GemStone®

GemStone/S 64 Bit™ Release Notes

Version 3.1

July 2012

vmware®

GEMSTONE S™ 64

INTELLECTUAL PROPERTY OWNERSHIP

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

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

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

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

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

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

COPYRIGHTS

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**, **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. 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 64 Bit™ version 3.1 release. Read these release notes carefully before you begin installation, conversion testing, or development with this release.

For information on installing or upgrading to this version of GemStone/S 64 Bit, please refer to the *GemStone/S 64 Bit Installation Guide* for version 3.1.

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; the GemStone Smalltalk programming language; and may also be used to refer to the company, previously GemStone Systems, Inc., now a division of VMware, Inc.

Technical Support

GemStone Website

http://support.gemstone.com

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

- **Documentation** for released versions of all GemStone products, in PDF form.
- **Downloads** and **Patches**, including past and current versions of GemBuilder for Smalltalk.

- **Bugnotes**, identifying performance issues or error conditions 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.gemstone.com

Email: techsupport@gemstone.com

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

When submitting a request, please include the following information:

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

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

Training and Consulting

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

Contents

Chapter 1. GemStone/S 64 Bit 3.1 Release Notes

Overview **11**

Supported Platforms and GBS Versions **12**

Platforms 12
GBS versions 12

Documentation **12**

Changes and New Features **13**

Secure RPC logins using SSL 13
New shared library required 13
Library naming convention changed 14
GBS library loading 14
Smalltalk methods to get library name 14
Support for multiple network interfaces 14
IPv6 support 14
Changes in class comment handling 15
Full support for upgrade of Seaside images from 2.x or 3.0.x to 3.x 16
Backup and restore reimplemented as multi-threaded 16
Improvements to tranlog replay 17
Logins during tranlog replay 17
Symbol garbage collection 17
New hot standby functionality 17
Support for SSL sockets from Smalltalk 18
Message Digest Functions 19
GsSocket added methods to manage blocking 19
Code invoked on session login 19
Nested in-memory transactions 20

Support for UTF and Locale-specific collation using ICU 21
IcuLocale 22
New Unicode String classes 22
IcuCollator 22
IcuSortedCollection 23
GsFile reading and writing 23
Changes in encode/decode UTF8 and decodeIntoDoubleByteStringFromUTF8 23
Hidden Sets public API. 23
Adding application-specific information to an instance of GsProcess 24
Additional protocol to enable/disable signalling 24
Garbage Collection Enhancements 26
Ability to defer reclaim under low free space conditions 26
reclaimAll now performs postReclaimAll 26
Further optimizations to MFC 26
New markForCollection variant with buffer size 27
RcIdentityBag >> _unsafeCleanupBag added 27
descriptionOfSession: now includes gem type, temporary OOPs and pages. . . 27
GciInterface changes 27
New login method nbLogin:flags: 27
Distinguishing OOP from Integers in results. 27
Change in return class for Set and IdentitySet 27
Privilege Changes 28
New privilege to allow non-SystemUser to compile primitive methods . 28
FedAdminPrivilege renamed 28
OtherPassword now required to modify another users's defaultObjectSecurityPolicy 28
GsObjectSecurityPolicy new now allowed 28
Repository>>name: method removed 28
Direct IO for tranlogs now only available on Solaris 28
OS level System stats for processes now available from Smalltalk 28
Improved error reporting for remote cache loss 29
dictionaryAndSymbolOf: deprecated 29
Changes in messages and log file output 29
Disallow block device raw partitions 29
Improvement to communication between Page Manager and remote caches. . . 29
true, false, and nil now allowed as message selectors. 29
Limitation on native code on Mac 29
FFI changes 29
Removed methods 30
cachewarmer will run in recovery/restore 30

Distribution Changes **31**

Cache Statistics Changes **31**

Configuration Parameter Changes **32**

Configuration parameters queries from image now return consistent true/false 32

STN_TRAN_LOG_DIRECTORIES and STN_TRAN_LOG_SIZES, requirements change and related image changes . . . 32
Default added for STN_TRAN_FULL_LOGGING . . . 32
Changes in default for STN_FREE_SPACE_THRESHOLD . . . 33
Changes in STN_LOG_IO_FLAGS . . . 33
Changes in default for SHR_NUM_FREE_FRAME_SERVERS . . . 33
GEM_TEMPOBJ_CACHE_SIZE changes . . . 33
Increased maximum for GEM_TEMPOBJ_CACHE_SIZE . . . 33
TOC on AIX now flexibly-sized without growable configuration . . . 33
GEM_HALT_ON_ERROR now has runtime equivalent . . . 33
Added configuration parameters . . . 34
GEM_RPC_USE_SSL . . . 34
GEM_TEMPOBJ_START_ADDR . . . 34
SHR_PAGE_CACHE_PERMISSIONS . . . 34
STN_LISTENING_ADDRESSES . . . 34
STN_SYMBOL_GC_ENABLED . . . 35

Changes in Utilities *36*

New utility commands . . . 36
copydbf changes . . . 36
copydbf now allowed on extents that are in use . . . 36
copydbf -i/-I output omits destination . . . 36
copydbf -i/-I includes backup version . . . 36
Statmonitor new option to collect samples for only stone and spcmon . . . 36

Topaz Changes *37*

Terminology clarification . . . 37
Saved stacks . . . 37
STK deprecated . . . 37
Improved handling of pasting large texts . . . 37
Handling of errors in .topazini . . . 37

GCI changes *38*

Added GciNbEnd_ . . . 38

Compile and Link Information *38*

Linux . . . 38
Compiler version . . . 38
Debugger version . . . 38
Compiling a user action or GCI application . . . 38
Linking user actions into shared libraries . . . 38
Linking GCI applications . . . 38
Solaris on Sparc . . . 39
Compiler version . . . 39
Debugger version . . . 39
Compiling a user action or GCI application . . . 39
Linking user actions into shared libraries . . . 39

Linking GCI applications . . . 39
Solaris on x86 . . . 39
Compiler version . . . 39
Debugger version . . . 39
Compiling a user action or GCI application . . . 39
Linking user actions into shared libraries. . . . 39
Linking GCI applications . . . 39
AIX . . . 40
Compiler version . . . 40
Debugger version . . . 40
Compiling a user action or GCI application . . . 40
Linking user actions into shared libraries. . . . 40
Linking GCI applications . . . 40
Darwin . . . 41
Compiler version . . . 41
Debugger version . . . 41
Compiling a user action or GCI application . . . 41
Linking user actions into shared libraries. . . . 41
Linking GCI applications . . . 41
Windows . . . 41
Compiler/Debugger version . . . 41
Compiling GCI applications . . . 41
Linking GCI applications . . . 41

Bugs Fixed ***42***

Support for AIX on POWER7 . . . 42
GciInterface >> remotePerform:selector:args: failed with OOPs of non-specials 42
SIGTERM can cause fatal stuck spin lock. . . . 42
Failure in pregrow of dynamically added extent causes extent corruption. . . . 42
Hole in voting may result in corruption . . . 42
Removing element from KeyValueDictionary could result in incorrect size . . . 42
Tranlog generation and restore issues . . . 43
Risk of bad tranlog record written with nonzero STN_GEM_TIMEOUT . 43
Memory leak in tranlog restore . . . 43
Commit Record Backlog during restore. . . . 43
Replayed persistent counter changes not visible until commitRestore . . 43
Tranlog close could have truncated log, resulting in corrupted tranlogs . 43
Shared counter changes recorded too soon in transaction logs. . . . 43
Restore operations logouts reported as errors in RPC logs . . . 43
Upgraded repositories did not have correct PositionableStream classes . . . 44
Numerics issues . . . 44
ScaledDecimal // could round in the wrong direction . . . 44
0.0 negated returned 0.0, not -0.0 . . . 44
Float roundTo: returns a SmallInteger 0 rather than a Float . . . 44
Subnormal Large Integers not converted by + 0 . . . 44
ScaledDecimal rounding bias . . . 44
to:by:do: handling of zero by: argument . . . 44

Busy gem does not notice client's death 44
Allow %Z timezone specifier in GS_CFTIME and enforce format 45
Slow checkpoints with multiple AIO pgsvrs and fewer Free Frame pgsvrs 45
Writing a large character to GsFile caused unexpected problems 45
Unclear errors on failures in setting IO flags 45
Error from validatePasswordForUser:password:isEncrypted: 45
Corrupt object error referred to nil rather than actual object 45
Linux host stats computed incorrectly 46
GCI application wrote login information to stdout 46
canUnderstand: could introduce changes, affecting commit state 46
Invalid attempt to use Stone's cache as mid-level cache crashed rather than errored 46
searchlogs script could fail with large number of tranlogs 46
Classes subclassed from nil were versioned on recompile with no changes. . . . 46
Recompile of unconverted GsMethods failed 46
GcGems deleted log files with #verboseLogging enabled 46
Missing error message for #'objErrDiffSizeColl' 47
UserProfile privileges cleanup 47
PublishedSegment missing from Globals 47

Chapter 1

GemStone/S 64 Bit 3.1 Release Notes

Overview

GemStone/S 64 Bit 3.1 is a new version of the GemStone/S 64 Bit object server. This release provides many feature enhancements and fixes a number of bugs; we recommend everyone using GemStone/S 64 Bit upgrade to this new version.

These release notes provide changes between the previous version of GemStone/S 64 Bit, version 3.0.1, and version 3.1. If you are upgrading from a version prior to 3.0.1, review the release notes for each intermediate release to see the full set of changes. In particular, if you are upgrading from version 2.4.x, note that there were substantial changes in v3.0 that impact your application.

GemBuilder for Smalltalk applications will also need to upgrade. Version 3.1 requires GBS version 7.5 or later.

For details about installing GemStone/S 64 Bit 3.1 or upgrading from earlier versions of GemStone/S 64 Bit, see the *GemStone/S 64 Bit Installation Guide* for version 3.1.

Supported Platforms and GBS Versions

Platforms

GemStone/S 64 Bit version 3.1 is supported on the following platforms:

- Solaris 10 on SPARC
- Solaris 10 on x86
  v3.1 will not run on Solaris 9 or earlier.
- AIX 6.1, TL1, SP1, and AIX 7.1
- SuSE Linux ES 10 SP1 and ES11 on x86;
  Red Hat Linux ES 5.0, 5.5, and 6.1, on x86
- Mac OSX 10.6.4 (Snow Leopard), with Darwin 10.4.0 kernel, on x86

For more information and detailed requirements for each supported platforms, please refer to the GemStone/S 64 Bit v3.1 Installation Guide for that platform.

GBS versions

The following version of GBS is supported with GemStone/S 64 Bit version 3.1. Note that versions of GBS earlier than 7.5 cannot log in to GemStone/S 64 Bit v3.1.

GBS version 7.5

VW 7.9 32-bit	VW 7.9 64-bit
▸ Windows 2008 R2, Windows 7, Windows 7 64-bit, and Windows XP ▸ Solaris 10 on SPARC ▸ SuSE Linux ES 10 and 11, and RedHat Linux ES 5.5 and 6.1	▸ Solaris 10 on SPARC ▸ SuSE Linux ES 10 and 11, and RedHat Linux ES 5.5 and 6.1

For more details on supported GBS and client Smalltalk platforms and requirements, see the *GemBuilder for Smalltalk Release Notes* for version 7.5.

Documentation

GemStone/S 64 Bit version 3.1 includes an updated *System Administration Guide* and *Programming Guide*, with details on the new features in this release.

The Topaz and GemBuilder for C manuals have not been updated for this release, as changes in them are minor. See page 37 for Topaz changes and page 38 for GCI changes and compile and link updates.

Changes and New Features

Secure RPC logins using SSL

All RPC logins now use Secure Socket Layer (SSL) and Secure Remote Password (SRP) to establish the initial connection between the GCI client and gem and to authenticate passwords. Passwords are now stored in GemStone in the encrypted form used by SRP.

GemStone now includes and uses OpenSSL libraries. OpenSSL is a commonly used open-source implementation of the SSL protocol. OpenSSL v1.0.1 is used in v3.1. For more information, see `www.openssl.org`.

For more information on SRP, see `srp.stanford.edu`.

During the login process, the GCI client (topaz, GBS, or a GCI application) now always establishes an SSL connection to the gem. Connection failures will be retried up to 4 times.

If the client and gem are on the same machine, communication reverts to a normal socket connection after the login completes.

If the GCI client is remote from the gem, then communication continues on the secure socket, unless the new configuration parameter GEM_RPC_USE_SSL is set to false in the gem's configuration file. This allows communication over a network to be secure, although there is a slight overhead introduced by encryption and decryption of data. This encryption/decryption overhead is low since the shared key established during login can be used.

The login does not require any certificates since SRP does not need them. This avoids the need to manage certificates in order to establish a secure login.

Note that passwords in upgraded repositories remain encrypted using the pre-3.1 encryption. To further protect passwords for existing accounts, each user in the repository should update their password; this will store the changed password in the new encryption.

The openssl executable is included in the $GEMSTONE/bin directory, which may assist in debugging SSL issues.

New shared library required

To support SSL logins, new libraries are supplied in the $GEMSTONE/lib and $GEMSTONE/lib32 directory (on Windows, %GEMSTONE%\bin), with the base name `libssl`.

This library must be available, along with the regular shared library, to allow RPC logins to complete. If the ssl library is not available, a new fatal error, ERR_SSL_LIB_LOAD_FAILED (4150), is raised.

The gem will search for the SSL library in a number of places:

- a directory set by the new environment variable "GEMSTONE_SSL_LIB_DIR"
- The directory containing the GCI library. This would be the most common location.
- a lib subdirectory of the $GEMSTONE directory. For 64-bit clients, it looks in $GEMSTONE/lib, and for 32-bit clients, it looks in $GEMSTONE/lib32
- The current working directory

A new environment variable, GS_DEBUG_SSL_LOG_DIR, has been added to help debugging SSL library load issues. If this variable is set to a directory for its value, a process will create a file named GsSslDebug_<pid>.log with the output of any SSL calls.

To make testing easier, the GemStone distribution now includes 32-bit topaz executables, in the new $GEMSTONE/bin32 directory. If you have problems with loading the SSL libraries into a 32-bit GBS application, testing with 32-bit topaz will more closely duplicate the GBS login environment.

Library naming convention changed

There is a new naming convention for shared libraries. This is to avoid confusion between the 64-bit and 32-bit shared libraries and the 32-bit GemStone/S product. Now, the name is composed as <name>-<gs-version>-<width>.<ext>. So for example, the library that in v3.0.1 was named:

```
libgcirpc64-301.so
```

in v3.1 is named:

```
libgcirpc-3.1.0-64.so
```

This is designed to make it clear that the '64' or '32' refers to whether this library can be linked to 32-bit or 64-bit processes, and does not refer to the GemStone/S 32-bit or 64-bit server products.

The symbolic links previously included in the $GEMSTONE/lib directory are no longer part of the distribution.

GBS library loading

GBS application must upgrade to version 7.5 in order to log into GemStone/S 64 Bit v3.1. GBS v7.5 has new protocols for locating shared libraries. The new process is described in detail in the *GemStone/S 64 Bit Installation Guide* for v3.1, and in the *GemBuilder for Smalltalk Installation Guide* for v7.5.

Smalltalk methods to get library name

The following methods have been added to determine the name of the shared library:

```
System class >> fetchLinkedGciLibraryName
System class >> fetchRpcGciLibraryName
```

For details, see method comments in the image.

Support for multiple network interfaces

With the configuration parameter STN_LISTENING_ADDRESSES, the stone can selectively listen for login connections on machines with multiple network interfaces. By default, the stone listens on the wildcard address, which means it will listen on any active network interface and the loopback address.

See "STN_LISTENING_ADDRESSES" on page 34 for more details.

IPv6 support

Gemstone uses the IPv6 networking stack by default on Linux, Solaris and AIX. On those platforms, IPv4 addresses are represented internally using IPv4-mapped IPv6 addresses.

A bind or connect which fails using the IPv6 networking stack will be retried using IPv4 if the address is a IPv4-mapped IPv6 address.

On Macintosh, the IPv4 networking stack is used unless the address is not representable as an IPv4 address.

Changes in class comment handling

In version 3.0, GemStone deprecated class comments that used instances of the internal class GsClassDocumentation and related classes, which were stored in the class and accessed via description/description:. Instead, class comments were implemented as Strings returned by a class method #comment.

To avoid complications relating to inheritance, and for compatibility with other Smalltalk dialects and Seaside, comment handling has been changed again in v3.1.

Now, comments are stored in the class in string form and accessed via comment/comment:. Technically, comments are stored at the key #comment in the extraDict of the class instance. Comments are now written out by fileout, and recreated on filein.

The following methods have been added:

```
Class >> comment
Class >> comment:
```

The following methods are deprecated:

```
Class >> description
Class >> description:
Class >> _description
Class >> _description:
```

During the conversion or upgrade process, GemStone kernel class comments are installed per the new handling.

To upgrade class comments for application Classes, the script `upgradeComments` is provided. This should be run as part of the upgrade process. If comments are not upgraded, comment handling in GBS browsers may be inconsistent. More details on upgrading comments are provided in the *GemStone/S 64 Bit Installation Guide* for v3.1.

Note that, if you have class methods named #comment on your application classes, running the `upgradeComments` script sets the class comment to the string returned by this method, and deletes the #comment method. `upgradeComments` will also set the comment based on an existing GsClassDocumentation.

If you have class methods named #comment that do not return a String, `upgradeComments` will fail. You should rename these methods. No user classes should implement class methods named comment or comment:. If they do so, the interface from GBS to manage comments, and potentially other GemStone code, may not behave correctly.

You may also directly call the new method `Behavior >> upgradeComments`.

The new method `Behavior >> fileOutCommentsOn:` is used by fileout to file out class comments.

Full support for upgrade of Seaside images from 2.x or 3.0.x to 3.x

The upgrade for GLASS/Seaside applications from 2.x and 3.0.x to 3.1 is now available, and scripts to support upgrade for specific application requirements are provided in version 3.1.

Detailed instructions for upgrading Seaside applications are included in the *GemStone/S 64 Bit Installation Guide* for v.3.1.

Backup and restore reimplemented as multi-threaded

Both the interface and the internal behavior of full backup and restore from full backup has been changed in this release.

Repository backup using `fullBackupTo:` and restore from full backup are now multi-threaded. For multi-file backups, the protocol that created or restored backup files in a sequence of commands has been removed; now, all backup files are created or restored simultaneously. Multifile backups are now created or restored by a single command rather than repeated calls to continue* methods.

The methods `fullBackupTo:` and `fullBackupCompressedTo:` have the same interface, although they also use the multithreaded backup code.

To create a multi-file backup, you pass in an array of one or more filenames as the first argument to `fullBackupTo:MBytes:`. The second argument is either a single integer, which will be used as a size limit for each file, or an array with the same number of integer elements as the array of filenames.

Regardless of the `MByte:` limit passed in, the last filename in the array will be considered to have unlimited size, to avoid backups failing for lack of space (other than lack of space for this final file).

Multi-file backups are written in parallel, using separate threads to write to each file after initial material is written to the first file.

Backup and restore now use one or more additional sessions. If the number of logins has reached the maximum allowed by your licence or configured limits, backup will fail. On systems with many extents, up to 16 sessions may be used by backup for the best performance; if fewer sessions are available, performance will be reduced.

The process to restore full backups is also now multi-threaded, and the interface has been simplified, similarly to the backup.

For details, see the updated information on backup and restore in the *System Administration Guide*, chapter 9.

GcLock

The multi-threaded backup and restore must be able to get the GcLock, which is held for the duration of the backup. The backup will wait up to 5 minutes for another session holding the GcLock to release the lock before failing.You should avoid starting or running other multi-threaded operations that require the GcLock prior to starting a backup. You can determine if another session is holding the GcLock by executing `System sessionIdHoldingGcLock`. This method returns 0 if the GcLock is available.

Improvements to tranlog replay

Transaction log replay and recovery was made multi-threaded in earlier releases. In this release, a number of conditions that can cause fork-in-time have been cleaned up and the recovery made more reliable.

You should now use the -R option when starting up a stone on fresh extents with existing tranlogs in the configured tranlog directories. Alternatively, you can move the existing transaction logs to an archive directory. The stone will fail to start on a fresh extent with transaction logs present unless the -R option is used.

Logins during tranlog replay

In a repository that is in recovery mode, sessions login to perform the restores. It is also possible for other sessions to log in and execute code, although sessions cannot commit.

Previously this created some risk, if OOPs provided to the logged-in-session for temporary use were also found in a transaction log being restored. Now, sessions that log in while the stone is in restore mode are allocated OOPs from a much higher range, which are unlikely to be used in restored transactions.

Symbol garbage collection

GemStone/S 64 Bit v3.1 includes garbage collection of unused symbols. If enabled, this occurs automatically in the background and requires no management.

This feature is disabled by default and can be enabled by the new configuration parameter, STN_SYMBOL_GC_ENABLED, or by the runtime equivalent, #StnSymbolGcEnabled.

When enabled, unreferenced symbols are located and put in a possibleDeadSymbols collection during a markForCollection operation. These symbols are hidden, to remove references from AllSymbols but retain the OOPs until the voting, union, and finalization are done. Any lookups on the symbol will return the existing symbol, hidden or not, and restore it to AllSymbols.

Once voting and write-set union sweep have completed, the symbols that are otherwise unreferenced are removed from the possibleDeadSymbols, so they will be collected by the next MFC.

Methods such as `_existingWithAll:` that determine if a symbol exists only check AllSymbols, and will not find symbols in possibleDeadSymbols.

New hot standby functionality

To support a truly hot standby—one that does not have to wait for a final transaction log to be copied to and replayed into the standby system—new functionality has been added to continuously transmit and restore transaction records.

New terminology is used to describe the repositories in a hot standby configuration. The primary system is now referred to also as the master system, while the standby is the slave system.

Two new processes have been added, the logsender and the logreceiver. These are started and stopped using the utilities startlogsender, stoplogsender, startlogreceiver, and stoplogreceiver. They run independently of the stone and do not stop when the stone

shuts down. If the connection is broken or any components are shut down and restarted, these processes will automatically reconnect and resynchronize.

As with the warm standby, the slave system is restored from a backup of the master system.

The master system runs the logsender process, which continuously transmits transaction log records to the slave system. The records are transmitted in compressed form. By connecting to the master stone, the logsender knows when there are new log records to transmit.

The slave system runs the logreceiver process, which receives the log records from the logsender and writes them to a local directory. By connecting to the stone, the logreceiver can make the slave stone aware that there are new transaction log records to restore.

The slave stone runs in "continuous restore" mode. This mode is entered using the method `Repository >> continuousRestoreFromArchiveLogs:` and is terminated by `Repository >> stopContinuousRestore`. While in this mode (provided the slave stone is notified of new records by the logreceiver) the slave stone restores each transaction log record as it is available.

For a more detailed description of the hot standby process and how to set up a hot standby system, refer to the *System Administration Guide* for v3.1, Chapter 10. The details of the utilities that start and stop the logsender and logreceiver processes are also provided in *System Administration Guide* appendix 8, and in man pages.

The environment variables GS_DEBUG_LOGSENDER and GS_DEBUG_LOGRECEIVER, if set, allow more detail to be printed to the logsender and logreceiver logs. This is intended for use when working with GemStone Technical Support to resolve issues.

Support for SSL sockets from Smalltalk

The OpenSSL code that GemStone uses internally to perform secure logins is also available for applications to use to create secure sockets.

GsSecureSocket is a new subclass of GsSocket. This class provides the interface for creating and binding TLS or SSLv3 TCP sockets through the operating system of the machine that is running the session's Gem process, and for communicating across those sockets.

Only SSLv3 and TLSv1 connections are supported. SSLv2 connections are not supported since these connections are known to be insecure.

To create a secure connection, a regular socket connection is first established using methods inherited from GsSocket, and the GsSecureSocket configured with the appropriate certificates. Then further methods are executed to make the connection secure.

See class methods serverExample and clientExample, and serverExample2 and clientExample2, for examples of using GsSecureSocket. Detailed documentation is in the *Programming Guide* for v3.1, chapter 11.

The GemStone distribution includes example certificates that are used by the GsSecureSocket example methods. These are in $GEMSTONE/examples/openssl/certs. The scripts and configuration files used to generate the certificates are also included.

Like instances of GsSocket, instances of GsSecureSocket automatically have their C state closed when the instance is garbage collected or when a persistent instance drops out of local memory. Do not retain an instance of GsSecureSocket from one session to another; GsSecureSockets that are used across sessions always generate an error.

The environment variable GEMSTONE_SOCKET_DEBUG, if set to any value, will cause tracing messages to stdout of bind(), connect(), getaddrinfo() and related socket calls.

Message Digest Functions

Message digest functions in the OpenSSL package are now used to provide message digest functionality. The previous open-source md5 code is no longer used.

The following instance methods have been added to ByteArray and CharacterCollection. For DoubleByteString and QuadByteString, computations are based on viewing the string as a ByteArray holding big-endian characters.

See image method comments for details.

```
md5sum
asMd5String
sha1Sum
asSha1String
sha256Sum
asSha256String
sha512Sum
asSha512String
```

GsSocket added methods to manage blocking

The following convenience methods have been added to GsSocket:

```
GsSocket >> isNonBlocking
GsSocket >> isBlocking
GsSocket >> makeNonBlocking
GsSocket >> makeBlocking
```

Code invoked on session login

It is now possible to specify code that will be executed at the start of each session, after login, for specific UserProfile instances.

This is done by calling the new method `UserProfile >> loginHook:`, passing in a block or Symbol. The argument can be nil, the selector of a unary instance method on UserProfile, or a zero-argument ExecBlock.

If you have previously modified `GsCurrentSession>>initialize` to perform equivalent login initialization, review and merge your changes to that method before upgrading. The initialization specified using `UserProfile >> loginHook:` is invoked from the `GsCurrentSession>>initialize` method.

Note this functionality uses the previously reserved UserProfile instance variable spare3.

Nested in-memory transactions

Within a transaction, GemStone/S 64 Bit now allows you to group units of work into logical transactions, which can be committed or aborted within the given session. These logical transactions can be nested with up to 16 levels of nesting (including the outer level actual transaction). When the full set of changes are ready to be committed, committing the outer transaction will make the changes persistent and detect any conflicts.

While the same protocol is used to commit the actual (outer) transaction and the nested transactions, the semantics are different. A commit of a nested transaction does not detect conflicts with changes by other users, does not update current session state, and does not make the changes persistent if the session exits unexpectedly, nor recoverable on system shutdown. Abort of a nested transaction returns the session to the state it was in at the beginning of the nested transaction, without updating the session's view with any changes by other users.

`System class >> transactionLevel` will return the level of the transaction. 0 means not in transaction, 1 means in transaction, and -1 means the session is in transactionless mode. In addition, this method can return values up to 16 to indicate the number of levels of nested transaction.

To begin a nested transaction, use the new method `beginNestedTransaction`. The method `beginTransaction` is used to start a actual transaction, and cannot be used when in a nested transaction, i.e. the transactionLevel must be 1 or less. These method produce the same results, starting a new transaction, at transaction level 0. These methods can both be used at transaction level 1, but produce different results. When the transaction level is 1, `beginNestedTransaction` will begin a nested transaction and raise the transactionLevel to 2, while `beginTransaction` will abort the current actual transaction and start a new one, leaving the system still in transaction level 1.

To rollback changes in a nested transaction, use `abort` or `abortTransaction`. These methods will return the session to the state it was in at the time the current nested transaction was started, and leave the session at the parent level of transaction. There are no updates from repository changes outside the session, only in-memory state is rolled back.

However, if the current transaction is not a nested transaction, i.e. the current transaction level is 1, then the view is updated to the most recent view of the repository, as usual.

The method `abortAll` aborts all levels of nested transaction as well are performing the actual outer level abort.

To preserve changes in a nested transaction, use `commit` or `commitTransaction`. These methods will preserve changes in the current session state and return the session to the parent level of transaction. The changes are not saved to the repository and no conflict detection is done.

However, if the current transaction is not a nested transaction, i.e. the current transaction level is 1, then normal commit processing is done. If successful, the changes are made persistent, otherwise conflicts are detected.

The method `commitAll` saves the changes from all levels of nested transaction and performs the actual outer level commit.

`continueTransaction` cannot be used when there is more than one level of transaction active.

The following methods are new or have new behavior:

`System class >> commit`
: If it is an outer level commit, commits changes to the repository and returns true for success or signals a TransactionError if conflicts prevent commit. If outer level commit is successful, updates session's view, and leaves session out of transaction or begins a new transaction according to transaction mode. If inner level commit, preserves changes in memory and returns system to parent transaction level.

`System class >> commitTransaction`
: Same as `System class >> commit`, except that on conflict of outer level commit, returns false rather than signalling an error.

`System class >> commitAll`
: Commit all levels of nested transaction, including the outer level commit. Returns true for success or signals a TransactionError if conflicts prevent commit.

`System class >> abortAll`
: Abort all levels of nested transaction, including the outer level abortTransaction that updates the session's view.

`System class >> abort`
: If it is an outer level abort, discards any changes and updates the session's view. If it is an inner level abort, returns the system to the state it was at the beginning of the nested transaction and returns system to the parent transaction level.

`System class >> abortTransaction`
: Same as `System class >> abort`.

`System class >> beginNestedTransaction`
: Begin a nested transaction, when already in transaction. A maximum of 16 levels, including the outer level, are supported. If outside of transaction (transaction level 0), updates the session to the current view and begins a new transaction.

`System class >> transactionLevel`
: Return the current level of transaction. -1 is transactionless, 0 is not in transaction, 1 is in transaction, and 2-16 are nested transaction.

Support for UTF and Locale-specific collation using ICU

GemStone/S 64 Bit v3.1 includes an interface to the International Components for Unicode (ICU) libraries. ICU is a widely used open source set of libraries providing Unicode and globalization support for software applications. ICU available on many platforms, and provides consistent results on all platforms and software environments.

Previous versions of GemStone/S 64 Bit provided string collation using the CharacterDataTables feature, which loaded Unicode collation tables. This allowed control over absolute ordering of Characters, but did not allow sorting that was not case or diacritical-mark sensitive. This CharacterDataTable collation functionality is deprecated.

GemStone/S 64 Bit v3.1 includes ICU v4.8.1.1 libraries. These are provided as shared libraries that are loaded automatically by gem code when required.

For more details on ICU collation, see the updated *Programming Guide*, chapter 5.

IcuLocale

Instances of this class represent a specific language, country, and language variant. The available IcuLocales are in the shared library and can be listed using #availableLocales.

A default instance of IcuLocale is instantiated on first reference, and stored in session state. The default IcuLocale is based on the operating system locale setting for the Gem.

To set a specific default IcuLocale, use the method `IcuLocale class > default:`. This sets the default locale for the session executing this code. While the instance of IcuLocale can be made persistent, the default IcuLocale does not persist from session to session.

New Unicode String classes

The following classes have been added to allow locale-specific comparisons:

Unicode7
: A subclass of String, limited to holding Characters with codepoints in the range 0..127 that are represented in 7 bits.

Unicode16
: A subclass of DoubleByteString, holding Characters with codepoints in the range 0..16rFFFF (65K), excluding the range D800-DFFF. This range is reserved for surrogates as per the UTF-16 standard.

Unicode32
: A subclass of QuadByteString, holding Characters with codepoints in the range 0..16r10FFFF. GemStone Unicode strings are limited to Characters with codepoints less than or equal to 16r10FFFF.

The following class has been added to hold UTF-8 encoded data:

Utf8
: A subclass of ByteArray, holding strings encoded as UTF-8. Instance of Uft8 are created using `encodeIntoUTF8` and should not be manipulated at the byte level.

These classes are the equivalent of their superclasses, with the given difference in range for the Unicode String classes, and with additional comparison behavior supplied by the ICU libraries.

Sending the message #asUnicodeString to an instance of String, DoubleByteString, QuadByteString, ByteArray, or Utf8 will create an instance of the appropriate Unicode String class, depending on the smallest representation that can hold the given data.

Instances of Unicode String classes and Utf8 will compare as equal to instances of their superclass containing the equivalent characters.

Instances of the Unicode Strings classes and Utf8 currently cannot be replicated to GBS.

IcuCollator

An IcuCollator encapsulates the rules involved in collation for a specific IcuLocale. IcuCollators can be configured using a number of attributes, which allow (among other features) control over case and can enable sort to disregard punctuation and whitespace. These sorting attributes are in addition to the language- and region-specific sorting rules based on the IcuLocale.

When instances of the Unicode String classes are compared, the ordering is always determined based on the specific IcuCollator and IcuLocale, using the method `compareTo:collator:`. If no IcuCollator is specified, such as with >, the default IcuCollator is used.

The comparison itself is done by the ICU libraries, and follows the ICU comparison rules for the specific IcuLocale and configured IcuCollator.

A default instance of IcuCollator is instantiated on first reference based on the default IcuLocale, and stored in session state.

When comparing an instance of a string class with an instance of a Unicode string class, the comparison is based on the receiver. As a result, sorting of collections that mix string and Unicode string classes may produce inconsistent ordering.

IcuSortedCollection

IcuSortedCollection is a subclass of SortedCollection. Instances of IcuSortedCollection reference an instance of IcuCollator (which in turn references an IcuLocale), and sort according to the specific rules of the locale and any customizations in the IcuCollator. IcuSortedCollections are restricted to containing instances of Unicode7, Unicode16, Unicode32, and Utf8. IcuSortedCollections should not have a sortBlock set explicitly; sorts are always done by sending `compareTo:collator:` to the elements.

GsFile reading and writing

To read UTF8 data from a file, a new method `GsFile >> contentsAsUtf8` has been added. This returns an instance of Uft8 (subclass of ByteArray) that holds the UTF-8 encoded contents of the entire file. The method `Utf8 >> decodeFromUTF8` can then be used to decode the contents into an instance of the appropriate Unicode String class.

Changes in encode/decode UTF8 and decodeIntoDoubleByteStringFromUTF8

The methods `encodeAsUTF8` and `decodeFromUTF8` have new behavior.

`encodeAsUTF8` now returns an instance of Utf8 containing the UTF-8 encoded contents of the receiver.

`decodeFromUTF8` now returns an instance of Unicode7, Unicode16, or Unicode32; the smallest required to hold the decoded contents.

The method `decodeIntoDoubleByteStringFromUTF8` has been removed. This was a convenience method to avoid creating a QuadByteString when s DoubleByteString was sufficient to hold the data. The new behavior of decodeFromUTF8 makes this method unnecessary.

Hidden Sets public API

Hidden sets are structures that were originally intended for internal use, but have proved useful for certain purposes. Customer-usable hidden sets have been available for some time.

For version 3.1, the full API for accessing hidden sets, including reading system hidden sets as well as modifying customer hidden sets, has been made public. The following methods have been added:

```
System Class >> add: anObject toGciSet: hiddenSetSpecifier
System Class >> add: anObject toHiddenSet: hiddenSetSpecifier
System Class >> addAll: anArray toGciSet: hiddenSetSpecifier
System Class >> addAll: anArray toHiddenSet: hiddenSetSpecifier
System Class >> gciDirtyObjsInit
System Class >> gciTrackedObjsInit
System Class >> getAndClearGciDirtySet: hiddenSetSpecifier
   into: anArray
System Class >> hiddenSetAsArray: hiddenSetSpecifier
System Class >> hiddenSetEnumerate: hiddenSetSpecifier
   limit: maxResultSize
System Class >> hiddenSetEnumerateAsInts: hiddenSetSpecifier
   limit: maxResultSize
System Class >> hiddenSetReinit: hiddenSetSpecifier
System Class >> hiddenSetSize: hiddenSetSpecifier
System Class >> remove: anObject fromGciSet: hiddenSetSpecifier
System Class >> remove: anObject fromHiddenSet: hiddenSetSpecifier
System Class >> removeAll: anArray fromGciSet: hiddenSetSpecifier
System Class >> removeAll: anArray fromHiddenSet: hiddenSetSpecifier
System Class >> testIf: anObject isInHiddenSet: hiddenSetSpecifier
```

These are in addition to the previously-existing public hidden set methods. The private methods, in most cases the same name with a leading underscore, are also still available, but we recommend updating your code to use the public API.

For more information on these methods, refer to the method comments in the image.

Adding application-specific information to an instance of GsProcess

GemStone's GsProcess class includes an instance variable clientData, which was originally intended to allow data to be attached to a GsProcess. This variable is also used by the Seaside product.

To allow legacy use of clientData, the methods clientData and clientData: have been added.

So that applications can safely attach data to GsProcess instances, without concern for conflict with Seaside, the dictionary at the #environment instance variable is now also public for use to store GsProcess specific data. This can be accessed and updated using the following methods:

```
GsProcess >> environmentAt:
GsProcess >> environmentAt:put:
GsProcess >> environmentAt:ifAbsent:
```

Additional protocol to enable/disable signalling

Before a session can receive certain signals, they must be enabled, and if a signal is received, receipt must be re-enabled before another signal can be received. The following Notifications have new methods for enabling, disabling, and determining the status. The existing System class protocol remains available.

AlmostOutOfMemory

AlmostOutOfMemory notifications are sent, if enabled, when a gem session is low on memory. The following methods enable and disabled and manage the threshold at which signals are sent:

```
AlmostOutOfMemory class >> disable
AlmostOutOfMemory class >> enable
AlmostOutOfMemory class >> enabled
AlmostOutOfMemory class >> enableAtThreshold: anInteger
AlmostOutOfMemory class >> threshold
AlmostOutOfMemory class >> threshold: anInteger
```

These added methods are the equivalent of existing methods on System class: `enableAlmostOutOfMemoryError`, `disableAlmostOutOfMemoryError`, `signalAlmostOutOfMemoryThreshold:`, `almostOutOfMemoryErrorThreshold`.

TransactionBacklog

The TransactionBacklog notification is sent, if enabled, when a session there is a commit record backlog. It is sent to both to session in transaction and not in transaction.

```
TransactionBacklog class >> disableSignalling
TransactionBacklog class >> enableSignalling
TransactionBacklog class >> signallingEnabled
```

The TransactionBacklog notification is the equivalent to two legacy signals; sigAbort for sessions not in transaction, and finishTransaction for sessions that are in transaction. These new methods enable or disable handling of both.

To manage just the sigAbort variant sent to sessions not in transaction, use the methods on System class: `enableSignaledAbortError`, `disableSignaledAbortError`, and `signaledAbortErrorStatus`.

To manage just the finishTransaction variant sent to sessions in transaction, use the methods on System class: `enableSignaledFinishTransactionError`, `disableSignaledFinishTransactionError`, and `signaledFinishTransactionErrorStatus`.

ObjectsCommittedNotification

The ObjectsCommittedNotification is sent when objects in a session's notify set have changed.

```
ObjectsCommittedNotification class >> disableSignalling
ObjectsCommittedNotification class >> enableSignalling
ObjectsCommittedNotification class >> signallingEnabled
```

These added methods are the equivalent of the existing methods on System Class: `enableSignaledObjectsError`, `disableSignaledObjectsError`, and `signaledObjectsErrorStatus`.

InterSessionSignal

The InterSessionSignal is sent when a session sends a signal to another session.

```
InterSessionSignal class >> disableSignalling
InterSessionSignal class >> enableSignalling
InterSessionSignal class >> signallingEnabled
```

These added methods are the equivalent of the existing methods on System Class: `enableSignaledGemStoneSessionError`, `disableSignaledGemStoneSessionError`, and `signaledGemStoneSessionErrorStatus`.

There are further class methods added on InterSessionSignal:

`InterSessionSignal class >> poll`
Return a new instance of InterSessionSignal if a signal is available from another session. If no signal is available, return nil.

`InterSessionSignal class >> pollAndSignal`
Signal a new instance of InterSessionSignal if a signal is available from another session, otherwise do nothing.

`InterSessionSignal class >> sendSignal:` *aSmallInt* `to:` *aSessionId* `withMessage:` *aString*
Send a signal to another session, which can result in an InterSessionSignal being signalled within that session.

The method `InterSessionSignal >> sendingSession` is deprecated.

Garbage Collection Enhancements

Ability to defer reclaim under low free space conditions

Immediately following markForCollection, there can be a heavy load of reclaim activity. On fast systems, the reclaim gems may use pages faster than the freed up pages can be made available again, resulting in sharp drops in free space.

To avoid reclaim activity causing low free space problems, the following GcGem configuration variable (in GcUser's UserGlobals) has been added:

#deferReclaimFreeSpaceThreshold
Specifies a threshold number of freePages in the repository below which page reclaiming is deferred. While waiting in this mode, reclaim gems run outside of a transaction and respond to SigAborts from stone. Setting the value to 0 disables this feature. Default: 0.

reclaimAll now performs postReclaimAll

Previously, following a reclaimAll operations, you were instructed to invoke postReclaimAll: with the results of the reclaimAll. Now, this is handled automatically.

Further optimizations to MFC

In addition to the improvements to markForCollection to run multi-threaded in previous releases, further optimizations have been done in version 3.1.

New markForCollection variant with buffer size

The method `Repository >> fastMarkForCollectionWithPageBufSize:` has been added. This is the same as `fastMarkForCollection`, but allows you to specify the buffer size.

RcIdentityBag >> _unsafeCleanupBag added

The method is a variant of `RcIdentityBag >> cleanupBag`. Unlike `cleanupBag`, this method handles the removalBags for all sessions, not just inactive ones. This method will cause concurrency conflicts and should only be used while no sessions are adding to or removing from the RcIdentityBag.

descriptionOfSession: now includes gem type, temporary OOPs and pages

The information returned by `System class >> descriptionOfSession:` now includes additional fields. See the method comments in the image for the complete list of values returned by this method.

17. Gem type. This is nil for regular user sessions, or a descriptive string for system gems.
18. Number of temporary (uncommitted) object IDs allocated to the session.
19. Number of temporary (non-persistent) page IDs allocated to the session.

GciInterface changes

In addition to the changes listed, a serious bug in GciInterface has been fixed in this release; see "GciInterface >> remotePerform:selector:args: failed with OOPs of non-specials" on page 42.

New login method nbLogin:flags:

A new method, `GciInterface >> nbLogin:flags:` allows you to pass in flags to the login. The flags are specified as bits in a SmallInteger argument; the description and specific bits are listed in `$GEMSTONE/include/gci.ht`.

The flags GCI_LOGIN_FULL_COMPRESSION_ENABLED, GCI_LOGIN_ERRS_USE_REF_SET, GCI_LOGIN_QUIET, and GCI_CLIENT_DOES_SESSION_INIT can be optionally specified; GCI_LOGIN_PW_ENCRYPTED and GCI_LOGIN_IS_SUBORDINATE are always used for login from the GciInterface.

Distinguishing OOP from Integers in results

To avoid ambiguity in the return value of GciInterface operations that return objects, the method `GciInterface >> lastResultAsOop` has been added. This will always return an OOP.

Change in return class for Set and IdentitySet

Previously, `Set>>collect:` returned an instance of Bag, and `IdentitySet>>collect:` returned an instance of IdentityBag. However, the ANI specification requires that the results not include duplicates. Now, `Set>>collect:` will return an instance of Set, and `IdentitySet>>collect:` will return an instance of IdentitySet. (#41894)

Privilege Changes

New privilege to allow non-SystemUser to compile primitive methods

A new privilege has been added, CompilePrimitives. UserProfiles that are granted this privilege may compile methods containing primitives; previously, only SystemUser could compile primitive methods. (#41804)

This is intended to avoid login as SystemUser for the rare cases where a modification to a kernel primitive method is required.

FedAdminPrivilege renamed

The privilege FedAdminPrivilege has been renamed PrivUnused5, since the operations for which this privilege was intended are not available in GemStone/S 64 Bit.

OtherPassword now required to modify another users's defaultObjectSecurityPolicy

Previously this privilege was not required.

GsObjectSecurityPolicy new now allowed

Previously, creating a GsObjectSecurityPolicy (Segment) requiring using the method `newInRepository:` and passing in the argument SystemRepository. Now, invoking the `new` method is permitted and will perform the same action.

Repository>>name: method removed

This method was a historical artifact that is not meaningful, since the Repository class has a singleton instance "SystemRepository".

Direct IO for tranlogs now only available on Solaris

Since tranlog writes may be smaller than appropriate for direct I/O, they cause alignment problems on most operating systems. As a result, STN_LOG_IO_FLAGS can now only be used on Solaris.

OS level System stats for processes now available from Smalltalk

The following methods have been added to allow access to process level host statistics from Smalltalk. Note that some stats may be zero if the gem and the process sampled do not have the same effective user ID.

`System class >> hostStatisticsForProcess:` *anInteger*
Returns an array of SmallIntegers which represent the host statistics for the given process ID. The names of each statistic are returned by the #hostProcessStatisticsNames method.

`System class >> hostStatisticsForMyProcess`
Returns an array of SmallIntegers which represent the host statistics for this process. The names of each statistic are returned by the #hostProcessStatisticsNames method.

`System class >> hostProcessStatisticsNames`
Returns an array of Strings which are the names of the per-process statistics provided by this host.

Improved error reporting for remote cache loss

Previously, any problem with remote caches was reported as a timeout. Now, the specifics of the problem are reported, including what operation was in progress and for timeouts, how long the page manager waited for a response from the remote cache.

dictionaryAndSymbolOf: deprecated

The methods `UserProfile >> dictionaryAndSymbolOf:` and `SymbolList >> dictionaryAndSymbolOf:` are deprecated.

These methods are replaced by `dictionariesAndSymbolsOf:`, since an object may exist in multiple places.

Changes in messages and log file output

A number of changes in the details of progress messages and log file output have been made in this release.

Messages in the stone log regarding growth are reduced to every 100MB of growth.

Disallow block device raw partitions

GemStone has generally discouraged use of block devices for raw partitions. In this release, the use of block partitions has been disallowed for raw partitions for extents and transaction logs.

Improvement to communication between Page Manager and remote caches.

The TCP socket send and receive buffers that are used between the PageManager and remote caches have been resized for more efficient handling of data transfers. (#41927)

true, false, and nil now allowed as message selectors

The specials true, false, and nil are now allowed to be used as message selectors.

Limitation on native code on Mac

On sessions running with very large temporary object caches, on the Macintosh it is possible that the memory offset will exceed 32 bits. In this case, native code is disabled for that session.

FFI changes

The FFI has a number of changes and improvements, in particular to the header parsing code. A much wider range of C header files can now be parsed. In addition, the documentation has been updated and expanded; see the *Programming Guide*.

The method `CCallout class >> errno` has been added, to allow retrieval of the C level error from a C call.

Removed methods

The following methods have been removed, since with support for NotTranloggedGlobals, explicit manipulation of dirty sets is no longer allowed.

```
System class>>  rollbackDirtyList:
System class>>  setDirtyList:
```

cachewarmer will run in recovery/restore

Cachewarming, either using the configuration options or by executing the utility script startcachewarmer, will now run when the stone is recovering after unexpected shutdown or when in restore mode.

Distribution Changes

There are a number of changes in the files and directories in the installed distribution. These are documented in the relevant place in these release notes. To summarize some of the changes:

- Most platforms now include the bin32 directory, providing a 32-bit topaz executable to assist in debugging SSL-related login issues. bin32 and lib32 are not provided on AIX, which does not support 32-bit GCI client applications.
- The set of shared libraries has changed; additional shared libraries are required. The naming convention has changed. Symbolic links are no longer used to provide a short, general name; the entire shared library name is now used by processes that load libraries.
- Additional utility scripts are provided to support hot standby.
- Helpful tools for working with SSL sockets is provided under examples/openssl.
- Webtools, in preparation for being made open-source, has been removed from the distribution.

Cache Statistics Changes

The following statistics have been added in this release:

CompressedLogPagesWrittenByGem (Shrpcmon)
The number of compressed log pages in the cache because of a write done by a Gem.

CompressedLogPagesWrittenByStone (Shrpcmon)
The number of compressed log pages in the cache because of a write done by the Stone.

ContinueTransactionCount (Gem)
Total number of times the session executed the "System continueTransaction" method.

MaxUserSessions (Stone)
The maximum number of user sessions stone is configured for.

PossibleDeadSymbols (Stone)
The number of symbols found to be not referenced in a markForCollection.

ReadLocksSize (Gem)
The number of objects that are read locked.

ReclaimedSymbols (Stone)
The number of symbols that have been reclaimed since the stone was started.

WriteLocksSize (Gem)
The number of objects that are write locked.

Configuration Parameter Changes

Configuration parameters queries from image now return consistent true/false

Setting and fetching configuration parameters values using the image methods configurationAt: and configurationAt:put:, for settings that are boolean values, was not previously consistent. Some settings accepted and reported true/false, others 1/0.

Now, all queries for boolean configuration parameters return true or false. Both true/false and 1/0 may be used to update these values.

This has no effect on the configuration files, only on the interface from Smalltalk.

STN_TRAN_LOG_DIRECTORIES and STN_TRAN_LOG_SIZES, requirements change and related image changes

Previously, STN_TRAN_LOG_DIRECTORIES had a minium size of two. For smaller repositories that kept all transaction logs in one file system directory, this meant a single directory name wa listed twice. To simply this, it is now permitted to specify only one directory for all transaction logs.

The related configuration parameters STN_TRAN_LOG_SIZES also has been changed to allow a single value. As previously, the number of values mst equal that in STN_TRAN_LOG_DIRECTORIES.

A single entry on a raw partition will causes errors when the raw partition becomes full. In partial tranlog mode, while usable, log creation deletes the existing log, resulting in periods in which the ability to recover from unexpected shutdown is lost.

The generic way to fetch configuration parameters, using `System class >> stoneConfigurationAt:`, may now return either an Array or a String, depending on if the configuration is set to a list or a single value. To simplify access to this information, the following methods have been added.

`Repository >> allTranlogDirectories`
: Return an array of one or more Strings, each of which represents 1 element of the STN_TRAN_LOG_DIRECTORIES configuration parameter.

`Repository >> allTranlogSizes`
: Return an array of one or more SmallIntegers, each of which represents 1 element of the STN_TRAN_LOG_SIZES configuration parameter.

Default added for STN_TRAN_FULL_LOGGING

Previous, STN_TRAN_FULL_LOGGING did not have a default, and the example system.conf set this to false.

Now, the default for STN_TRAN_FULL_LOGGING is true, and the example in system.conf has been also been changed to true.

Changes in default for STN_FREE_SPACE_THRESHOLD

The default for STN_FREE_SPACE_THRESHOLD has been 1M, which is much too small for modern disk and repository sizes.

The default is now 0, which means that the stone will compute a free space threshold that is 1/1000 of the current size of the repository and at least 5M. This is adjusted as the repository grows.

You should examine your current configuration settings to confirm that if you have explicitly set STN_FREE_SPACE_THRESHOLD, that this value is sufficiently large to allow the system enough free space to handle repository full situations as gracefully as possible.

Changes in STN_LOG_IO_FLAGS

STN_LOG_IO_FLAGS now is limited to 0 (default) or 1 (Direct I/O). A setting of 1 (Direct I/O) can only be used on Solaris.

Changes in default for SHR_NUM_FREE_FRAME_SERVERS

Previously, the default number of Free Frame Page servers was 1. Using this default, if the number of AIO page servers was increased, could cause performance bottlenecks for the AIO page servers (see "Slow checkpoints with multiple AIO pgsvrs and fewer Free Frame pgsvrs" on page 45).

Now, the default setting for this configuration parameters is -1, which specifies to use the value for STN_NUM_LOCAL_AIO_SERVERS on the primary cache, or 1 on a remote cache, as the value for SHR_NUM_FREE_FRAME_SERVERS.

You should examine your current configuration settings to check if you have explicitly set SHR_NUM_FREE_FRAME_SERVERS.

Also, the maximum for this has been raised from 30 to 255.

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 2000000 (2GB).

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 on AIX. 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_HALT_ON_ERROR now has runtime equivalent

GEM_HALT_ON_ERROR is used to generate C level stacks when the specified GemStone error occurs. Now this can be configured for a specific gem while the gem is running, rather than requiring it to be set in the configuration file prior to gem startup.

To configure, use the runtime parameter name #GemHaltOnError.

Added configuration parameters

GEM_RPC_USE_SSL

Controls whether a remote RPC gem uses a secure socket layer (SSL) connection to converse with its RPC client. RPC sessions always establish a secure connection during the login sequence. This parameter controls whether the gem and its remote RPC client continue using the SSL connection after login is complete. If this setting is true, after login is complete, a standard TCP/IP socket connection is used for communication.

This option has no effect for linked gems and local RPC gems (i.e., a gem running on the same host its client). Local RPC gems always revert to a standard TCP/IP socket after login.

Secure sockets are slightly slower than insecure sockets due to the overhead of encrypting and decrypting data.

Default: TRUE

GEM_TEMPOBJ_START_ADDR

Used on AIX only, if default mmap of temp obj memory fails. The value is used to define the starting address at which to attempt to mmap temp obj memory using MAP_FIXED at fixed addresses and munmap to simulate MAP_NORESERVE.

A config file value of zero results in an internal default of 0xA00000000000000 for AIX 7, and 0x700000000000000 for AIX 6. A non-default value must be coded as an exact address and may be affected by use of mmap by other shared libraries.

Default: 0

SHR_PAGE_CACHE_PERMISSIONS

SHR_PAGE_CACHE_PERMISSIONS specifies the UNIX permission settings of the shared page cache, expressed as an octal number. The first two digits are constant and must always be 06. The 0 indicates an octal constant and 6 indicates the UNIX user which created the cache has read/write permissions.

The last two digits specifies the group and other permissions respectively. Each of the last two digits must be one of the following:

6 - read/write
4 - read only
0 - no access

By default, the shared page cache is created with group read/write permission but no access for other users.

Default: 0660

STN_LISTENING_ADDRESSES

A list of 0 to 10 addresses upon which stone should listen for login connections. If list is empty, the default address "::" is used, which means listen on any active network interfaces, plus the loopback ("::1") interface.

Each element of the list may be a name or a numeric IPv6 address. Each named address must resolve via getaddrinfo() to at least one address legal to listen on, i.e. resolve to the

loopback or wildcard address, or to an address assigned to a network interface on this machine. Each numeric address must be an address legal to listen on.

Numeric IPv6 addresses may be any form recognized by inet_pton(), including AF_INET6, address, etc., or by inet_pton(), including AF_INET, address, etc., on the host operating system. Per RFC 2373 this includes these forms:

- IPv4 dotted-decimal format, d.d.d.d
- IPv6 hex format x:x:x:x:x:x:x:x where x is a 16 bit hexadecimal number
- IPv4-mapped IPv6 ::FFFF:d.d.d.d where d is an 8 bit decimal number

IPv6 format may contain at most one :: which is a contiguous group of zeros. The loopback address 0:0:0:0:0:0:0:1 can be written as ::1. The wildcard address 0:0:0:0:0:0:0:0 can be written as ::.

If the list contains the wildcard address ::, the other elements of the list are ignored.

If the list does not contain ::, then the loopback addresses ::1 and 127.0.0.1 are always listened on even if not explicitly in the list, to support logins from system gems.

See public documents RFC 4291 and RFC 4038 for more information on IPV6 addressing.

Default: An empty list, "::"

STN_SYMBOL_GC_ENABLED

Determines if symbol garbage collection is allowed to run on the system. Setting this value to true enables symbol garbage collection.

Updating the runtime parameter requires the GarbageCollection privilege.

Runtime equivalent: #StnSymbolGcEnabled
Default: FALSE

Changes in Utilities

New utility commands

The following utility commands have been added as part of the new hot standby feature:

startlogsender
stoplogsender
startlogreceiver
stoplogreceiver

For more information, see the description of the hotstandby under "New hot standby functionality" on page 17, the updated information in the *System Administration Guide*, or the man pages.

copydbf changes

copydbf now allowed on extents that are in use

Historically, copydbf was disallowed on extents that are in use, to avoid copying live extents. With online extent snapshot backups, however, it is legal to copy live extents (provided checkpoints are disabled). Copydbf can now be used to make extent copies of running repositories. Note that if checkpoints are not disabled for the copy, the resulting extents will be unusable.

copydbf -i/-I output omits destination

Previously, the copydbf -i or -I output included the destination, which was /dev/null when using the information option. These lines are no longer included in the output.

copydbf -i/-I includes backup version

Backup files now include information regarding the GemStone version that created them. Copydbf now reports this information. When using copydbf -i or -I to get information about a backup file from version 3.1 or later, the results include the version of GemStone/S 64 Bit that created the backup.

The additional output of copydbf includes a line similar to:

```
Backup was created by GemStone Version: 3.1
```

Statmonitor new option to collect samples for only stone and spcmon

The added statmonitor option -J specified to only collect stats for the stone, shared page cache monitor, and page manager only.

Topaz Changes

Terminology clarification

Previously, GemStone documentation and topaz help text used the terms "context", "activation", and "frame" to refer to the same thing. The term used now going forward is "frame", although there may still be references to the other terms.

Saved stacks

Previously, you could save up to 8 stacks. This limit has been increased to 500.

STK deprecated

The command STK is deprecated in this release; the command WHERE provides the equivalent behavior.

Improved handling of pasting large texts

Topaz's paste buffer has been increased and paste handling has been improved, so that a long single line of text that is pasted into topaz is handled correctly. This is specific to long texts that do not include line breaks.

Note that Solaris has a known bug that may cause the terminal to hang when long lines without line breaks are pasted. This OS bug is exposed in version 3.0 and later; in v2.x, pastes of lines with more then 256 characters were not possible. On Solaris, in v3.0.x and v3.1, avoid pasting text that contains long, unbroken text into topaz. This problem does not occur when text is input from file.

Handling of errors in .topazini

Previously, errors in topazini were silently suppressed. Now, if errors occur while executing .topazini, the .topazini input is echoed and the error is reported.

Note that the echo of topazini input is new; if the topazini includes password information this will be visible if an error occurs.

GCI changes

Added GciNbEnd_

The new function GciNbEnd_ is a version of GciNbEnd with a different result argument type, to allow issues with big and little endian differences to be handled.

With GciNbEnd, the caller must know whether the result argument is a 4 byte or 8 byte value, and dereference the result pointer accordingly on big endian machines.

With GciNbEnd_, the result is 8 bytes and is correct on big endian machines.

Compile and Link Information

Updated compile and link examples are listed below for each platform. Not all commands and platforms have changes; complete current information is listed here for convenience.

Linux

Compiler version

g++ (GCC) 4.1.2 20070115 (prerelease) (SUSE Linux)

Debugger version

GNU gdb 6.6

Compiling a user action or GCI application

```
/usr/bin/g++ -fmessage-length=0 -fcheck-new -O3 -m64 -pipe
-D_REENTRANT -D_GNU_SOURCE -pthread -fPIC -fno-strict-aliasing
-IGEMSTONE/include -x c++ -c userCode.c -o userCode.o
```

Linking user actions into shared libraries

```
/usr/bin/g++ -shared -Wl,-Bdynamic,-hliburAct.so userCode.o
$GEMSTONE/lib/gciualib.o -o liburAct.so -m64 -lpthread -lcrypt
-ldl -lc -lm -lrt -Wl,-z,muldefs -Wl,--warn-unresolved-symbols
```

Linking GCI applications

```
/usr/bin/g++ userCode.o $GEMSTONE/lib/gcirtlobj.o
-Wl,--warn-unresolved-symbols -m64 -lpthread -lcrypt -ldl -lc -lm
-lrt -Wl,-z,muldefs -ouserAppl
```

Solaris on Sparc

Compiler version

CC: Sun C++ 5.8 Patch 121017-05 2006/08/30

Debugger version

Sun Dbx Debugger 7.5 Patch 121023-02 2006/05/26

Compiling a user action or GCI application

```
CC -xO4 -xcode=pic32 -xarch=v9 -mt -xchip=ultra2 -D_REENTRANT
-D_POSIX_PTHREAD_SEMANTICS -I$GEMSTONE/include -c userCode.c
-o userCode.o
```

Linking user actions into shared libraries

```
CC -xarch=v9 -G -Bsymbolic -h libuserAct.so -i userCode.o
$GEMSTONE/lib/gciualib.o -o libuserAct.so -Bdynamic -lc -lpthread
-ldl -lrt -lsocket -lnsl -lm -lCrun -z nodefs
```

Linking GCI applications

```
CC -xildoff -xarch=v9 -i userCode.o $GEMSTONE/lib/gcirtlobj.o
-z nodefs -Bdynamic -lc -lpthread -ldl -lrt -lsocket -lnsl -lm
-lCrun -o userAppl
```

Solaris on x86

Compiler version

CC: Sun C++ 5.10 SunOS_i386 128229-09 2010/06/24

Debugger version

Sun DBX Debugger 7.7 SunOS_i386 2009/06/03

Compiling a user action or GCI application

```
CC -xO4 -m64 -xarch=generic -Kpic -mt -D_REENTRANT
-D_POSIX_PTHREAD_SEMANTICS -I$GEMSTONE/include -c userCode.c
-o userCode.o
```

Linking user actions into shared libraries

```
CC -m64 -xarch=generic -G -Bsymbolic -h libuserAct.so -i userCode.o
$GEMSTONE/lib/gciualib.o -o libuserAct.so -Bdynamic -lc -lpthread
-ldl -lrt -lsocket -lnsl -lm -lCrun -z nodefs
```

Linking GCI applications

```
CC -xildoff -m64 -xarch=generic -i userCode.o
$GEMSTONE/lib/gcirtlobj.o -z nodefs -Bdynamic -lc -lpthread -ldl
-lrt -lsocket -lnsl -lm -lCrun -o userAppl
```

AIX

Note that the supported versions have changed in this release; support for 7.1 is added and AIX 5.3 is no longer supported.

Compiler version

IBM XL C/C++ for AIX, V11.1 (5724-X13)
Version: 11.01.0000.0004

Debugger version

dbx

Compiling a user action or GCI application

```
/usr/vacpp/bin/xlC_r -O3 -qstrict -qalias=noansi -q64 -+
-D_REENTRANT -D_THREAD_SAFE -qpic -qthreaded -qarch=pwr5
-qtune=balanced -D_LARGEFILE64_SOURCE -DFLG_AIX_VERSION=61
-qminimaltoc -qmaxmem=-1
-qsuppress=1500-010:1500-029:1540-1103:1540-2907:1540-0804:1540-
1281:1540-1090 -I$GEMSTONE/include -c userCode.c -o userCode.o
```

Note that there is no space in the -qsuppress arguments that are continued on the following line.

Linking user actions into shared libraries

```
/usr/vacpp/bin/xlC_r -G -q64 userCode.o $GEMSTONE/lib/gciualib.o
-o libuserAct.so -e GciUserActionLibraryMain -L/usr/vacpp/lib
-lpthreads -lc_r -lC_r -lm -ldl -lbsd -Wl,-berok
```

Linking GCI applications

```
/usr/vacpp/bin/xlC_r  -Wl,-bdatapsize:64K -q64 userCode.o
$GEMSTONE/lib/gcirtlobj.o -Wl,-berok -L/usr/vacpp/lib -lpthreads
-lc_r -lC_r -lm -ldl -lbsd -Wl,-brtllib -q64 -o userAppl
```

Darwin

Compiler version

g++: i686-apple-darwin10-g++-4.2.1 (GCC) 4.2.1 (Apple Inc. build 5664)

Debugger version

GNU gdb 6.3.50-20050815 (Apple version gdb-1469) (Wed May 5 04:36:56 UTC 2010)

Compiling a user action or GCI application

```
/usr/bin/g++ -DOBJ_gct -fmessage-length=0 -fcheck-new -O3
-DFLG_FAST=1 -m64 -pipe -D_XOPEN_SOURCE -D_REENTRANT -D_GNU_SOURCE
-fPIC -fno-strict-aliasing -D_LARGEFILE64_SOURCE -pipe -fPIC
-I$GEMSTONE/include -x c++ -c userCode.c -o userCode.o
```

Linking user actions into shared libraries

```
/usr/bin/g++ -dynamiclib userCode.o $GEMSTONE/lib/gciualib.o
-o libuserAct.dylib -m64 -lpthread -ldl -lc -lm -undefined
dynamic_lookup
```

Linking GCI applications

```
/usr/bin/g++ userCode.o $GEMSTONE/lib/gcirtlobj.o -undefined
dynamic_lookup -m64 -lpthread -ldl -lc -lm -o userAppl
```

Windows

Compiler/Debugger version

Microsoft Visual Studio 2008 Version 9.0.21022.8 RTM

Microsoft Visual C++ 2008 91605-270-6514982-60495

Compiling GCI applications

```
cl /W3 /MD /Zi /TP /nologo /DWIN32 /D_CONSOLE /D_DLL /DNATIVE
/I 'VisualStudioInstallPath\VC\atlmfc\include'
/I 'VisualStudioInstallPath\VC\include'
/I 'C:\Program Files\Microsoft SDKs\Windows\v6.0A\Include'
/I%GEMSTONE%\include -c userCode.c -FouserCode.obj
```

Linking GCI applications

```
link /LIBPATH:'VisualStudioInstallPath\VC\lib'
/LIBPATH:'VisualStudioInstallPath\VC\atlmfc\lib'
/LIBPATH:'C:\Program Files\Microsoft SDKs\Windows\v6.0A\Lib'
-INCREMENTAL:NO -nologo 'userCode.obj' '%GEMSTONE%\lib\gcirpc.lib'
wsock32.lib netapi32.lib advapi32.lib comdlg32.lib user32.lib
gdi32.lib kernel32.lib winspool.lib -out:userAppl.exe
```

Bugs Fixed

Support for AIX on POWER7

The POWER7 chip provides more aggressive prefetch of data and instructions. This required additional memory barriers on atomic updates within GemStone, to avoid the risk of coherency and audit problems.

Previous versions of GemStone/S 64 Bit were not certified on POWER7. These earlier versions could have encountered cache coherency and corruption problems when running with production loads on POWER7 systems. (#42089)

GciInterface >> remotePerform:selector:args: failed with OOPs of non-specials

The methods `GciInterface >> remotePerform:selector:args:` and `nbRemotePerform:select:args:` both failed when one or more of the arguments were the OOP of a non-special object. (#42002)

SIGTERM can cause fatal stuck spin lock

If a process is in the process of trying to get a queue lock for a spin lock, and gets a SIGTERM, and the spin lock is a non-recoverable type, the SPC monitor shut down, crashing the system. Now, the process will not immediately exit on SIGTERM, but complete the (short) wait for the queue lock. (#42224)

Failure in pregrow of dynamically added extent causes extent corruption

When an extent is added dynamically using methods such as `createExtent:withMaxSize:`, the new extent is pregrown if DBF_PRE_GROW is true. If the operation fails, for example if the specified size is larger than available disk, the system did not clean up properly, and it could have resulted in the wrong information written to the root page. (#42147)

Hole in voting may result in corruption

After a mark/sweep has completed, the voting step handles references to objects from transaction that were committed 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. (#41584)

GemStone/S 64 Bit versions 2.4.4.7 and later included partial fixes for this bug. The fix in version 3.1 is a complete fix.

Removing element from KeyValueDictionary could result in incorrect size

When an element is removed from a KeyValueDictionary using `removeKey:otherwise:`, if the element is not present it is still possible for the cached size of the dictionary to be decreased. Sending the `size` message to the dictionary then does not return the actual number of entries. This results in errors when removing the last items, but rarely other symptoms. Most access and iteration operations do not use `size`. (#41950)

Tranlog generation and restore issues

Risk of bad tranlog record written with nonzero STN_GEM_TIMEOUT

The way transaction log records are generated when there is a gem terminated due to STN_GEM_TIMEOUT created a risk of writing a bad record to the tranlog. This is not visible when writing the tranlog, but when the transaction log containing this record is replayed, including during recovery after a shutdown, it fails with the error "unhandled recordKind". (#42114)

Memory leak in tranlog restore

During transaction log restore operations, there was a memory leak of variable size. This memory leak was primarily an issue for warm backups, which repeatedly perform tranlog restores; the stone would eventually run out of memory and shut down. (#42141)

Commit Record Backlog during restore

When restoring transaction logs containing a large amount of reclaim, it was possible for a commit record backlog to develop. This could use up all available free space and cause the restore to fail. (#42125)

Transaction log disks almost full handling

When transaction log disks are almost full, there have been problems seen where the transaction log root record or the first complete record could not be completely written, or many small transaction logs created. To avoid these issues, tranlogs are now created only when there is at least 1M of free space. (#42563, #41563)

Replayed persistent counter changes not visible until commitRestore

When there are changes to persistent counters recorded in transaction logs, when these were replayed, the changes were not visible until a later commitRestore. (#42216)

Tranlog close could have truncated log, resulting in corrupted tranlogs

It was possible for incorrect tranlog close logic to truncate the transaction log. The resulting transaction log was not usable for restore. (#42029)

This bug is in 3.0.1 only.

Shared counter changes recorded too soon in transaction logs

Changes to shared counters may have been written to the transaction log too soon, so they were out of sequence and replayed as part of the preceding checkpoint. If replay was stopped at that checkpoint, the state for the shared counter may have been inconsistent. (#42014)

Restore operations logouts reported as errors in RPC logs

When the session performing a restore operations completes, it terminates. If these gems were logged in RCP, previously the gem log would report an error and not be deleted, although the exit is normal. Now, these are handled as normal exists. (#42068)

Upgraded repositories did not have correct PositionableStream classes

In version 3.0, GemStone introduced a new set of PositionableStream classes, PositionableStreamPortable and several subclasses, while the previous implementation is PositionableStreamLegacy. The classes named PositionableStream, ReadStream and WriteStream refer to one or the other, depending on what you decide to install in your application.

On upgrade from 2.x, by default the legacy classes should have been installed, however, incorrectly the portable classes were installed. Now, the correct classes should be installed following upgrade. (#41911)

Numerics issues

ScaledDecimal // could round in the wrong direction

The division with floor operation, `//`, should result in the Integer that is closest to the precise value of the division result in the negative direction. However, for ScaledDecimal receiver and ScaledDecimal or Integer arguments, it is possible for the answer to be one greater than it should be. (#42196)

0.0 negated returned 0.0, not -0.0

IEEE 754 specifies that negating 0.0 should produce -0.0. (#42041)

Float roundTo: returns a SmallInteger 0 rather than a Float

If the result of `Float >> roundTo:` with a SmallInteger argument is zero, it should return the float 0.0. Previously, it returned SmallInteger 0. (#41947)

Subnormal Large Integers not converted by + 0

LargeIntegers that are in the range of SmallInteger can be converted to SmallIntegers by adding 0 or multiplying by 1. In 3.0,x, adding 0 did not perform the conversion. (#42244)

ScaledDecimal rounding bias

When `ScaledDecimals>>roundTo:` was sent to a value that was exactly halfway between two multiples of the argument, the rounding was always away from zero. This introduced a slight bias; the correct rounding for cases such as this is to round to the even multiple of the argument. (#41881)

to:by:do: handling of zero by: argument

If the method `Number>>to:by:do:` is passed a stepvalue of zero, ANSI specifies that it should return an error. (#42073)

Busy gem does not notice client's death

If an RPC gem is executing GemStone code, it may continue to execute code even if its client (such as a GBS application) is killed. Now, the gem will notice if the client process disappears and terminate. (#42238)

Allow %Z timezone specifier in GS_CFTIME and enforce format

GemStone allows you to customize the way timestamps are printed in log files using the GS_CFTIME environment variable. This previously always embedded the timezone, so including the %Z specifier within the GS_CFTIME resulted in the timezone included twice. (#41857)

In addition, logic has been added to verify that the GS_CFTIME contains sufficient information for debugging of log files following incidents. $GS_CFTIME must be a valid strftime format string containing fields for the month, day, hour, minute and seconds. If the setting does not meet these restrictions, default formatting is used.

Slow checkpoints with multiple AIO pgsvrs and fewer Free Frame pgsvrs

With multiple AIO page servers, and only one Free Frame page server, free pages may be returned to the free frame list in such a way that only one of the AIO page servers can write these pages to disk. During a checkpoint when pages are written to disk, the other AIO page servers may be idle while one does the writes, resulting in checkpoints taking much longer than necessary. (#41809)

The fix for this is to have the number of Free Frame page servers match the number of AIO page servers. The default for SHR_NUM_FREE_FRAME_SERVERS has been changed, so this will be the default; see "Changes in default for SHR_NUM_FREE_FRAME_SERVERS" on page 33.

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)

Unclear errors on failures in setting IO flags

When starting a stone with the configuration parameters STN_EXTENT_IO_FLAGS or STN_LOG_IO_FLAGS configured to 1, specifying to use direct I/O, on a file system that did not support direct I/O, the stone fails to start; this is expected behavior. The error messages in these cases were unhelpful in diagnosing and correcting the problem. (#42034)

Error from validatePasswordForUser:password:isEncrypted:

The method `System class>>validatePasswordForUser:password:isEncrypted:` signaled an error, rather than returning false, if the password failed to validate. (#41653)

This method also failed to check for the required privilege.

Corrupt object error referred to nil rather than actual object

In the case where an object is corrupted and an error returned from the C level, in some cases the error indicated that nil was corrupt, rather than the actual corrupt object. This has been fixed, so corrupt object errors refer to the corrupted object. (#41849)

Linux host stats computed incorrectly

Stats for memory pages on Linux were computed incorrectly, which resulted in values much smaller than the correct values. (#42153)

GCI application wrote login information to stdout

When a GemBuilder for C application performed a login, login details were always written to stdout. Now, when the flag GCI_LOGIN_QUIET, is included in the argument to GciLoginEx(), the output of login details is suppressed. (#42106, $42340)

canUnderstand: could introduce changes, affecting commit state

Seaside uses additional pathways to resolve methods, which are included in the base product method lookup. The policy governing this lookup was lazy initialized as an association in UserGlobals, if needed, by operations that perform method lookup such as `canUnderstand:`. The new association was a change that required commit. Now, all userProfiles have the default policy in UserGlobals. (#41433)

Invalid attempt to use Stone's cache as mid-level cache crashed rather than errored

Attempts to use the stone's cache as a mid-level cache resulted in a crash, rather than a connect error. (#41928)

searchlogs script could fail with large number of tranlogs

If the number of tranlogs being searched was large, an internal table could fill up and return errors when trying to open files. (#41630)

Classes subclassed from nil were versioned on recompile with no changes

GemStone no longer versions a class if the class is recompiled (for example by filein) but there are no changes in the class structure. This change was not made for classes that were subclasses of nil. (#42201)

Recompile of unconverted GsMethods failed

Following conversion from 2.x, methods are instances of GsMethod and have obsolete bytecodes; they must be recompiled into instances of GsNMethod. This can be done by filein (as specified in the *GemStone/S 64 Bit Installation Guide*) or by iterating through all classes and recompiling each method, using `Behavior>>recompileAllMethods`.

A bug in the method `GsMethod >> recompileIntoMethodDict:intoCategories:` caused this method to return an error on method recompile. (#42285)

GcGems deleted log files with #verboseLogging enabled

When the Admin or Reclaim GcGems are shut down, the log files were being deleted, even when #verboseLogging is enabled. Now, these log files are not deleted. (#41829)

Missing error message for #'objErrDiffSizeColl'

Using `replaceFrom:to:with:` or `with:do:` with incorrect argument collection size results in an error objErrDiffSizeColl. This error did not have an appropriate error message. (#41889)

UserProfile privileges cleanup

There were a number of methods on UserProfile in which the comments did not mention required Privileges, and some methods that did not enforce privileges. These have been cleaned up. (#41661, #41515)

PublishedSegment missing from Globals

With the rename of Segment to ObjectSecurityPolicy in v3.0, associations were set up so the named segments could resolve by name correctly to the ObjectSecurityPolicy. This was not done for PublishedSegment. (#42259)