GemStone®

GemStone/S 64 Bit™

Release Notes

Version 2.4.6

December 2013

INTELLECTUAL PROPERTY OWNERSHIP

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

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 GemTalk Systems.

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

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

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

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

COPYRIGHTS

This software product, its documentation, and its user interface © 1986-2013 GemTalk Systems, LLC. All rights reserved by GemTalk Systems.

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 GemTalk Systems, LLC., or 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**, **Windows Vista** and **Windows 2008** 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**, **POWER6**, and **POWER7** 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. GemTalk Systems cannot attest to the accuracy of all trademark information. Use of a term in this documentation should not be regarded as affecting the validity of any trademark or service mark.

GemTalk Systems
15220 NW Greenbrier Parkway
Suite 240
Beaverton, OR 97006

Preface

About This Documentation

These release notes describe changes in the GemStone/S 64 Bit™ version 2.4.6 release. Read these release notes carefully before you begin installation, conversion testing, or development with this release.

No separate Installation Guide is provided with this release. For instructions on installing GemStone/S 64 Bit version 2.4.6, or upgrading or converting from previous products or versions, see the Installation Guide for version 2.4.5.

These documents are also available on the GemStone customer 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

GemStone Website

http://gemtalksystems.com/techsupport

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**, including current and recent versions of GemStone/S products.

‣ **Bugnotes**, identifying performance issues or error conditions that you may encounter when using a GemStone product.

‣ **TechTips**, providing information and instructions that are not 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 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: http://techsupport.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 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 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 your GemStone account manager.

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.

Contents

Chapter 1. GemStone/S 64 Bit 2.4.6 Release Notes

Chapter

1 GemStone/S 64 Bit 2.4.6 Release Notes

Overview

GemStone/S 64 Bit 2.4.6 is a new version of the GemStone/S 64 Bit object server. This release provides feature enhancements and fixes a number of serious bugs; we recommend everyone using GemStone/S 64 Bit 2.x upgrade to this new version, if not able to upgrade to 3.x.

These release notes provide changes between the previous public release of GemStone/S 64 Bit, version 2.4.5.1, and version 2.4.6. All changes in the limited-distribution special release versions 2.4.5.2 and 2.4.5.3 are in version 2.4.6 and included in these release notes. If you are upgrading from a version prior to 2.4.5.1, please also review the release notes for each intermediate release to see the full set of changes.

Installation and Upgrade

No separate Installation Guide is provided with this release. For installation instructions, use the Installation Guide for version 2.4.5.

Upgrading from 2.4.5.x versions

To upgrade from an earlier 2.4.5.x version to version 2.4.6, follow the instructions in the version 2.4.5 Installation Guide chapter "Upgrading from previous GemStone/S 64 Bit 2.x versions", replacing 2.4.4.7 with your previous version and replacing 2.4.6 for 2.4.5 in the instructions.

Supported Platforms

Platforms

GemStone/S 64 Bit version 2.4.6 is supported on the following platforms:

‣ Solaris 9 and 10 on SPARC
‣ Solaris 10 on x86
‣ AIX 5.3, 6.1, and 7.1
‣ SuSE Linux ES 10 and 11, and Red Hat Linux ES 5.0, 5.5, and 6.1

Note that HP-UX is no longer a supported platform.

For more information and detailed requirements for each supported platform, please refer to the GemStone/S 64 Bit v2.4.5 Installation Guide for that platform.

GBS versions

The following versions of GBS are supported with GemStone/S 64 Bit version 2.4.6.

GBS version 7.6

VW 7.9.1 32-bit	VW 7.9.1 64-bit
‣ Windows 7, and Windows XP ‣ Solaris 10 on SPARC ‣ SuSE Linux ES 10 and 11; Red Hat Linux ES 5.5 and 6.1	‣ Solaris 10 on SPARC ‣ SuSE Linux ES 10 and 11; Red Hat Linux ES 5.5 and 6.1

GBS version 7.4.1

VW 7.8.1 32-bit	VW 7.8.1 64-bit
‣ Windows 7 and Windows XP ‣ Solaris 10 on SPARC ‣ SuSE Linux ES 10 and 11; Red Hat Linux ES 5.5 and 6.1	‣ Solaris 10 on SPARC ‣ SuSE Linux ES 10 and 11; Red Hat Linux ES 5.5 and 6.1

GBS version 5.3.2

VA Smalltalk 8.5.2	VA Smalltalk 8.0.2	VA Smalltalk 7.5.2
‣ Windows 2008, Windows 7, and Windows XP	‣ Windows 2008, Windows 7, and Windows XP	‣ Windows XP

For more details on supported platforms and requirements, see the Release Notes for that version of GemBuilder for Smalltalk.

Changes and New Features

Finding multiple reference paths to an object.

The methods findReferencePathToObject:* return after finding a single reference to the object, which may require the method to be run multiple times to locate all references.

Methods have been added to find all reference paths in a single execution.

```
Repository>>findAllReferencePathsToObject:
Repository>>findAllReferencePathsToObjects:
Repository>>findAllReferencePathsToObjects:limitObjArray:printToLog:
Repository>>findAllReferencePathsToObjects:printToLog:
```

These methods return all reference paths to the given object or objects, rather than stopping after the first reference path is found. Otherwise, the behavior is similar to `findReferencePath*` methods, with the return value providing an array of reference paths rather than one reference path.

See comments in the method `Repository >> findAllReferencePathsToObjects:limitObjArray:printToLog:` for details.

Pause Reclaim under low free space conditions

On some systems, reclaim can proceed so quickly that freeing up pages cannot keep up, causing the repository to run out of free space.

To avoid this issue, reclaim is now suspended when it reaches a given limit in free space size in the repository. When free space again becomes higher than this limit, reclaim will restart. By default, reclaim will be suspended with the repository freespace is 0.1%, with a minimum of 5MB. So for example, with a 100GB repository, reclaim is suspended when freespace is down to 100MB.

To configure this, use the new GcUser parameter:

UserGlobals at: #reclaimMinFreeSpaceMb put: 0.
    Minimum repository free space, expressed in megabytes, which must be available in order for reclaims to proceed. Reclaims will be temporarily suspended if the repository free space drops below this threshold. The default value of 0 specifies a varying reclaimMinFreeSpaceMb that is computed as the current size of the repository divided by 1000, with a minimum value of 5 mega bytes. The default calculation is the same as that used by the stone for the STN_FREE_SPACE_THRESHOLD configuration parameter.
    Minimum: 0
    Maximum: 65536
    Default: 0

When reclaims are suspended or resumed related to low free space, each reclaim GcGem will print a message to its log file.

Performance improvements when entire Repository in SPC

Repository operations that place heavy demand on pages normally manage pages to reduce page "churn" on the shared page cache. This handling is unnecessary for systems that are configured with a shared page cache large enough that the entire repository can fit into the cache.

For better performance for systems in which the entire repository is in memory, alternate algorithms have been added. These options are not used by default.

Repository-wide scan

A new configuration parameter, GEM_REPOSITORY_IN_MEMORY, has been added to improve performance of operations that scan the entire repository, when the entire repository is in memory. This can be set at runtime using #GemRepositoryInMemory.

The standard repository scan code either copies pages from the SPC into its private buffer, or performs large sequential reads to load a set of pages into its buffer, before processing. This places less pressure on the SPC when scanning all data pages in the repository.

The new algorithm does not use a private buffer, nor does it copy pages. If a page is found in the SPC, it is pinned (locked), processed, and unlocked. If a data page is not in the cache, it is loaded into the shared cache in a single-page read, then locked and processed. With most pages already in the cache, this avoids the overhead of copying pages.

If this configuration setting is FALSE (the default), the old algorithm is used and there is no change in performance from previous releases.

For details on this parameters and the affected methods, see
"GEM_REPOSITORY_IN_MEMORY" on page 16

Fast FindDisconnectedObjects

The Fast FDC operation, which is invoked using Repository>> findDisconnectedObjectsAndWriteToFile:pageBufferSize:saveToRepository:numCacheWarmers: and specifying a non-zero number of cache warmers, now also supports optimization for the entire repository in memory.

If numCacheWarmers is specified as -1, no cache warmers are started. Internally, an algorithm similar to described for repository scans is used; pages are pinned in the cache, rather than being copied to a buffer.

Access to write set union

The Admin GC Gem can now write the write set union bitmap to a binary bitmap file before sweeping the union. To activate, use the new GcUser parameter #saveWriteSetUnionToFile. This is false by default.

When true, it writes to a file named following the pattern for the Admin GC Gem log, appended by "wsu_*N*.bm", where *N* is 0, or a higher number if a file with that name already exists. This file can be loaded and accessed using hidden set protocol.

```
UserGlobals at: #saveWriteSetUnionToFile put: false
```
> If true, causes the Admin GC gem to write out the write set union bitmap to a binary file before sweeping the union. The file will be placed in the same directory as the Admin GC gem log file.

Methods added to combine repository profiling and listInstances

The following methods have been added to allow a single repository-wide scan to both perform a listInstances and execute profileRepository.

GsObjectInventory >> profileRepositoryAndListInstancesInPageOrder: toFile:

This method combines the functions of the following methods in a single scan of the repository:
   GsObjectInventory>>profileRepository
   Repository>>listInstancesInPageOrder: toFile:

Returns an Array containing 2 elements:
   [1] - An new instance of the receiver containing the profiling result.
   [2] - An Integer indicating the number of object identifiers written to the file.

GsObjectInventory >> profileRepositoryAndSkipHiddenClassesAndListInstancesInPageOrder:toFile:

Combines the functions of the following methods in a single scan of the repository:
   GsObjectInventory>>profileRepositoryAndSkipHiddenClasses
   Repository>>listInstancesInPageOrder: toFile:

Returns an Array containing 2 elements:
   [1] - An new instance of the receiver containing the profiling result.
   [2] - An Integer indicating the number of object identifiers written to the file.

Enhancements to persistent shared counters

Increased number of counters

The number of persistent shared counters has been increased from 128 to 1536.

Query for number of persistent shared counters

A method has been added to query for the number of persistent shared counters

```
System class >> numberOfPersistentSharedCounters
```

Enhancements to shared counters - increased range

Shared counters now permit Integers in the range -2^{63} to 2^{63}

Note that these are the transient shared counters, distinct from persistent shared counters.

Ability to skip call to shmdt() on cache detach

On POWER7 hardware, especially with hyperthreading, processes may hang for several seconds in shmdt() on cache detach. It is possible similar issues exist on other platforms. To avoid this system call, set

```
export GS_DISABLE_SHMDT=1
```

If not in quiet mode, a message similar to the following is printed to stdout at cache disconnect:

```
[Info]: Skipping shared memory segment detach system call shmdt()
```

Improvements to Cache Warming

Fast Cache Warmers

A new type of cache warmer has been added that loads data pages using a new, more efficient algorithm. Previously, as each cache warmer read its section of the object table, it created its own list of data pages. The data pages on these lists overlapped with those of other cache warmers, causing duplicate data page reads. With the new fast cache warming, after OT cache warming, one gem scans the entire OT to determine the complete list of data pages, which it writes to a bitmap file. Then each cache warmer reads a unique section of the data pages range from this file. The bitmap file by default is written to /tmp/*stoneName*_all_data_pages.bm; a different location and filename can be set using the -b option.

T o start fast cache warmers, use the new utility script, startfastcachewarmer. Options are similar to startcachewarmer.

```
startfastcachewarmer [-b fileName][-d|-D][-h][-l limit][-L path]
    [-n numGems][-p password][-s stone][-u userID][-w delayTime][-W]
```

-b *fileName*	File name to which the data page bitmap file will be written. Only used if -d or -D is specified. Default: tmp/<stone>_all_data_pages.bm.
-d	read data pages into the cache. Default: only object table pages are read.
-D	Read data pages into the UNIX file buffer cache and not the SPC.

NOTE: -d and -D are mutually exclusive. If both -d and -D are passed on the command line, the last of these flags will be used.

-h	Display this message and exit.
-l *cacheFullLimit*	Stop cache warming if the free frame count falls below *cacheFullLimit*. Use -1 to have system compute a default value. Use 0 to force cache warming to continue even if the shared cache is full. Default: -1.
-L *path*	Path to a writable log file directory. Default: current directory.
-n *numGems*	Number of gems to start (default: 1)
-p *password*	GemStone password for logging in gems. Default: 'swordfish'.
-s *stone*	Name of running stone. Default: 'gs64stone'.
-u *userID*	GemStone user for logging in gems. Default: 'DataCurator'.
-w *delayTime*	Wait *delayTime* seconds between spawning gems. Default: 1.
-W	Wait for cache warming gems to exit before exiting this script. Default: spawn gems in the background and exit immediately.
-T	Setting for GEM_TEMPOBJ_CACHE_SIZE used by cache warmers gems. Default: 5000 KB.

Controlling size of private buffer for cache warmers

Cache warmers use a private buffer that was previously hardcoded to 128 pages (2MB). This can now be controlled. The default remains 128 pages. Values between 8 and 1024 can be specified using the new environment variable GS_CACHE_WARMER_PAGE_BUF_SIZE.

export GS_CACHE_WARMER_PAGE_BUF_SIZE=128

Internal Performance Improvements

Internal changes have improved performance in a number of areas.

Performance degradation when number of extents increases

The page hash function did not scale well, so performance degraded for the same operation with an increasing number of extents. The page hash algorithm has been improved in this release. (#42475)

Other internal performance improvements

There have also been improvements in the page cache entry (PCE) search and copy operations. In addition, read-only hash table locks, which were added to 3.x, have been back-ported to this release.

Cache Statistics Changes

Tracking out-of-memory with added stat OldGenPreGcSizeBytes

The cache statistic **OldGenSizeBytes** tracks old-gen memory use at the end of an in-memory garbage collection operation. When there is heavy demand on memory, this may not be helpful in tracking the peak memory use, to avoid out of memory conditions. **The following stat has been added, which records the memory use prior to the in-memory GC.**

OldGenPreGcSizeBytes (Gem)
Number of used bytes in the old generation at the beginning of mark/sweep.

Other Added Cache Statistics

ProcessesWaitingForQueueLocks (Gem)
Number of processes attached to the shared cache which are spinning while attempting to acquire a queue lock.

TimeInWaitsForOtherReaders (Gem)
The real number of milliseconds the process spent waiting for the read of another process to complete.

Removed Cache Statistic

The following cache statistics has been removed:

LocalPageCacheWrites

Support for Large Page Memory on AIX

Support for 64K memory pages on AIX

The default size for memory pages on AIX is 4K (small). Current releases of AIX also support 64K (medium), 16MB (large) and 16GB (very large) memory pages. AIX automatically adjusts between 4K and 64K pages, but using the large memory page sizes require special configuration.

GemStone/S 64 Bit now uses 64K page size on AIX for data, text, and stack. This is done automatically, no user configuration is require.

This release also provides support for configuring GemStone to use 16MB pages for the Shared Page Cache only, as described in the next section.

Support for 16M Memory Pages for SPC on AIX

GemStone now supports using Large Memory Pages on AIX for the shared page cache. Large memory pages are 16MB. While default 4KB pages and medium 64KB page sizes are handled automatically, AIX requires special configuration to use large pages.

To use large memory pages on AIX for the shared page cache, you must do the following:

1. Determine how much space you require for the SPC and configure AIX to allocate at least this many large memory pages. Before AIX can allocate large memory pages, the AIX kernel must be rebuilt.

2. The UNIX user running the shared page cache monitor must be configured with permission to use large memory pages.

3. You must enable GemStone/S 64 Bit to request large memory pages by setting an environment variable prior to startup.

Determining the required number of large memory pages

You will first need to determine how much space is needed for the shared page cache. This is a two-step process.

(1) The easiest way to get the required values 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 1TB (1000GB) cache. These are the values that are needed:

```
Number of pages              62500000
Number of processes          3066
TargetFreeFrameCount         5000
Number of shared counters    1900
```

(2) Once you have the GemStone parameters, start the shrpcmonitor directly and enter the parameters to compute the target size.

```
unix>$GEMSTONE/sys/shrpcmonitor

 <prints header information>

SHRPCMON>62500000 setnumpages
SHRPCMON>3066 setnumprocs
SHRPCMON>1900 setnumsharedctrs
SHRPCMON>5000 settargetfflimit
SHRPCMON>getrequiredsize
For 62500000 pages, 3066 processes and 1900 shared counters,
required cache size is 1076836515840 bytes.
```

In this example, the result of the computation is that the shared page cache will require 1076836515840 bytes.

Large memory pages are 16 MB, so divide 1076836515840 by the page size (16777216), rounding up. This gives a minimum of 64185 large memory pages that are required for a SPC with this configuration.

Configure AIX to use the required number of Large Memory Pages

AIX by default is not configured to allocate large memory pages. Before large memory pages can be used, AIX must be configured to use a specific number of large memory pages. These commands must be run as root.

the following example command configures AIX with 64185 large memory pages:

```
#vmo -r -o lgpg_regions=64185 -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

Enable Unix user to use Large Memory Pages

The UNIX user running the shared cache monitor must also 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
```

you can confirm large memory pages are available for use using vmstat -l

Enabling 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 use a line of the form:

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

GemStone Shared Memory Debugging

For debugging problems the shared memory, an environment variable has bee added. The SPC monitor process will print extra debugging information to its log file if the following is set:

export GS_DEBUG_SHARED_MEM=1

Configuration Parameter Changes

GEM_TEMPOBJ_CACHE_SIZE changes

Increased maximum for GEM_TEMPOBJ_CACHE_SIZE

Previously, the maximum for GEM_TEMPOBJ_CACHE_SIZE was 1000000 (1GB). This limit has been raised to 64000000 (64GB).

TOC on AIX now flexibly-sized without growable configuration

In previous versions, the GEM_TEMP_OBJ_CACHE_SIZE on AIX was configured to a base size using GEM_TEMPOBJ_INITIAL_SIZE, and was dynamically growable to the GEM_TEMP_OBJ_CACHE_SIZE limit. Now this is no longer necessary. The TOC has an initial smaller size based on the space needed, and can expand as needed to the upper limit, as is done on Solaris and Linux.

GEM_TEMPOBJ_POMGEN_PRUNE_ON_VOTE increased range

The previous upper limit of 80 for the configuration parameter GEM_TEMPOBJ_POMGEN_PRUNE_ON_VOTE has been increased to 100.

Added configuration parameter GEM_REPOSITORY_IN_MEMORY

For details, see "Performance improvements when entire Repository in SPC" on page 10.

GEM_REPOSITORY_IN_MEMORY

Determines the performance behavior of the gem for certain operations that scan the entire repository. If set to TRUE, the gem assumes most or all of the data pages in the repository have been previously loaded into the shared page cache. If set to FALSE, the gem assumes most or all of the data pages in the repository are not in the shared page cache and must be read from disk.

This setting affects performance only. All operations affected by this setting will succeed and produce the same results.

Methods affected by this setting are:
```
Repository >> findReferencePathToObject: (and related methods)
Repository >> listInstances: (and related methods)
Repository >> listReferences  (and related methods)
Repository >> _scanForParentObjects:
Repository >> markGcCandidatesFromFile: (and related methods)
Repository >> fullBackupTo: (and related methods)
Repository >> listInstancesInPageOrder: (and related methods)
Repository >> listObjectsInSegments: (and related methods)
Repository >> pagesWithPercentFree: (and related methods)
System class >> findObjectsLargerThan:
GsObjectInventory class profileRepository: (and related methods)
```

Note that the `markForCollection` and `findDisconnectedObjects` and related methods are NOT affected by this setting.

Runtime equivalent: #GemRepositoryInMemory
Default: FALSE

Bugs Fixed

The following bugs in 2.4.5.1 have been fixed in this release:

Backup while reclaim running had risk of error or corrupted backup

Under certain conditions, the Reclaim Gem may ignore the high water page sent by the stone, or not refreshthe high water page correctly. This can cause problems such as the backup gem receiving a page cache fault and/or a corrupted backup. (#42461, #42465)

Unsafe aborts by Reclaim gems

If the reclaim gem must pause during a transaction, it was aborting but not discarding the list of pages to reclaim, which was not safe. (#42470)

CommitRecordBacklog created by terminated session

It was possible for a session that is no longer logged in to hold the oldest commit record, causing a commit record backlog that could not be recovered normally (by terminating the session). The specific scenario involved a session killed by idle session timeout and exiting with a fatal error in such a way that the OOB socket was not properly terminated. (#42690)

Improved statmonitor tracking

In addition to ensuring terminated sessions are always entirely terminated, information is now recorded about session state in cache statistics, using GlobalStat43..47. Also, statmonitor will now continue to record session data until the internal sessionId, as well as the processId, is released.

During the session shutdown process, any application data in GlobalStat43...GlobalStat47 will be overwritten by shutdown tracking information.

Terminate of idle session may not have completed correctly

When an idle gem was killed, there was a possible race condition between the internal disconnect and the process kill of the session. This could result in the session's socket not getting the disconnect. The gem remained in AFTER_LOGOUT_PGSVR_WAIT, and continued to occupy its slot in the process table, though the OS process itself was gone. (#43205)

Zombie gem if Gem's pageserver recycled during gem termination

If a session makes a request of the stone but is terminated by the stone before that request is serviced, and if the terminated session's pageserver's PID is reused for a new pageserver before the stone gets to servicing the request from the terminated session, it may cause a zombie-gem situation, in which the commit record is held by a session that is not there, causing a commit record backlog. (#43211)

Error on stone startup after clean shutdown

When transaction logs are on raw partitions, and the stone is shutdown shortly after a new transaction log has been started, it was possible to get tranlog mismatch errors on restart, although there was no actual error. (#42771, #42441)

Memory leak in Stone with use of notifiers

Gem signalling allocates memory on the C heap of the Stone's process. When notifiers are used, in some cases this memory was not freed, resulting in growth of the stone memory footprint. This ultimately could cause the stone to run out of memory and crash with a malloc error. (#42463)

Login failures due to race condition between regular and OOB sockets

Changes have been made to avoid a possible race condition in which logins fail due to messages on the regular socket to the stone before the Out of band socket is setup. (#43506)

Large IdentityDictionaries may be corrupted during rebuild

Instances of IdentityDictionary must be rebuilt periodically as elements are added. There was a code path in rebuild that could cause values in key-value pairs to be set to Associations containing the key and value. This occurred in large IdentityDictionaries, and the specific conditions depend on how elements hash into the dictionary. (#43515)

Writing a large character to GsFile caused unexpected problems

GsFile can read and write byte characters only. If a Character larger than can be contained in one byte was passed to an instance of GsFile, for example using nextPut:, it did not fail or error, but a later GsFile or UserAction could encounter an error with a potentially confusing error message. (#42185)

Remote User Actions may fail if GCI compression enabled

The RPC protocol for certain packet types may fail with GCI compression, which may result in LGC protocol errors. (#41096)

Wrong sign bit for values returned by GciFltToOop()

On Windows, the function GciFltToOop() for a negative argument returns a value with the wrong sign (positive), but otherwise correct, as the result of a C compiler bug. (#39516)

Excessive creation of GsMethodDictionaries

An error in the internal calculation of dictionary tableSize caused code to be executed unnecessarily, in some cases creating many instances of GsMethodDictionary and degrading performance. (#43287)

Socket buffer sizes smaller than OS default

The internal socket buffer sizes were smaller than the OS default. This could impact performance. (#43364)

RcQueue>>changeMaxSessionId: may cause problems

The code implementing changeMaxSessionId: did not correctly initialize the new sections of the RcQueue, and could potentially result in later errors. (#42719)

Stone kill -TERM may hang if aio wait thread terminates early

If stone is sent a kill -TERM, which should do a clean shutdown, there was a risk that the aio wait thread would exit before the main thread, which left the main thread hanging. (#42776)

Image upgrade resets #sleepTimeBetweenReclaimMs to 0

On image upgrade, the configured value for the GcUser configuration parameter #sleepTimeBetweenReclaimMs (in the GcUser's UserGlobals) is reset to 0. (#43521)

Statmonitor started with -f could overwrite existing active file with same name

Starting statmonitor with the -f option allows you to specify a filename; this overwrite any existing file with that name. If two statmonitor processes were started specifying the same file name, both wrote to this same filename, which corrupts the file. (#42913)

Now, it is an error to specify an existing file with the -f option.

Admin GcGem risk of hang in Write Set Union

There is a small risk that the Admin GcGem may hot hang during a WSU sweep. (#42057)

findReferencePath* returned nil on invalid objects

If the argument to Repository>>ffindReferencePathToObject: and related methods was a special or an uncommitted object, it returned nil rather than an appropriate error. It now triggers error 2027; the text of this error has been updated.

Stone crash during recovery

Changes in this release add additional thread-safety and buffer protection to recovery. (#40004)