GemStone®

GemStone/S Release Notes

Version 6.6

September 2011

vmware®

GEMSTONE S™

INTELLECTUAL PROPERTY OWNERSHIP

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

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

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

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

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

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

COPYRIGHTS

PATENTS

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

TRADEMARKS

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

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

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

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

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

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

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

Linux is a registered trademark of Linus Torvalds and others.

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

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

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

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

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

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

Preface

About This Documentation

These release notes describe changes in the GemStone/S version 6.6 release. We recommend that everyone migrating to this version read these release notes before beginning installation, testing or development.

For information on installing or upgrading to this version of GemStone/S, please refer to the *GemStone/S Installation Guide* for version 6.6.

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

Technical Support

GemStone's Technical Support website provides a variety of resources to help you use GemStone products.

Documentation for released versions of all GemStone products is provided in PDF form on this website:

GemStone Web Site: http://support.gemstone.com

Documentation is also available at

http://www.gemstone.com/documentation

In addition to documentation, the GemStone support website provides:

- Downloads and Patches, including past and current versions of GemBuilder for Smalltalk.
- 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.

Requests for technical assistance may be submitted online 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.gemstone.com

Email: techsupport@gemstone.com

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

Your GemStone support agreement may identify specific designated contacts who are responsible for submitting all support requests to GemStone. If so, please submit your information through those individuals.

If you are reporting an emergency by telephone, select the option to transfer your call to the Technical Support administrator, who will take down your customer information and immediately contact an engineer. Non-emergency requests received by telephone will be placed in the normal support queue for evaluation and response.

When submitting a request, please include the following information:

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

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

24x7 Emergency Technical Support

GemStone 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.

Contents

Chapter 1

GemStone/S 6.6 Release Notes

Overview

GemStone/S 6.6 is a new version of the GemStone 32-bit Smalltalk object server. This release provides new features and fixes a number of bugs; we recommend everyone using GemStone/S upgrade to this new version. The details of these changes are provided in this document.

These release notes provide changes between the previous version of GemStone/S, version 6.5.8, and version 6.6. If you are upgrading from a version prior to 6.5.8, please also review the release notes for each intermediate release to see the full set of changes.

For details about installing GemStone/S 6.6 or upgrading from earlier versions of GemStone/S or other GemStone server products, see the *GemStone/S Installation Guide* for version 6.6 for the appropriate platform.

Supported Platforms and GBS Versions

Platforms

GemStone/S version 6.6 is supported on the following platforms:

- Solaris 8, 9 and 10 on SPARC
- AIX 5.3 and AIX 6.1, POWER5 or later
- Red Hat Linux ES 5.0 and 5.5
- Windows XP, Windows 2003, Windows Vista, Windows 7, and Windows 2008

For more information and detailed requirements for each supported platforms, please refer to the GemStone/S v6.6 Installation Guide for that platform.

GBS versions

The following versions of GBS are supported with GemStone/S version 6.6, with the following client Smalltalk and platforms versions.

GBS version 7.4

VisualWorks 7.8 with 7.8 OE	**VisualWorks 7.7.1** with 7.7.1 OE
▸ Windows XP, Windows 2003 Server, Windows Vista, and Windows 7 ▸ Solaris 9 and 10 on SPARC ▸ RedHat Linux ES 5.0 and 5.5	▸ Windows XP, Windows 2003 Server, Windows Vista, and Windows 7 ▸ Solaris 9 and 10 on SPARC ▸ RedHat Linux ES 5.0 and 5.5

GBS version 7.3.3

VisualWorks 7.7.1 with 7.7.1 OE	**VisualWorks 7.7** with 7.6c OE
▸ Windows XP, Windows 2003 Server, Windows Vista, and Windows 7 ▸ Solaris 9 and 10 on SPARC ▸ RedHat Linux ES 5.0 and 5.5	▸ Windows XP, Windows 2003 Server, Windows Vista, and Windows 7 ▸ Solaris 9 and 10 on SPARC ▸ Red Hat Linux ES 5.0

GBS version 5.2.7

VA Smalltalk 8.0.2	**VA Smalltalk 7.5.2**
▸ Windows 7, Windows Vista, Windows 2003 Server, and Windows XP	▸ Windows XP and Windows 2003 Server

For more information and details, see the GemStone/S Installation Guide for 6.6, chapter 3, or the GemBuilder for Smalltalk Installation Guide for that version.

Changes and New Features

Updated Documentation

This release includes updated *System Administration Guide for Unix, System Administration Guide for Windows*, and *GemBuilder for C* manuals.

Updated Compile/Link information

There are a number of platform-specific changes in supported OS versions, OS kernel/patch versions, and in compiler and debugger versions, and changes to compile and link flags.

If you compile user actions or GemBuilder for C applications, please review the System Requirements section of the *GemStone Installation Guide* for v6.6, and Chapter 4 of the v6.6 *GemBuilder for C* manual.

Improvements to markForCollection and findDisconnectedObjects

Multi-threaded markForCollection and findDisconnectedObjects.

This release of GemStone/S includes support for multi-threaded mark/sweep operations. This is a different, more limited multi-threaded operation than is used in the GemStone/S 64 Bit product.

The multi-threaded garbage collection operations are not supported on Windows, and must be run from a gem that is local to the Stone.

Multi-threaded garbage collection algorithms specify the page buffer size and the number of read threads per page buffer. There are always two page buffers; the resulting memory requirement and threads used is calculated as two times the arguments specified.

So for example, if you specify 4 threads as the argument, the final number of threads will be 8, plus the main and stats threads. The main thread does not read pages into the buffers and is not included in these counts.

Likewise, the pages allocated per buffer are per buffer, so total page allocation is twice as much as the argument specified. To avoid running out of address space this must be taken into account when calculating memory budgets for the multi-threaded garbage collection gem.

Multi-threaded garbage collection always operates in "safe" mode; OT lookups are performed on every non-swept object on every data page.

Multi-threaded garbage collection will respond to SigAbort requests from the stone, but it may take significantly longer, particularly when using larger buffer sizes. In repositories that are in use, it may cause s significant commit record backlog. The stone will not send a LostOtRoot signal to a session while it is performing multi-threaded garbage collection.

Methods that support multi-threaded garbage collection

The following public methods have been added to support multi-threaded garbage collection:

`Repository >> markForCollectionMt`
Performs a multi-threaded markForCollection with the default settings:
- 640 total pages buffered (320 per page buffer)
- 4 read threads (2 per page buffer)
- wait 120 seconds for the GC lock

To specify values other than the default, use `markForCollectionWaitMt:pageBufferSize:threads:`.

`Repository >> markForCollectionWaitMt:` *waitTimeSeconds*
Performs a multi-threaded markForCollection with the default settings, waiting the specified number of seconds for the GC lock to become free.

`Repository >> markForCollectionWaitMt:` *waitTimeSeconds*
`pageBufferSize:` *numPages* `threads:` *numThreads*
This method performs the same function as **`markForCollectionWait:`** , but with multiple threads and the specified page buffer size.

`Repository >> findDisconnectedObjsMtWithPageBufferSize:` *numPages*
`threads:` *numThreads*
Performs a multi-threaded findDisconnectedObjects.

`Repository >> supportsMtGc`
Answer true if the multi-threaded garbage collection may be run by this session, otherwise answers false.

Multi threaded algorithms and offline GC

The GemStone/S distribution includes unsupported user action code which performs offline mark/sweep, findDisconnectedObjects. These user actions also include multi-threaded algorithms, accessed vis the new user action #uaRunMtFdcAndWriteOopsToFile. This user action takes 3 arguments.

On UNIX platforms, the distribution now includes shared library binaries for the user actions, under the new directory $GEMSTONE/examples/offlinegc/lib.

Other Improvements to mark/sweep algorithms

These improvements apply to both single threaded and multi-threaded garbage collection.

- In safe mode, or in the multi-threaded garbage collection (which always operates in safe mode), object table look-ups are not performed for objects which have already been swept.
- leaf object handling is optimized. Leaf objects are objects which by definition are "leaves" in the object graph, i.e.: cannot reference any other objects except for the class or segment. Small byte objects with no tags fall into this category as do small zero-length OOP objects with no tags. Once a leaf object is marked (i.e., a live object is found to reference it), it can be added to the swept set immediately without being

read again, since the object cannot reference any other objects. The benefit from this optimization is dependent on the percentage of leaf objects in the database.

Repository >> safeMarkGcCandidates removed

The method `Repository >> safeMarkGcCandidates` has been removed. All markGcCandidate operations are now performed safe (with validation), since the overhead to do so is slight. The extra look-up required for validation is only done if there is a non-candidate object referencing a GC candidate, and correct lists should have no or very few of these.

Improved management of AllSymbols

Cleanup of dead Symbols

A method has been added that allows the repository to be scanned for unused Symbols; that is, symbols whose only reference is from the AllSymbols collection. These symbols can be optionally then also removed from the system.

In addition, this method will detect non-canonical symbols (Symbols that are not in the AllSymbols collection) and report them, for later manual repair.

This operation, with or without performing the removal, can only be performed by SystemUser, and runs in transaction for its entire duration. It is intended to be run in non-production environments.

`Repository >> cleanupDeadSymbols:` *readOnlyBoolean*

Scan the repository to locate Symbols which are referenced only by the AllSymbols collection.

If the argument *readOnlyBoolean* is true, the operation is read-only and no changes are made to the repository, the method does not commit.

If the argument *readOnlyBoolean* is false, AllSymbols is write-locked for the duration of the method; no symbols can be created or removed while this method is running. The dead symbols are removed from AllSymbols, and the method commits one or more transactions. This results in non-canonical symbols, which are removed after the next garbage collection cycle.

This method returns an Array of 5 Integers:

1 - Number of live symbols found.
2 - Number of dead symbols found; theses symbols are stored in hidden set 26.
3 - Number of non-canonical symbols found; these symbols are stored in hidden set 27.
4 - Number of repairable non-canonical symbols found.
5 - Number of non-repairable, non-canonical symbols found; these symbols are stored in hidden set 28.

Improved handling of AllSymbols bitmap rebuild on login

The first GcGem to login to a stone rebuilds the AllSymbols bitmap. If another GcGem logs in, rather than rebuilding the bitmap, it will check with the stone if rebuild is required. If not, an entry like this will appear in the GcGem log file:

```
[Info]: Deferred building of AllSymbols bitmap.  Bitmap already
built by another GcGem.
```

Tuning AllSymbols bitmap rebuild

Two new GcUser parameters have been added to control the frequency of the rebuild of the AllSymbols bitmap:

#allSymbolsMinTimeBetweenBmRebuilds
: The minimum interval in seconds between AllSymbols bitmap rebuilds performed by the GcGem or EpochGcGem.
Minimum: 5
Maximum: 536870911
Default: 60

#allSymbolsMinCommitsBetweenBmRebuilds
: The minimum number of total commits to the repository between AllSymbols bitmap rebuilds performed by the GcGem or EpochGcGem. A value of 0 disables symbol bitmap rebuilds except immediately after the first GcGem/EpochGcGem login after the stone is started.
Minimum: 0
Maximum: 536870911
Default: 500

AllSymbols manual rebuild

It is now possible to manually rebuild the AllSymbols dictionary, using the new public method:

```
System class >> rebuildAllSymbols:createAllBuckets:
```

AllSymbols is now also rebuilt more efficiently, using hidden sets rather than array.

UserProfile setting to avoid commit on login and stale account check

When either stale account aging or password aging is turned on (UserProfileSet passwordAgeLimit: or staleAccountAgeLimit: set to a non-zero value), each session's login performs a commit to update the lastLoginTime.

It is now possible to set individual accounts to not commit this update to the lastLoginTime on login, and to be immune from being disabled due to no login for a time exceeding the stale account age limit, when the repository otherwise has stale account or password age checks enabled.

In addition to avoiding account disable due to stale account for specific accounts, this setting will improve login time, particularly for cases where there are multiple simultaneous logins by the same account.

Note that lastLoginTime will not be correct for users with disablePasswordAgeChecks. If checks are reenabled, the account may be disabled immediately, if lastLoginTime is not manually updated (see "Updating lastLoginTime" on page 13)

The following methods have been added:

```
UserProfileSet >> findProfilesWithStaleAccountChecks
```
: Return an IdentitySet containing all UserProfiles in the receiver which do not commit on login and have stale account checks disabled.

`UserProfile >> disableStaleAccountChecks`
Disables the update to lastLoginTime on login, and disable stale account checks for the receiver only.

`UserProfile >> enableStaleAccountChecks`
Enable the update to lastLoginTime on login (if stale account or password aging is enabled in the repository), and enable stale account checks for the receiver only.

`UserProfile >> hasDisabledStaleAccountChecks`
Answer true if commit of lastLoginTime on login, and stale account checks, have been disabled for the receiver, false otherwise.

Updating lastLoginTime

The UserProfile's lastLoginTime is not updated automatically by the system unless password aging or stale account aging is enabled. This avoids the overhead of an extra commit at each login. However, initially enabling stale account aging can mean that existing accounts are disabled immediately.

To avoid this, you can manually execute the following method for each account that is subject to stale account aging, using the current DateTime (**`DateTime now`**), or nil. Accounts with a nil value for lastLoginTime do not fail aging tests.

`UserProfile >> lastLoginTime:` *aVal*
Updates the lastLoginTime for the receiver to be *aVal*, which must be an instance of DateTime or nil.

Improvements to backup and restore

Support for backup operations on NFS

Previously, any operations performed over NFS were considered remote, and disallowed without netldi-brokered communications.

Now, writing backups to NRS locations, or restoring backups or tranlogs from NFS locations, is supported.

Compressed tranlog restore performance improvement

Compressed tranlogs have historically taken much longer to restore than the same, uncompressed tranlog. Using improved buffering, the performance has been increased greatly. Now, the time taken to restore a compressed tranlog will be similar to the time to restore the same uncompressed tranlog.

Backup files contain version information

Backup files now include information regarding the GemStone version that created them. Copydbf has been enhanced to report this information. Now, when using copydbf -i or -I to get information about a backup file from version 6.6 or later, the results include the version of GemStone/S that created the backup. This makes it simpler to determine the GemStone version required to restore the backup, since backups should be restored in the same version in which they were created.

The additional output of copydbf includes a line similar to:

```
Backup was created by GemStone Version: 6.6.0 .
```

Method to determine session holding GC Lock

A method has been added to retrieve the session ID of the session that is holding the GC Lock.

`System class >> sessionIdHoldingGcLock`
Returns the session ID of the session currently holding the garbage collection lock, or 0 if the GC lock is free.

Other added methods

The following methods have been added to support features added in this release.

`System class >> sessionIsRemote`
Answers true if this session is running on a different host than the stone, false otherwise.

`System class >> sessionIsLocal`
Answers true if this session is running on the same host as the stone, false otherwise.

`System class >> sessionIsUNIX`
Answers true if this host operating system for this session is a type of UNIX, false otherwise.

`Object >> levels`
Answer the number of levels in the receiver. Special objects and small POM objects have a levels of zero. Large objects have levels of 1 or 2.

`Object >> isLarge`
Answer a Boolean indicating if the receiver is a large object (an object that spans across more than 1 disk page).

Added Cache Statistics

The following cache statistics have been added:

GcLockSession (Stone)
session ID holding the GC lock or 0 if the GC lock is free.

StoredPomObjsMapSize (Gem)
Number of objects in the stored POM objects map.

StoredLomObjsMapSize (Gem)
Number of objects in the stored LOM objects map.

Bugs Fixed

The following bugs in version 6.5.8 are fixed in version 6.6.

Hole in voting may have resulted in corruption

After a mark/sweep has completed, the voting step handles references to objects from sessions that were logged in during the mark/sweep. If a session creates a temporary reference to a persistent object X, removes all references to X from other persistent objects, and commits, the analysis of references to X during voting will incorrectly determine that X is not referenced. This allows X to be garbage collected. If the session then commits a persistent reference to X, it creates repository corruption. (#41730)

To fix this bug, Gems now do additional tracking of stored references using bitmaps.

Such referenced objects cannot be tracked based on transaction logs. As a side effect of this fix, after a stone recovery, or after tranlog restore, the possible dead objects must be discarded unless the tranlogs also contain the promote to dead.

Two new cache statistics have been added for tracking, **StoredPomObjsMapSize** and **StoredLomObjsMapSize**; see page 14 for details.

Page leak in Reclaim GcGem

The Reclaim GcGems were subject to a slow page leak. (#41834)

Safe backup may have produced corrupt backup files

Protocol to perform backups with validation, `safeFullBackupTo:`, were added in version 6.5.7.5; this code validates objects that are backed up, and does not backup invalid objects.

This safe backup code failed to handle a case where an object had been modified by another session since the backup began, and the backup session aborted due to a SigAbort. While the object is now a shadow, it still must be written to the backup since it was current when the backup began. However, the safe backup incorrectly determined that it was a missing shadow object and omitted it from the backup. (#41295)

Chance of duplicate keys for shared memory segments

GemStone uses ftok() to acquire a key for a shared memory segment, which is used by all processes that need to attach to that segment. Since ftok() uses only 16 bits of the inode, as well as the device and constant, there was a small risk that the shared memory key could be created for two repositories running on the same host. This required multiple repositories on the same host. To reduce the risk, instead of the constant, ftok() is passed a randomly generated value. The resulting key is stored in the SPC monitor lock file. (#41642)

Shared cache monitor lock file regeneration

If GemStone process lock files (*filename*`..LCK`) are deleted, normally they are recreated within a minute. The shared page cache monitor lock previously did not regenerate itself; now it will do so. (#41586)

Gems unnecessarily built private copy of AllSymbols bitmap

As a result of a code error in the fix for #41345 in version 6.5.8, Gems in version 6.5.8 built private copies of the AllSymbols bitmap periodically. This is unnecessary and has a negative effect on performance. (#41610)

SortedCollection includes: may have failed if = and <= defined differently

SortedCollection optimized the `includes:` method to by using a binary search based on the sortBlock; this assumed that any element that equals the search object would be found at the sorted location for that element.

If the elements in the SortedCollection implement =, and the SortedCollection defines a sortBlock that uses different criteria (such as a different instance variable than used for the = method), `SortedCollection>>includes:` may fail to find an element that is in the collection. (#41013)

To correct this, the `includes:` method now performs a linear search, rather than a binary search. This may result in slower performance for `SortedCollection>>includes:`, particularly on larger, persistent SortedCollections. To get the behavior of previous releases, use the new method **`SortedCollection>>`**`binarySearchIncludes:`.

RcQueue changeToSegment:

Sending RcQueue the changeToSegment: message changed the top level object, but not the components of the RcQueue. Now, RcQueue implements changeToSegment:, which also updates the Segments of the RcQueue components. (#41004)

findString:startingAt: incorrect for empty string argument

When `CharacterCollection >> findString:startingAt:` was called with a search argument of an empty string '', it returned 1, rather than 0. (#39922)