTONE/bin:$PATH
$ upgradeLogDir=tempDir
$ export upgradeLogDir
```

where *InstallDir301* is the GemStone/S 64 Bit version 3.0.1 installation and *tempDir* is a temporary directory for which you have write permission.

NOTE

Use a separate log directory for each repository you convert.

8. Copy your version 2.4.4.7 extent files into the location specified by the configuration file option `DBF_EXTENT_NAMES`:

- a. Using a text editor, open the configuration file that the version 2.4.4.7 repository uses.
- b. Locate the last occurrence of the option `DBF_EXTENT_NAMES`, and note its value, a list of `.dbf` files.
- c. Copy each `.dbf` file to the noted location in the version 3.0.1 installation. For example:

```
% cp InstallDir2447/data/extent0.dbf 301location
% cp InstallDir2447/data/extent1.dbf 301location
% cp InstallDir2447/data/extent2.dbf 301location
```

where *301location* is the location specified by `DBF_EXTENT_NAMES` in the configuration file that will be used in version 3.0.1.

If version 3.0.1 will run on a platform with a different byte ordering than the version 2.4.4.7 repository, or if you are using raw partitions, you will need to use copydbf.

9. 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 3.0.1. If the transaction log directories will be reused for version 3.0.1, any transaction logs should be deleted or copied elsewhere.

Perform the Upgrade

1. Conversion is done using the `-C` flag to `startstone`. Perform the conversion on the 2.4.4.7 extents you just copied:

```
% startstone -C stoneName301
```

2. 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 stoneName301
```

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

3. 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 `$GEMSTONE/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.

Restore Your Site-Specific Settings

1. Log in to GemStone/S 64 Bit version 3.0.1 as DataCurator.
2. Change the password for SystemUser, which you changed to `swordfish` prior to the conversion, back to its previous value. Also, change the passwords for GcUser and

DataCurator, which were reset by the conversion process, back to the version 2.4.4.7 value:

```
topaz 1> printit
(AllUsers userWithId: 'SystemUser') password: '2447Password'.
(AllUsers userWithId: 'GcUser') password: '2447Password'.
(AllUsers userWithId: 'DataCurator') password: '2447Password'.
System commitTransaction
%
```

where *2447Password* is the account password used in GemStone/S 64 Bit version 2.4.4.7.

The upgraded repository is now usable. Other users can log in to assist with 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 into your repository at this time. Version 3.0 requires new versions of these products; version 2.2.3 of GemConnect or version 3.0.1 of GemBuilder for Java. Contact GemStone Technical Support for more information. To install, use the procedure in the installation guide for the specific product.

2. Recompile application code by filein

The upgrade process requires filein of all customer code, in order to recompile to the new bytecodes and classes introduced in version 3.0. You should have fileins available, from step 1 on page 26.

Before filing in application code, you must verify your code does not include square-bracket Array constructors. If your application code contains Array constructors using the syntax `#[a, b, c]`, this is not valid syntax for version 3.0. In v3.0, you must use the syntax `{ a . b . c }` for Array constructors.

To allow filein of application code that includes the square bracket Array constructors, in the gem that will perform the filein, prior to filein, execute:

```
System configurationAt:#GemConvertArrayBuilder put: true
```

This allows code including the old Array constructor syntax to be compiled; the resulting compiled method will be correct, and its source code will be updated to the correct syntax. This configuration variable is runtime only, so it is not necessary to unset it manually.

Now, file in all development and application code. Verify that errorcount is 0, and commit.

3. Convert persistent blocks

ExecutableBlocks have been reimplemented in version 3.0; persistent instances of ExecutableBlocks are not usable in version 3.0. Executable blocks in source code are recompiled during application filein.

a. Application-specific stored blocks

If your application stores persistent blocks, you will have to locate and recompile all such blocks before they can be executed.

b. SortBlocks in SortedCollections

To convert the sortBlocks in persistent SortedCollections in your application, you may run the `postconv` script. This script has optional flags to specify stone name and the number of sessions to use.

```
postconv [-c <numCacheWarmerGems>] [-h] [-s <stoneName>] [-r]
[-n <numberOfSessions>] [-t <tempObjCacheSize>] [-u <userId>]
```

`-c <numCacheWarmerGems>` specifies the number of cache warmer threads in a single gem to load the object table into the shared cache before starting post-conversion. If not specified, no cache warming is done.

`-h` prints this usage information.

`-s <stoneName>` sets the name of the running stone to scan; if this option is not used, the script uses **gs64stone**.

`-n <numberOfSessions>` specifies the number of parallel sessions which will convert the instances of SortedCollection and its subclasses. By default, use one session.

`-r` specifies to reuse an existing version of `$upgradeLogDir/AllSortedCollections.bms`, if it exists. This file contains the OOPs of all instances of SortedCollections and its subclasses. By default, the existing file is deleted and a new one created.

`-t <tempObjCacheSize>` sets the size of GEM_TEMPOBJ_CACHE_SIZE in KB; by default, 20000.

`-u <userId>` is the UserId whose SymbolList includes all subclasses of SortedCollection, for which instance's sortBlocks will be converted. If not specified, defaults to SystemUser

For example,

```
% postconv -s stoneName301
```

The `postconv` script will write progress messages to stdout. When it completes, it will report:

```
postconv[INFO]: Congratulations! NNN SortedCollections were
successfully converted. No errors were detected.
```

4. Convert FixedPoint instances

If for step 3 on page 26, you decided to continue to use the old ScaledDecimal class, now named FixedPoint, do not perform this step.

If your 2.4.4.7 application included instances of ScaledDecimal, following conversion to 3.0.1 they are now instances of FixedPoint. If you intend to use the new ScaledDecimal implementation rather than FixedPoint, you must perform a manual step to convert each instance to a new ScaledDecimal. To do this, find all instances of FixedPoint and execute code similar to the following:

```
newScaledDecimal := aFixedPoint asScaledDecimal.
newScaledDecimal become: aFixedPoint.
```

Verify that errorcount is 0, and commit.

5. Rebuild indexes, if necessary.

If your application includes indexes on instances of kernel classes which have new implementations in version 3.0, these indexes must be rebuilt to update internal information in the Btree indexing structure, otherwise query results may be incorrect.

See step 4 on page 27 for more information.

6. Convert reimplemented classes

Optionally, you may wish to convert instances of reimplemented classes. Instances of Float, SmallFloat, LargePositiveInteger, LargeNegativeInteger, DoubleByteString, DoubleByteSymbol or QuadByteString in your version 2.4.4.7 repository are not converted during conversion to 3.0. The instances auto-migrated to the new class each time they are faulted into the VM.

To avoid this overhead, you can perform a listInstances of the classes ObsLargePositiveInteger, ObsLargeNegativeInteger, ObsDoubleByteString, ObsDoubleByteSymbol, ObsQuadByteString, ObsFloat, and ObsSmallFloat, sending #convert to each instance, and committing.

7. Restore Class descriptions

If you saved Class descriptions from your version 2.4.4.7 repository, you can restore them using Class >> description:.

8. 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 3.0.1 to see whether your changes are still necessary or appropriate. Carefully review the Release Notes for v3.0 and v3.0.1 as well as examining code in the image. If your changes are still applicable, file in the changes and commit.

Backup repository and configure GBS

1. Create a full backup of the upgraded repository. For details, see the *System Administration Guide for GemStone/S 64 Bit*.
2. If you are using GBS clients, configure these to use the version 3.0.1 client libraries. See Chapter 4, 'Configuring GBS for GemStone/S 64 Bit' for details.

Configuring GBS for GemStone/S 64 Bit

This chapter describes how to configure your GemBuilder for Smalltalk (GBS) application to run with GemStone/S 64 Bit version 3.0.1.

NOTE

GemStone/S 64 Bit version 3.0.1 requires the use of GBS version 7.4. Earlier GBS versions will encounter problems with GemStone/S 64 Bit 3.0 and above.

See “VisualWorks versions and platforms” on page 34 for the versions of VisualWorks that may be used with GemStone/S 64 Bit v3.0.1.

In addition to using the appropriate version of GBS, you must use GemStone/S 64 Bit 3.0.1 client libraries to be able to log in to the GemStone/S 64 Bit 3.0.1 server. Applications using GBS version 7.4 can log in to either GemStone/S or GemStone/S 64 Bit repositories (but not simultaneously), provided that the appropriate client libraries for each are used.

On UNIX and Linux platforms, GemStone/S 64 Bit provides both 64-bit libraries and 32-bit libraries. 32-bit VisualWorks processes cannot load 64-bit libraries; you must use 32-bit libraries with 32-bit VisualWorks installations. 64-bit and 32-bit libraries have identical names, and are provided in separate directories to distinguish them.

The following sections describes the procedure for installing client libraries and getting GBS to recognize them.

VisualWorks versions and platforms

The following tables list supported versions of GemBuilder for Smalltalk with the compatible versions of client Smalltalk and Operating System.

GBS version 7.4

The following VisualWorks and OS platforms and versions are supported with GemStone/S 64 Bit 3.0.1 and GemBuilder for Smalltalk version 7.4:

VW 7.8 32-bit with 32-bit 7.8 OE	VW 7.8 64-bit with 64-bit 7.8 OE	VW 7.7.1 32-bit with 32-bit 7.7.1 OE	VW 7.7.1 64-bit with 64-bit 7.7.1 OE
<ul style="list-style-type: none"> ▶ Windows 7, Windows 7 64-bit, Windows Vista, Windows 2003 Server, and Windows XP ▶ Solaris 10 on SPARC ▶ SuSE Linux ES 10 and RedHat Linux ES 5.5 and 5.0 	<ul style="list-style-type: none"> ▶ Solaris 10 on SPARC ▶ SuSE Linux ES 10 and Red Hat Linux ES 5.5 and 5.0 	<ul style="list-style-type: none"> ▶ Windows 7, Windows 7 64-bit, Windows Vista, Windows 2003 Server, and Windows XP ▶ Solaris 10 on SPARC ▶ SuSE Linux ES 10 and RedHat Linux ES 5.5 and 5.0 	<ul style="list-style-type: none"> ▶ Solaris 10 on SPARC ▶ SuSE Linux ES 10 and Red Hat Linux ES 5.5 and 5.0

GemStone/S 64 Bit 3.0.1 Windows Client Installation

To allow GemStone/S 64 Bit clients to connect from GBS applications and to run topaz on Windows, GemStone/S 64 Bit provides a separate Windows installation, with the name GemBuilder for C. This installation provides:

- ▶ The ability to run topaz on Windows, to login RPC to a GemStone/S 64 Bit server on UNIX.
- ▶ The ability to run gslis to get information about remote UNIX servers
- ▶ The client libraries to be used by GBS for RPC logins.

If you do not require topaz or gslis on Windows, you may wish to extract only the client libraries from the Windows installation, and move these files to a directory on the path or to the GBS installation. See “Installing GemStone/S 64 Bit 3.0.1 Libraries onto Windows Clients” on page 35.

To install the GemStone/S 64 Bit Windows Client product, use the following procedure. The procedure is similar regardless of what platform the GemStone/S 64 Bit server is running on.

1. The GemStone/S 64 Bit client installation for Windows is provided as a zipped archive with a name similar to GemBuilderC3.0.1-x86.Windows_NT.zip. Move this distribution file to the directory location in which this will be installed.
2. Create a directory named GemBuilderC3.0.1-x86.Windows_NT, into which the client will be installed, and make this directory the working directory. For example:

```
C:\InstallDir> mkdir GemBuilderC3.0.1-x86.Windows_NT
C:\InstallDir> cd GemBuilderC3.0.1-x86.Windows_NT
```

3. Unzip the distribution file using unzip. For example,

```
C:\InstallDir\GemBuilderC3.0.1-x86.Windows_NT> unzip
GemBuilderC3.0.1-x86.Windows_NT.zip
```

In addition to several subdirectories, C:\InstallDir\GemBuilderC3.0.1-x86.Windows_NT also contains two text files: `PACKING.txt`, which lists all of the GemStone files, and `version.txt`, which identifies the product and release.

4. Add the directory

```
C:\InstallDir\GemBuilderC3.0.1-x86.Windows_NT\bin
```

to the machine search path.

Installing GemStone/S 64 Bit 3.0.1 Libraries onto Windows Clients

This section describes the procedure for installing client libraries on Windows and getting GBS to recognize them.

1. Quit any running client Smalltalk VM that is using GBS.
2. Your client libraries may be placed in the directory in which the client Smalltalk executable resides, or in any directory that is on the path, `%PATH%`.

NOTE

In step 3., <GBS> refers to the directory in which the client Smalltalk executable resides, or any directory that is on the path that will be used by the GBS application, and <SRC> refers to the GemStone/S 64 Bit Windows client installation directory (for example, `installDir/GemBuilderC3.0.1-x86.Windows_NT/bin`).

3. To copy the files from the GemStone/S 64 Bit Windows client directory <SRC> to the GemBuilder installation directory, or another directory on the Windows path, <GBS>, execute the following:

```
C:\> copy <SRC>\libgcirpc64-301.dll <GBS>
C:\> copy <SRC>\libgcirpc.dll <GBS>
C:\> copy <SRC>\msvcr90.dll <GBS>
```

NOTE

The shared library name includes exactly three digits of the server version number. If the server version number contains a larger or smaller number of digits, it is truncated or extended with zeros to make these three digits.

4. Ensure that there are no other copies of these libraries on the search path.
5. If the GBS configuration parameter `libraryName` is not set, or is set to `libgcirpc.dll`, installation is complete. The correct client library will be loaded when the client Smalltalk VM is restarted and the application logs in to GemStone.

If you are converting from a different GemStone server product, your GBS application is likely to have set `libraryName` to the client library used with that product. This parameter needs to be updated for GemStone/S 64 Bit. If you are converting, continue with these instructions.

6. Start up the client Smalltalk VM.

7. Clear the library name. There are several ways to do this. Using the GBS settings dialog, select the tab titled "Server Communication", and clear the field libraryName. You may also execute:

```
GbsConfiguration current libraryName: ''
```

8. Save your image, exit, and restart, to allow the correct libraries to be loaded.

Installing GemStone/S 64 Bit 3.0.1 Libraries onto Unix Clients

GemStone/S 64 Bit supports GBS clients running on Solaris and Linux with 32-bit VisualWorks and with 64-bit VisualWorks. GemStone/S 64 Bit includes both 64-bit libraries and 32-bit libraries, since the 64-bit libraries cannot be loaded into 32-bit VisualWorks. 64-bit libraries are in `$GEMSTONE/lib/`, and 32-bit libraries are in `$GEMSTONE/lib32/`.

1. Log in to your GBS client machine as the user who is the owner of the GBS installation files.
2. Your client libraries may be placed in any directory that is on the machine search path `$LD_LIBRARY_PATH`, or you may add the directory containing the client libraries to the `$LD_LIBRARY_PATH`. Ensure that there are no other copies of the client libraries on the search path. (In the following instructions, the term "path" means the path specified by `$LD_LIBRARY_PATH`)

Alternatively, you can place the shared libraries in any location, and in the GBS configuration parameter `libraryName`, specify the absolute path to the client library file.

NOTE

In step 3., <GBS> refers to your GemBuilder installation directory, or any directory that is on the path that will be used by the GBS application, and <SRC> refers to the lib32 subdirectory of your installation (for example, `installDir/GemStone64Bit3.0.1-sparc.Solaris/lib32`).

3. To copy the files from the GemStone/S 64 Bit server source directory `<SRC>` to the GemBuilder installation directory, or another directory, `<GBS>`, execute the following:

```
% cp <SRC>/libgcirpc64-301.so <GBS>
```

NOTE

The shared library name includes exactly three digits of the server version number. If the server version number contains a larger or smaller number of digits, it is truncated or extended with zeros to make these three digits.

4. Make the destination directory the current working directory, and create a symbolic link to the library:

```
% cd <GBS>
% ln -s libgcirpc64-301.so libgcirpc.so
```

5. If the client libraries are in a directory on the path, and the GBS configuration parameter `libraryName` is not set, or is set to `libgcirpc.so`, installation is complete. The correct client library will be loaded after the client Smalltalk VM is restarted, on login.

If you are converting from a different GemStone server product, your GBS application is likely to have set *libraryName* to the client library used with that product. This parameter needs to be updated for GemStone/S 64 Bit. If you are converting, continue with these instructions.

6. Start up the client Smalltalk VM.
7. Set the library name. There are several ways to do this. Using the GBS settings dialog, select the tab titled "Server Communication", and fill in the appropriate library name in the *libraryName* field. You may also execute:

```
GbsConfiguration current libraryName: '<libraryName>'
```

If the clientLibraries are in a directory on the path, clear the *libraryName* field, or set it to an empty string. The default client library search will find and load the correct library.

If the client libraries are not in a directory on the path, *libraryName* must be set to the absolute path and name of the client library `libgcirpc.so`.

8. Save your image, exit, and restart, to allow the correct libraries to be loaded.

