GemStone®

GemStone/S 64 Bit
Installation Guide

for AIX on IBM RS/6000

Version 2.4.7

December 2015

GEMTALK™
SYSTEMS

Preface

About This Documentation

This document explains how to install GemStone/S 64 Bit™ version 2.4.7, and how to upgrade from previous GemStone/S 64 Bit versions.

For information regarding new and modified features in GemStone/S 64 Bit v2.4.7, please refer to the *GemStone/S 64 Bit Release Notes* for version 2.4.7.

These documents are also available on the GemTalk website, as described below.

Terminology Conventions

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

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 form.

▸ **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.

▸ **TechTips**, providing information and instructions that are not in the documentation.

▶ **Compatibility matrices**, listing supported platforms for GemTalk 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 assistance may be submitted online, by email, or by telephone. We recommend you use telephone contact only for more serious requests that require immediate evaluation, such as a production system down. The support website is the preferred way to contact Technical Support.

Website: support.gemtalksystems.com

Email: techsupport@gemtalksystems.com

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

When submitting a request, please include the following information:

▶ Your name and company name.

▶ The versions of GemStone/S 64 Bit and of all related GemTalk 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 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 3. Converting from GemStone/S 6.x

Chapter 4. Configuring GBS for GemStone/S 64 Bit

Chapter

1

Installing GemStone/S 64 Bit Version 2.4.7

This chapter describes the procedure for installing GemStone/S 64 Bit version 2.4.7 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 later chapter of this Installation Guide.

Adjust the installation to meet your specific needs. The topic "What Next?" on page 19 provides references to procedures and related information in the *System Administration Guide.*

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. 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

▸ IBM RS/6000 workstation running a POWER6 or POWER7 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.

Page space

▸ Total page space should be at least equal to the amount of RAM. We recommend installing twice as much page space as RAM.

Due to the way GemStone uses memory, systems with insufficient pageswap space allocated have a risk of memory errors even if there is available RAM.

Disk space

▸ Space for the installed distribution files—you need approximately 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 page space. Use of multiple disk drives is advisable for servers.

Operating system

▸ AIX version 6.1

▸ AIX version 7.1

C/C++ Compiler

▸ IBM XL C/C++ for AIX, V11.1

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

▸ dbx

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.

Configure the Operating System

1. **POSIX Asynchronous I/O (AIO) must be enabled.**

To ensure that POSIX AIO is enabled:

a. Invoke `smit`, and select **Devices**, then select **Asynchronous I/O**, then select **Posix Asynchronous I/O.**

b. Select Configure Defined Asynchronous I/O, and press Enter.

c. Select **Change/Show Characteristics of Asynchronous I/O**.

d. If you want to place GemStone transaction logs on more than one disk drive (spindle), at Maximum number of servers, specify the value as at least 10 times the number of spindles to be used for transaction logs.

GemStone uses asynchronous I/O for accessing transaction logs. Because each AIO request is handled by a separate kernel server process, this setting increases performance when transaction logs are located on more than one drive.

e. Select **STATE to be configured at system restart**, choose "available," and press OK.

f. Exit `smit`, and reboot the system.

g. After rebooting, ensure that the POSIX AIO device driver is available:

```
% lsdev -c posix_aio
```
You should see a message similar to:
```
posix_aio0 Available   Posix Asynchronous I/O
```

2. **File size and file descriptors**

To use extents in the file system larger than 2 GB or to increase the number of file descriptors available, make these changes in `/etc/security/limits`:
```
Fsize = -1
nofiles = 2000
```

You must also have the Large File System enabled when creating a jfs file system using `smit`.

We recommend that you pregrow such extents to a size greater than 2 GB so any configuration difficulty is detected when the server is first started.

The default 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. *Use caution and increase the default setting only when necessary because doing so can have system side effects.*

3. Large Memory Pages

The default size for memory pages is 4K (small). AIX also supports 64K (medium), 16MB (large) and 16GB (very large) memory pages. AIX automatically adjusts between 4K and 64K pages, but using large pages requires configuration.

If you have a large repository, using large (16MB) pages may improve performance. To use large pages, you must configure AIX to allocate large pages, and configure GemStone to request large pages.

To configure GemStone to use Large Pages:

a. Get the values for the number of pages and processes that the shared page cache will use (the configured values plus the overhead).

The easiest way to get this information is to start up GemStone with the desired configuration, and inspect the shared page cache monitor log. These example numbers are for a system with a 20 GB cache. Looking at the *stonename_pid*pcmon.log, these are the values that are needed:

```
Number of pages            1250000
Number of processes        1022
TargetFreeFrameCount       7000
Number of shared counters  1900
```

b. Enter these numbers at the shared page cache monitor shell

Start the shared page cache monitor shell

```
unix> $GEMSTONE/sys/shrpcmonitor
```

Enter the intended shared page cache size, maximum number of GemStone processes, and the number of shared counters, using the commands `setnumpages`, `setnumprocs`, `setnumsharedctrs`, and `settargetfflimit`, with postfix notation.

```
SHRPCMON>1250000 setnumpages
SHRPCMON>1022 setnumprocs
SHRPCMON>1900 setnumsharedctrs
SHRPCMON>7000 settargetfflimit
```

c. Use the `getrequiredsize` command to compute the large page requirements.

```
SHRPCMON>getrequiredsize
For 1250000 pages, 200 processes and 1900 shared counters,
required cache size is 21942435840 bytes.
To use 16 MB large memory pages: 21944598528 bytes are
required for alignment.
Number of 16 MB large pages required: 1308
```

    d. Configure AIX to use large pages.

This is done using the following commands, which are also output by get requiredsize. These commands must be run as root.

```
vmo -r -o lgpg_regions=numPages -o lgpg_size=16777216
vmo -p -o v_pinshm=1
```

After this has been executed, the bosboot command must be run to build a new kernel image, and the system must be rebooted

You can confirm large memory pages are available for use using `vmstat -l`.

    e. Ensure process owner has permission to use large pages

The UNIX user running the shared cache monitor must be given permission to use large memory pages.

```
chuser capabilities=CAP_BYPASS_RAC_VMM,CAP_PROPAGATE <user id>
```

Alternatively, you can run the SPC monitor process with an effective user ID of root:

```
chown root $GEMSTONE/sys/shrpcmonitor $GEMSTONE/sys/startshrpcmon
chmod u+s $GEMSTONE/sys/shrpcmonitor $GEMSTONE/sys/startshrpcmon
```

    f. Enable Large Memory Pages in GemStone/64

GemStone will attempt to allocate the shared page cache using large memory pages based on the value of the added environment variable GS_SPC_USE_LARGE_PAGES.

▸ If this is not set, large pages are not used.

▸ if this is set to 1, it is advisory mode: Large memory pages are requested but the cache still starts if the request is not granted by the system.

▸ If this is set to 2, it is mandatory mode. Large memory pages are requested and the cache fails to start if the request is not granted by the system.

When you start GemStone, the shared page cache monitor log will include a line of the form:

```
[Info]: Shared page cache was successfully created using
large memory pages.
```

4. Configuring GemStone's Shared Page Cache

The configuration option SHR_PAGE_CACHE_SIZE_KB defines the size (in KBytes) of extent page space in the shared page cache. The maximum acceptable value for this configuration option is limited by system memory, kernel configurations, cache space allocated by SHR_PAGE_CACHE_NUM_PROCS and space allocated for other GemStone caches.

For more general information about these and other configuration options, see Appendix A of the *System Administration Guide*.

5. 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.

6. 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.

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/rhdisk*
> Optional raw partitions for repository extents and transaction logs.

/etc/services
> Internet services database, for NetLDI name lookup.

*/InstallDir/*GemStone64Bit2.4.7-RISC6000.AIX
> 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 page space is available.:

We recommend that you allow at least 10 LPs of page for running GemStone and an additional 10 LPs if you will be developing GemStone applications.

To determine the current size of the paging device(s), which is usually found on the root volume group, rootvg:

% /etc/lsvg rootvg -l

Paging devices are of TYPE paging, and the LPs column shows the size of each device. An LP is 4 MB.

Page space can be expanded dynamically (without rebooting) using SMIT.

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 -k
```

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

4. Select an installation directory, *InstallDir*, and make this directory the current working directory.

5. GemStone/S 64 Bit is provided as a zipped archive file with a name similar to `GemStone64Bit2.4.7-RISC6000.AIX.zip`.

6. Move this distribution file to the directory location in which GemStone will be installed, *InstallDir*.

7. Unzip the distribution file using unzip. For example:

```
% unzip GemStone64Bit2.4.7-RISC6000.AIX.zip
```

8. The *InstallDir* now contains a GemStone directory with a name similar to `GemStone64Bit2.4.7-RISC6000.AIX`.

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.

9. Log in as root.

NOTE
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, which explains the concept of GemStone server file permissions and how to change them.*

Set the Environment

GemStone operation requires the $GEMSTONE environment variable be set correctly.

1. If more than one installation of any GemStone/S product resides on this machine, check for existing GemStone environment variables:

```
% env | grep GEM
```

Unset any environment variables that are not appropriate for this new installation.

2. Set the environment variable GEMSTONE to the *full pathname* (starting with a slash) of your new GemStone installation directory. For example, depending on your shell:

```
% setenv GEMSTONE InstallDir/GemStone64Bit2.4.7-RISC6000.AIX
$ GEMSTONE=InstallDir/GemStone64Bit2.4.7-RISC6000.AIX
$ export GEMSTONE
```

Create the GemStone Key File

To run GemStone, you must have a key file for this version of GemStone/S 64 Bit and for the appropriate platform. Email keyfiles@gemtalksystems.com, or contact GemTalk Technical Support, if you have questions about your keyfile, or if you need an evaluation keyfile.

Once you receive your keyfile with the name *mykeyfile.key*:

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

```
% cd $GEMSTONE/sys
% chmod 755 .
```

2. Copy the keyfile into this directory, using the name gemstone.key.

cp *mykeyfile.key* $GEMSTONE/sys/gemstone.key

You may locate and name the file differently, but you will need to specify the path using the KEYFILE parameter in the configuration file used by this repository.

3. Change the file and directory permissions so that they are not 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.

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

```
% /etc/ping hostname 8 1
```

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 system services database. A NetLDI is required for certain kinds of local and remote sessions to log into GemStone, and if it cannot be resolved by name, you must refer to it by port number. For clients on remote machines, the same NetLDI service name and port number must be defined on the remote machines as well as the main host.

If you are upgrading from a previous version, you may need to keep the NetLDI for that version running. In this case, select a distinct name and port for the NetLDI for GemStone/S 64 Bit 2.4.7.

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

   % $GEMSTONE/install/getservbyname gs64ldi
   s name=gs64ldi s port = 50377 s proto = tcp

   If you are using a local copy of the system services database, **/etc/services**, then check in this file for a definition for `gs64ldi`.

   % grep gs64ldi /etc/services

   If you are using NIS or LDAP, consult your UNIX system administrator for assistance.

   If `gs64ldi` is defined, skip the rest of this procedure and continue with the installation at "Run the Installation Script" on page 15.

   If it is not defined, continue performing this procedure.

2. Add an entry similar to the following to the system services database:

   gs64ldi 50377/tcp #GemStone/S 64 Bit 2.4.7

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

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

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 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*.)

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.*

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 and Add Users

After installing GemStone/S 64 Bit, you must change the passwords for the 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.

You must then establish GemStone accounts for each of your system's users.

The chapter entitled User Accounts and Security in the *System Administration Guide* tells you how to change the passwords and set up accounts for other GemStone users, and how to create new GemStone user accounts. These functions can also be done using GemBuilder for Smalltalk tools; see the *GemBuilder for Smalltalk Users's Guide* for more information.

Users must 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 2.4.7 has been installed, you should notify each GemStone user of the installation and explain how to use the `gemsetup` files.

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 2.4.7 directory. For example, depending on your shell:

    ```
    %  setenv GEMSTONE InstallDir/GemStone64Bit2.4.7-RISC6000.AIX
    $  GEMSTONE=InstallDir/GemStone64Bit2.4.7-RISC6000.AIX
    $  export GEMSTONE
    ```

2. Invoke the script `gemsetup`. For example, depending on your shell:

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

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.

What Next?

This chapter has guided you through installation of GemStone/S 64 Bit 2.4.7 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*. This chapter describes in detail the many options for configuring your 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.*

▸ Chapter 3 of the *System Administration Guide* has additional information about setting up distributed configurations.

▸ To start and stop the GemStone object server, refer to instructions in Chapter 4 of the *System Administration Guide*.

2 Upgrading from GemStone/S 64 Bit 2.x versions

This chapter describes how to upgrade an existing GemStone/S 64 Bit 2.x installation to GemStone/S 64 Bit version 2.4.7. GemStone/S 64 Bit version 2.4.7 supports upgrade from GemStone/S 64 Bit versions 2.2.x (2.2.5 or later), and all version of 2.3.x and 2.4.x.

If you are upgrading from 32-Bit GemStone/S, which requires conversion, see the appropriate chapter for conversion instructions.

If you are using GemBuilder for Smalltalk (GBS), you should also upgrade the client libraries that are used by GBS. See Chapter 4 for supported versions of GBS for use with GemStone/S 64 Bit 2.4.7, and instructions on installing updated client libraries.

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 2.4.7.

NOTE
The following instructions use the version number 2.4.6 to refer to the version you are upgrading from, and version number 2.4.7 indicate the target version you are upgrading to. The process is the same when upgrading from any 2.x repository, and upgrading to any 2.4.x versions for which this Installation Guide applies.

Prior to Upgrade in existing application

1. 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 2.4.7 kernel methods, and refer to the *Release Notes* for version 2.4.7 and all release notes after the version you are upgrading from, to determine whether your changes are still necessary or appropriate. For a listing of Release Notes, see GemStone/S 64 Bit Release History.

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 2.4.7.

2. Remove indexes on upgrade from Version 2.2.x

If you are upgrading from version 2.2.x, you must remove all indexes in your application. There are changes in the indexing internal structures to accommodate QuadByteStrings, introduced in v2.3, and these structures for existing indexes are not modified by upgrade.

Prepare for Upgrade

1. Install and configure GemStone/S 64 Bit 2.4.7

Install GemStone/S 64 Bit 2.4.7 to a new installation directory, separate from the installation directory for version 2.4.6, as described in Chapter 1, starting on page 7.

Configure GemStone/S 64 Bit 2.4.7 the way you expect to use it — that is, with the appropriate extent locations and sizes.

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 2.4.6 system as a user with OtherPassword privilege, such as DataCurator, and reset the SystemUser password to 'swordfish':

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

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

3. Stop user activity

Log in to the version 2.4.6 system as a user with SessionAccess and SystemControl privileges, such as DataCurator, and halt all user activity on the repository.

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

4. Shut down the repository

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

% **stopstone** *stone246*

where *stone246* is the name of the version 2.4.6 stone on this machine. The repository must be cleanly shut down to avoid needing recovery when it is restarted with the new version's executables.

5. Set up the version 2.4.7 environment.

Set the environment variables required for the upgrade.

C shell:

% **setenv GEMSTONE** *InstallDir247*
% **set path = ($GEMSTONE/bin $path)**
% **setenv upgradeLogDir** *tempDir*

Bourne or Korn shell:

$ **GEMSTONE=***InstallDir247*
$ **export GEMSTONE**
$ **export PATH=$GEMSTONE/bin:$PATH**
$ **upgradeLogDir=***tempDir*
$ **export upgradeLogDir**

where *InstallDir247* is the GemStone/S 64 Bit version 2.4.7 installation and *tempDir* is a temporary directory for which you have write permission.

NOTE
Use a separate log directory for each repository you convert.

6. Copy extent files

Copy your version 2.4.6 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 2.4.6 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 2.4.7 installation. For example:

% **cp** *InstallDir246***/data/extent0.dbf** *247location*
% **cp** *InstallDir246***/data/extent1.dbf** *247location*
% **cp** *InstallDir246***/data/extent2.dbf** *247location*

where *247location* is the location specified by DBF_EXTENT_NAMES in the configuration file that will be used in version 2.4.7.

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 2.4.7. Transaction logs from earlier versions are not compatible with version 2.4.7. If the transaction log directories will be reused for version 2.4.7, any transaction logs should be deleted or copied elsewhere.

Perform the Upgrade

1. Start the Stone

Start the 2.4.7 Stone on the 2.4.6 extents you just copied:

```
% startstone stoneName247
```

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 stoneName247
```

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 `$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.

3. Restore System Account passwords

Log in to GemStone/S 64 Bit version 2.4.7 as DataCurator or SystemUser, and change the password for SystemUser, DataCurator, and GcUser to a secure password, such as the passwords used for these accounts in v2.4.6. For example:

```
topaz 1> printit
(AllUsers userWithId: 'SystemUser') password: '246Password'.
(AllUsers userWithId: 'GcUser') password: '246Password'.
(AllUsers userWithId: 'DataCurator') password: '246Password'.
System commitTransaction
%
```

where *246Password* is the account password used in version 2.4.6.

Post-upgrade Application Code Modifications

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

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

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

2. **File in Kernel class changes**

   If you have modified any kernel class methods of the previous version or if you have added methods to kernel classes, carefully compare your changes with the changes in version 2.4.7 to see whether your changes are still necessary or appropriate. Carefully review the *Release Notes* for each intervening version, as well as examining code in the image.

   If the kernel class changes are still applicable, file in the changes, verify that errorcount is 0, and commit.

Backup repository and configure GBS

1. **Make backup**

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

2. **Configure GBS**

   If you are using GBS clients, configure GBS to use the version 2.4.7 client libraries. See Chapter 4, 'Configuring GBS for GemStone/S 64 Bit' for details.

Chapter

3

Converting from GemStone/S 6.x

This chapter describes how to convert an existing GemStone/S installation to GemStone/S 64 Bit version 2.4.7. GemStone/S 64 Bit version 2.4.7 supports upgrade from GemStone/S version 6.5.2 or later. If you are converting from 6.1.x, you must first upgrade to version 6.5.2 or later, before converting to GemStone/S 64 Bit version 2.4.7.

While GemStone/S is similar to GemStone/S 64 Bit, there are differences in code and administration. Conversion to GemStone/S 64 Bit requires modifications to your existing application.

As part of the conversion process, you:

▸ Modify classes and methods that run in GemStone so they work properly with the GemStone/S 64 Bit classes and methods. This includes GemStone Smalltalk application code, and C, C++, andclient Smalltalk interface code. This includes ensuring that no methods are larger than 64K.

▸ Convert and upgrade your repository in a multi-stage process per the instructions starting on page 26.

▸ Reinstall any upgraded add-on products, and restore your GemStone kernel class changes.

▸ Recompile and relink any C or C++ user actions or client applications

The architecture of the repository in GemStone/S 64 Bit 2.4.7 is substantially different from GemStone/S 6.x. For that reason, the upgrade from v6.x to v2.4.7 is a conversion that cannot be done in place. The upgrade conversion process will construct a new GemStone/S 64 Bit repository, using the 6.x repository as a template. When the conversion is completed, both the 6.x and 2.4.7 repositories will exist. This means that enough disk space must be available to accommodate both repositories.

The GemStone/S 64 Bit 2.4.7 repository will be substantially larger than the GemStone/S 6.x repository. Repository size increases between 20% and 200% or more are likely, depending on the nature of the data in the repository.

The conversion process will not substantially modify the original 6.x repository. The 6.x repository will remain in a fully usable and coherent state after the conversion is completed.

Conversion Strategy

We recommend that you perform the upgrade conversion twice: first a pilot conversion and then the production conversion. With this strategy, you can keep your version 6.x production system running while you familiarize yourself with the conversion process. The steps are the same for the pilot and the production conversions; where there are difference, this is noted in the instructions.

If possible, the pilot conversion should be done on a copy of your production system. During the pilot conversion, the database integrity is verified before and after conversion, and growth information on the repository is determined that will allow you to configure the final version 2.4.7 system correctly.

Conversion Procedure

The conversion procedure from GemStone/S 6.x to GemStone/S 64 Bit 2.4.7 is performed in several stages. The first stage executes in the existing 6.x environment, that is, with $GEMSTONE pointing to the GemStone/S v6.x product tree. Do not reset $GEMSTONE until specified in the instructions. The following phases of conversion execute in the GemStone/S 64 Bit environment.

The following list summarizes the steps necessary to perform the conversion to GemStone/S 64 Bit v2.4.7.

> *NOTE*
> *The following instructions use the version number 6.6.5 to refer to the version you are upgrading from, and version number 2.4.7 indicate the target version you are upgrading to. The process is the same when upgrading from any 6.x repository, and upgrading to any 2.4.x versions for which this Installation Guide applies.*

Preprocessing in Old Environment

1. 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 2.4.7 kernel methods, to determine whether your changes are still necessary or appropriate. For a listing of GemStone/S 64 Bit Release Notes, see [GemStone/S 64 Bit Release History](#).

2. **Verify that you have no methods larger than 64K.**

    Before conversion, you must verify that you have no methods that are larger than 64K. Oversize methods do not cause an error in the conversion, but cannot be loaded or executed in the converted repository. GemStone/S 64 Bit 2.4.7 provides a tool to assist in detecting oversize methods - see "Detecting Oversize Methods" on page 3-34

3. **Install and configure GemStone/S 64 Bit 2.4.7**

    Install GemStone/S 64 Bit 2.4.7 to a new installation directory, separate from the installation directory for version 6.6.5, as described in Chapter 1, starting on page 7.

    The preprocessing in your 6.x repository requires scripts that are part of the GemStone/S 64 Bit 2.4.7 distribution. You do not need to configure the v2.4.7 repository at this point.

4. **Set GEMSTONE_64 and upgradeLogDir Environment Variables**

    Define a the environment variable, $GEMSTONE_64, to point to this product tree. At this point, do not reset $GEMSTONE or update the path to point to v2.4.7.

    Also define the environment variable that will hold the upgrade log files to a temporary directory for which you have write permission..

    For example, using the default GemStone/S 64 Bit 2.4.7 installation location from Chapter 1:

    C shell:

    ```
    %  setenv GEMSTONE_64 <InstallDir>/GemStone64Bit2.4.7-<platform>
    %  setenv upgradeLogDir tempDir
    ```

    Bourne or Korn shell:

    ```
    $  GEMSTONE_64=<InstallDir>/GemStone64Bit2.4.7-<platform>
    $  export GEMSTONE_64
    $  upgradeLogDir=tempDir
    $  export upgradeLogDir
    ```

5. **Remove $GEMSTONE from extent file paths in system configuration file**

    Check the system configuration file used by your existing version 6.x installation. If the environment variable $GEMSTONE is used in the paths for extent file locations in the DBF_EXTENT_NAMES, this must be changed to use the absolute path. The conversion will not proceed with $GEMSTONE in the extent file path/s.

6. **(Pilot conversion) Run an object and page audit**

    If this is the pilot conversion, to verify the integrity of your v6.x system, run an object audit and a page audit prior to conversion. For the page audit, which requires the stone to be shut down, you may choose to run this following the **convprep6x** step (Step 11. on page 28).

    For instructions in running object and page audits, see the *System Administration Guide*.

7. **Ensure there are no dead objects**

    Prior to conversion, perform reclaim to ensure there are no dead objects in your repository.

8. **(Optional) Set up classes for list instances**

Some customers may need to perform application level object conversions on their repository, after the GemStone conversion has completed. For such applications, GemStone can be instructed to create lists of instances of particular classes during conversion. This behavior is not enabled by default. To enable, you must define a UserGlobal with a specific name and structure. Collecting the lists of instances will slow down the conversion process; larger numbers of classes and instances of these classes will slow down the conversion more.

To collect lists of instance of classes, before invoking the **convprep6x** script, do the following:

a. Determine the list of classes for which instances are to be collected by the conversion.

b. Login to the v6.x system as SystemUser and ensure you are in transaction.

c. Create an IdentitySet containing one of more classes for which instances are to be located.

d. Store the IdentitySet in UserGlobals under the key `#ConversionClassesToFind`.

e. Commit the transaction.

During the execution of the **convprep6x** script, if the symbol `#ConversionClassesToFind` is found, information is written to the text file `$upgradeLogDir/ClassesForListInstances.txt`. Results are written to bitmap files which can be loaded into the final converted GemStone/S 64 Bit v2.4.7 repository.

9. **Reset SystemUser password**

Log in to the version 6.6.5 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.

10. **Stop user activity**

Log in to the version 6.6.5 system as a user with SessionAccess and SystemControl privileges, such as DataCurator, and halt all user activity on the repository.

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

11. **Run the GemStone/S 64 Bit 2.4.7 script convprep6x**

The **convprep6x** script is in the upgrade directory of the 2.4.7 product tree; this should not be in your path, so it will need to be invoked with an explicit path. For example,

```
% $GEMSTONE_64/upgrade/convprep6x -s stoneName665
```

This script has the following options:

```
convprep6x [-h] [-s <stonename>]
```
-h prints this usage information.
-s *<stoneName>* sets the name of the running v6.x Stone to scan; if this option is not used, the script will default to **gemserver63.**

The **convprep6x** script scans the v6.x repository and produces files that are used in the second phase of conversion. These files are written to **$upgradeLogDir**. The **convprep6x** script will write progress messages to stdout. When it completes, it will report:

```
convprep6x[INFO]: ...conversion preparation complete.
```

If this output is not produced, check the log files in the **$upgradeLogDir** directory for information.

This script also shuts down the Stone.

While the v6.x system is not substantially affected by conversion preparation and remains completely usable, do not restart or use these extents until the conversion is complete.

Prepare for Conversion

1. Configure GemStone/S 64 Bit 2.4.7

In particular, you should examine and update:

a. DBF_EXTENT_SIZES. The conversion process creates new v2.4.7 extents based on the v6.x extents; the v2.4.7 extents will be 20% to 200% larger. If this is the pilot, estimate the new size; if you are pre-growing extents allow for the larger growth.

For the production conversion, set the new sizes based on growth information from the pilot conversion.

Ensure that adequate space is available for the extents, as well as tranlogs and log files.

b. DBF_PRE_GROW. We recommend setting this to true for the conversion process, in particular for the production conversion. This ensures that you will not run out of disk space during the conversion process.

c. STN_TRAN_LOG_DIRECTORIES. For large repositories, we recommend setting the tranlogs to /dev/null for the conversion process. Tranlogs from the conversion process are not useful.

d. SHR_PAGE_CACHE_SIZE_KB. To ensure the performance of the v2.4.7 system equals that of the v6.x system, the shared page cache size, for all shared caches including the gem server SPC, should be increased by the same percentage as the growth in the extent files. If this is the pilot, start by estimating this to increase by 50%. It is important that the entire object table fit into the SPC.

For the production conversion, use the increase computed from growth information from the pilot conversion.

GEM_TEMPOBJ_CACHE_SIZE, GEM_PRIVATE_PAGE_CACHE_KB, and STN_PRIVATE_PAGE_CACHE_KB do not need to be increased.

2. Set the v2.4.7 Environment Variables

Set the $GEMSTONE and $path environment variables required for GemStone/S 64 Bit.

C shell:

```
% setenv GEMSTONE InstallDir247
% set path = ($GEMSTONE/bin $path)
```

Bourne or Korn shell:

```
$ GEMSTONE=InstallDir247
$ export GEMSTONE
$ export PATH=$GEMSTONE/bin:$PATH
```

where *InstallDir247* is the GemStone/S 64 Bit version 2.4.7 installation. You will no longer need the $GEMSTONE_64 environment variable.

The $upgradeLogDir environment variable that was defined in Step 4. on page 27 must be available and set to the same directory.

3. Copy a clean v 2.4.7 extent

Copy a clean v2.4.7 extent0.dbf to the location specified by the first entry in the of extent files, as it appears in the DBF_EXTENT_NAMES parameter setting in the configuration file that will be used by the 2.4.7 repository. For example,

```
% cp $GEMSTONE/bin/extent0.bdf $GEMSTONE/data
% chmod a+w $GEMSTONE/data/extent0.dbf
```

New extent files for v2.4.7 will be created as part of conversion. **Do not** copy your version 6.x extent files. All 6.x extents should be available in their 6.x locations.

Perform the Conversion and Upgrade

1. Start the v 2.4.7 Stone

Start the 2.4.7 Stone on clean extents

```
% startstone stoneName247
```

2. Convert using conv6xTo2x script

Extent conversion is performed by the script **conv6xTo2x**. This script is in the v2.4.7 product tree, and does not require a explicit path.

This script requires as an argument the configuration file used by the v6.x installation. This configuration file must not have the extent file names specified using $GEMSTONE.

It also has optional switches to specify the Stone name, the number of gems performing the conversion, and the number of OOPs per commit.

This script also includes the -L switch that enables recording of all references to Large Integers, and the -F switch to record all references to Float or SmallFloat. The conversion process does not convert instances of LargePositive- or LargeNegativeIntegers that fall within the new SmallIntegers range, nor instances of Float or SmallFloat that are representable as instances of SmallDouble. This can be done as an optional manual step following conversion. Note that using the -L or -F options will slow down the conversion process, since each object must be checked for references.

```
conv6xTo2x -e <oldSysConf> [-h] [-s <stoneName>] [-n
<numConversionGems>] [-o <oopsPerCommit>] [-F] [-L]
```
`-e <oldSysConf>` specifies the configuration file to use. This configuration file must contain the list of 6.x extents to read data from. This is required; the script will not run without this information.
`-h` prints this usage information.
`-s <stoneName>` sets the name of the running stone to convert; if this option is not used, the script will default to **gs64stone.**
`-n <numConversionGems>` sets the number of parallel gems that will do the conversion; if this option is not used, the script will default to use one.
`-o <oopsPerCommit>` sets the number of oops per transaction; if this option is not used, the script will default to use 1000000.
`-F` During conversion, determine which objects reference one or more instances of Float or SmallFloat and write the object IDs to the binary bitmap file: `$upgradeLogDir/AllFloatRefs.bm`.
`-L` During conversion, determine which objects reference one or more instances of LargePositiveInteger or LargeNegativeInteger and write the object IDs to the binary bitmap file: `$upgradeLogDir/AllLrgIntRefs.bm`

For example,

% **conv6xTo2x** -e *installLocation665***/data/system.conf -s** *stoneName247*

The **conv6xTo2x** script will write progress messages to stdout. When it completes, it will report:

`conv6xTo2x[INFO]: Successful conversion.`

The stone will shut down at the completion of the `conv6xTo2x` step. This may take some time as the conversion gems terminate. Wait until the stone has completed and shut down before continuing.

3. Restart the 2.4.7 Stone

The conversion shuts down the stone when it completes. Restart the Stone:

% **startstone** *stoneName247*

4. Upgrade using upgradeImageFrom6x script

Run the script that performs the kernel class filein to upgrade the image.

The image upgrade is performed by the script `upgradeImageFrom6x`. 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.

`upgradeImageFrom6x [-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,

% **upgradeImageFrom6x -s** *stoneName247*

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.

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.

5. Perform postprocessing using postconv script

To convert large objects found during the conversion process, run the postconv
script. This script has optional flags to specify stone name and the number of sessions
to use.

```
postconv [-c <numCacheWamerGems>][-h][-s <stoneName>]
[-n <numberOfSessions>] [-t <tempObjCacheSize]
```
-c <numCacheWamerGems> specifies the number of cache warmer gems to start
before starting post conversion; if this is not specified, no cache warmer gems are
started.
-h prints this usage information.
-s <stoneName> sets the name of the running stone to scan; if this option is not used,
the script will default to **gs64stone.**
-n <numberOfSessions> sets the number of parallel sessions to run to perform the
large object conversion. If this option is not specified, the script will default to one.
-t <tempObjCacheSize> sets the size of GEM_TEMPOBJ_CACHE_SIZE in KB; by
default, the same value as that used in the upgradeImage step, which defaults to
200000.

For example,

```
% postconv -s stoneName247
```

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

```
postconv[INFO]: Success! Conversion process from GemStone 6.3.x
or later to GemStone64 2.x completed.
```

6. (Pilot conversion) Run an object and page audit

If this is the pilot conversion, to verify the integrity of your v6.x system, run an object
audit and a page audit to verify the integrity of the repository following the
conversion.

This is also recommended for the production upgrade, if time allows.

7. Set System Account passwords

Log in to GemStone/S 64 Bit version 2.4.7 as DataCurator or SystemUser, and change
the password for SystemUser, DataCurator, and GcUser to a secure password, such as
the passwords used for these accounts in v6.6.5. For example:

```
topaz 1> printit
(AllUsers userWithId: 'SystemUser') password: '665Password'.
(AllUsers userWithId: 'GcUser') password: '665Password'.
(AllUsers userWithId: 'DataCurator') password: '665Password'.
System commitTransaction
%
```

where *665Password* is the account password used in version 6.6.5.

Post-upgrade Application Code Modifications

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

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

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

2. File in Kernel class changes

During the pilot upgrade, you should determine if any kernel class methods that you have modified or added in the GemStone/S v6.x repository are necessary or appropriate, or if changes are required to make them compatible.

If the kernel class changes are applicable and compatible, file in the changes, verify that errorcount is 0, and commit.

3. Perform any application specific post-processing for instances

If you set up `#ConversionClassesToFind` in your v6.x repository in order to collect instances of specified classes, the results will be written to bitmap files in the `$upgradeLogDir` directory.

There will be one file per Class, the file will have a name of the form:

<ClassName>_<oldOop>_<newOop>`.bm`

where:

<className> – Name of the class .
<oldOop> – GemStone/S v6.x OOP of the class.
<newOop> – GemStone/S 64 Bit v2.4.7 OOP of the class.

GemStone/S 64 Bit v2.4.7 contains Smalltalk methods to load bitmap files into customer-usable hidden sets. From these hidden sets, any required postconversion processing of the instances of these classes can be done.

Prior to the pilot conversion, you should determine if any application-specific post-processing is required on instances in your repository, and write and test any code required to perform the post-processing.

4. Perform any application specific post-processing for Large Integers or Floats

If you specified tracking of references to Large Integers or Floats, by using the `-L` or `-F` options during the `conv6xTo2x` script execution, the results will be written to a bitmap file:

```
$upgradeLogDir/AllLrgIntRefs.bm
$upgradeLogDir/AllFloatRefs.bm
```

GemStone/S 64 Bit v2.4.7 contains Smalltalk methods to load bitmap files into customer-usable hidden sets. From these hidden sets, you may load the individual objects and manually update the references.

This step is optional. Prior to the pilot conversion, if you determine that you should upgrade references to Large Integers and Floats, you should write and test any code required to perform the post-processing.

5. **Recompile and relink any C GCI client applications or user actions**

Backup repository and configure GBS

1. **Make backup**

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

2. **Configure GBS**

   If you are using GBS clients, configure GBS to use the version 2.4.7 client libraries. See Chapter 4, 'Configuring GBS for GemStone/S 64 Bit' for details.

Restore 6.x environment

The following steps are optional, if you will be continuing to use the originating GemStone/S 64 Bit v6.6.5 repository.

1. **Restart the Stone**

   ```
   % startstone stoneName665
   ```

2. **Set System Account passwords**

   Log in to the GemStone/S 6.6.5 repository 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 v6.6.5. For example:

   ```
   topaz 1> printit
   (AllUsers userWithId: 'SystemUser') password: '665Password'.
   (AllUsers userWithId: 'GcUser') password: '665Password'.
   (AllUsers userWithId: 'DataCurator') password: '665Password'.
   System commitTransaction
   %
   ```

   where *665Password* is the account password used in version 6.6.5.

3. **Remove ConversionClassesToFind**

   If you set up #ConversionClassesToFind in your v6.x repository, be sure to remove this variable to avoid retaining references. Log in as SystemUser, execute:

   ```
   UserGlobals removeKey: #ConversionClassesToFind.
   ```

   and commit.

Detecting Oversize Methods

Methods larger than 65K are not supported in GemStone/S 64 Bit, and cannot be loaded or executed. If your GemStone/S version 6.x repository contains methods larger than this limit, these methods must be factored into multiple smaller methods before the repository is converted; this task cannot be done after conversion.

GemStone/S 64 Bit 2.4.7 provides a script to detect if any methods exceed the size limit. To use this script, the GemStone/S version 6.x repository that you wish to verify must be

running, and not yet converted to GemStone/S 64 Bit, and GemStone/S 64 Bit 2.4.7 must have been installed.

1. Ensure that a stone is running on the GemStone/S version 6.x repository.

2. Verify that GemStone/S 64 Bit 2.4.7 is installed

3. Start topaz, and log in to the GemStone/S version 6.x repository as SystemUser.

4. Execute the script `methodsTooLargeFor64.topaz`. For example,

    `topaz 1>input` *gs64install*`/upgrade/methodsTooLargeFor64.topaz`

    where *gs64install* is the installation directory for GemStone/S 64 Bit 2.4.7.

5. Wait for the scan to complete. This will take some time as it scans all extents for instances of GsMethod. When it completes, log out and exit topaz.

6. The script prints the results of the scan both to standard out and to a text file `methodsToLargeFor64Report.log`, in the working directory. This report provides details on any methods that are exceed the size limit, including the current size, class and selector. If there are methods that exceed the size limit, factor these into multiple smaller methods and commit.

    Execute a markForCollection and allow reclaim to complete, before rerunning the script to verify that the modified methods are now of appropriate size. Otherwise, the script may detect old instances of methods that have not yet been garbage collected.

Chapter

4

Configuring GBS for GemStone/S 64 Bit

Versions of GemBuilder for Smalltalk (GBS) that are compatible with GemStone/S 64 Bit v2.4.7 do not support Smalltalk client applications running on Solaris/x86, AIX, or Macintosh.

GemStone/S 64 Bit v2.4.7 supports client Smalltalk/GBS applications running with VisualWorks on Solaris/SPARC, Linux, and Windows, and with VA Smalltalk running on Windows. For a table of all supported GBS and client Smalltalk platforms, see the *GemStone/S 64 Bit Release Notes* for v2.4.7.

For instructions for updating GBS clients that are running on Solaris/SPARC or Linux, see the *GemStone/S 64 Bit Installation Guide* for that platform; for GBS clients running on Windows, see the *GemStone/S 64 Bit Windows Client Installation Guide*.

For a table of supported GBS and client Smalltalk platforms, see the *GemStone/S 64 Bit Release Notes* for v2.4.7.