
GemStone®

GemStone/S 64 Bit™

Release Notes

Version 3.3

February 2016



INTELLECTUAL PROPERTY OWNERSHIP

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

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

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

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

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

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

COPYRIGHTS

This software product, its documentation, and its user interface © 1986-2016 GemTalk Systems LLC. All rights reserved by GemTalk Systems.

PATENTS

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

TRADEMARKS

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

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

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.

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

Microsoft, **MS**, **Windows**, **Windows 7**, **Windows 2008**, and **Windows 8** 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.

Ubuntu is a registered trademark of Canonical Ltd., Inc., in the U.S. and other countries.

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

AIX, **POWER6**, **POWER7**, and **POWER8** are trademarks or registered trademarks of International Business Machines Corporation.

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

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

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



Preface

About This Documentation

These release notes describe changes in the GemStone/S 64 Bit™ version 3.3 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 3.3.

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

Terminology Conventions

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

Technical Support

Support Website

gemtalksystems.com

GemTalk’s website provides a variety of resources to help you use GemTalk products:

- ▶ **Documentation** for the current and for previous released versions of all GemTalk products, in PDF form.
- ▶ **Product download** for the current and selected recent versions of GemTalk software.

- ▶ **Bugnotes**, identifying performance issues or error conditions that you may encounter when using a GemTalk product.
- ▶ **TechTips**, providing information and instructions that are not in the documentation.
- ▶ **Compatibility matrices**, listing supported platforms for GemTalk product versions.

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

Help Requests

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

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

Website: techsupport.gemtalksystems.com

Email: techsupport@gemtalksystems.com

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

When submitting a request, please include the following information:

- ▶ Your name and company name.
- ▶ The versions of GemStone/S 64 Bit and of all related GemTalk products, and of any other related products, such as client Smalltalk products.
- ▶ The operating system and version you are using.
- ▶ A description of the problem or request.
- ▶ Exact error message(s) received, if any, including log files if appropriate.

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

24x7 Emergency Technical Support

GemTalk offers, at an additional charge, 24x7 emergency technical support. This support entitles customers to contact us 24 hours a day, 7 days a week, 365 days a year, for issues impacting a production system. For more details, contact GemTalk Support Renewals.

Training and Consulting

GemTalk Professional Services provide consulting to help you succeed with GemStone products. Training for GemStone/S is available at your location, and training courses are offered periodically at our offices in Beaverton, Oregon. Contact GemTalk Professional Services for more details or to obtain consulting services.



Table of Contents

Chapter 1. GemStone/S 64 Bit 3.3 Release Notes

Overview	13
Upgrade	13
New keyfiles required	13
Application code recompile required	13
Supported Platforms	14
Platforms for Version 3.3	14
GemBuilder for Smalltalk (GBS) Versions	14
VSD Versions	15
GemBuilder for Java (GBJ) Versions	15
GemConnect Versions	15
Documentation Changes	15

Enhancements and Changes

1. Internal and System Changes	16
Updated libraries	16
Multithreaded page servers and pageaudit	16
Compression using LZ4	16
Extent sizes multiple of 16MB.	16
Class name and instance variable limitations raised	17
Zoneinfo library updated	17
Improvements in support for debugging/single stepping	17
Block execution	17
With tranlogs set to /dev/null, no records sent	17
Support for 1GB huge pages on Linux.	17
2. Distribution and File Changes	18
Slow and no-op libraries and executables	18
Process log files and deletion policy	18
More specific run* scripts	19

32-bit GCI now available on AIX	19
Name change for Windows Client Distribution	19
3. Improvements related to encryption.	19
Encryption between remote gem and page server.	19
Updated SSL protocol	19
Querying for connection encryption status.	20
4. Deprecation and Obsolete code changes.	20
5. Numerics changes	22
SmallFraction special added	22
Change in printing	22
Float and SmallDouble no longer default to exponential notation.	22
Exponent letter E is now lowercase e	22
Added methods	22
All Floats now understand #asStringLocaleC, #fromStringLocaleC:	23
Parsing custom numeric literals	23
to:do: and to:by:do: handling	23
6. Changes related to Garbage Collection	24
Runtime configuration of ReclaimGem and AdminGem.	24
API to configure Reclaim Gem.	24
API to configure Admin Gem	24
removed configuration options	25
verboseLogging replaced.	25
New Reclaim Parameters.	25
Mark/sweep by FDC/MGC no longer available	26
Symbol Garbage Collection restored	26
stopGcSession: deprecated	26
7. Changes in Backup and Restore	26
Multi-threaded backup and restore now transparently adjust sessions.	26
Optimization in restoring backups	26
8. Improvements in handling extended character strings	27
Writing to logs now writes as UTF8 as needed	27
performOnServer: now accept input beyond ASCII range	27
Converting between Strings, UnicodeStrings, and Utf8s	27
Handling strings containing compositional UTF-8 bytes	27
Utf8 and ByteArray	27
Utf8 at:put: now disallowed	28
Changing Unicode Comparison Mode	28
9. Enhancements to Indexing/Querying	29
ReversedReadStream	29
OptionalPathTerm indexes specification moved	29
Introductory support for automatic management of selector path indexes	29
Error handling.	30
Warning on reduced conflict for identity indexes	30
Indexing Stream Classes modified	30
Set valued paths may not begin with asterisk	30
CharacterCollection >> asArrayOfPathTerms no longer accepts initial *	30

10. PetitParser changes	31
Cached instances require recompile	31
11. Hot Standby Changes	32
Add continuousRestoreFromArchiveLogs: variant with delay	32
Methods to get information about Hot Standby sessions	32
Smalltalk reporting on logsender and logreceiver	32
Other changes.	32
12. GsFile changes	33
Improved performance for server files.	33
Improvements in handling extended characters	33
In filenames	33
In code filed out to GsFile	33
GsFile managing file locks.	33
nextPutAsUtf8 extended.	34
GsFile atEnd now returns true for empty file	34
Other added method.	34
13. ProfMonitor new features and improvements	35
New profiling options	35
Profiling reports down to a percentage	36
14. FFI Changes.	37
C function that accepts an array of callback function addresses	37
Removed methods	37
GemStone version no longer hardcoded in image	37
15. Other added and changed methods	38
System Class	38
New Hidden Set protocol	38
Change in return value range from timeNs	38
Improved clock resolution, new profiling method	38
Object >> subclassResponsibility	38
System gemVersionReport and gemVersionAt: include buildType	39
Convenience methods to get commonly needed statistics	39
Collection classes	39
Added methods.	39
removeAllSuchThat: now returns removed elements	40
Class information and Fileout changes	40
Default class comment now reports hierarchy	40
Fileouts now include expectvalue /Class	40
Filing out to GsFile now automatically handles extended characters.	40
Class definition for old class versions now provides SymbolDictionary	40
Added method instancesInMemory	40
Socket classes	41
Date and DateTime changes.	41
Date class creation methods now error on out-of-range day.	41
Date today now respects TimeZone	41
DateTime creation performance improvement	41
ClassOrganizer changes	42
ClassOrganizer references to literals	42

sendersOf:* and optimized selectors	42
Repository	42
Repository createExtent:withMaxSize: minimum is 16MB	42
findReferencePath excludes internal implementation instances	42
GsExternalSession API for host login	42
GsProcess stack report that includes method source code	43

New Features

16. Transient Methods	44
17. Object Canonicalization Framework	44
18. Support for LDAP authentication without anonymous binds	45
19. Interface to write to syslog	47

Utility and Environment Changes

20. Changes in Statmonitor and Cache Statistics	50
Millisecond monitoring	50
VSD version	50
64-bit statistics	50
Statmonitor automatic startup for Stones and remote caches	51
File patterns specification with new add -F option	51
Change in behavior when existing filename specified	51
Change in filename numbering of first file for -r restart filenames	52
Directory specification; add -d	52
Filenames on automatic restart; add -R	52
Changes related to multi-threaded page servers and remote page servers	52
Sampling persistent shared counters; add -B	52
Added -G option	53
-h option	53
Added statistics	53
Renamed Statistics	54
Removed statistics	54
21. Changes in Configuration Parameters	55
Extent sizes always multiple of 16MB	55
Additional option to manage commit record backlog	55
Additional options for handing configuration parameters	55
Runtime changes no longer write to system configuration file	56
Use of units in configuration values	56
Added configuration parameter features	56
Encrypted connection between remote gem and page server	56
Automatically start statmonitor for remote cache	56
Automatically start statmonitor on mid-level cache	57
Limit number of unsuccessful mark-sweeps	57
Create CSV file with memory statistics on out-of-memory error	57
Add ability to nest configuration files	57
Define port range for pageservers for remote gems	58
On commit record backlog, send sigAborts to multiple sessions	58

Automatically start statmonitor on Stone startup	58
Configure the startup timeout on shared page cache	59
22. Changes in Errors	59
RT_ERR_UPGRADE_WARNING	59
STDB_ERR_BAD_NUMERIC_LIT	59
23. Utility Changes	60
copydbf	60
Added -u option	60
disallowed for backups on raw partitions	60
Backups now report position and sequence	60
gslis	60
Sorted results	60
Additional information	60
pageaudit	61
Multithreaded	61
Changes in output	61
Added -n option to set number of threads	61
Default name for pageaudit stone	61
pstack	61
Added options to collect C stacks only	61
removedbf	62
startcachewarmer	62
startnetldi	62
added -r option	62
waitstone	62
24. Environment variable changes	62
GS_DEBUG_SSL_LOG_DIR disallowed in a fast build for security reasons	62
Additional variables to manage log deletion	62
25. Topaz changes	63
Single-line command serializing	63
UTF-8 related changes	63
fileformat UTF8 in .topazini now applies to subsequent input	63
topaz output files now UTF-8 encoded	63
Topaz executable argument changes	64
Add -C option to pass in arbitrary configuration parameter values	64
-e, -z, -C and -T disallowed with -r	64
Warning on configuration file not found	64
Line limit increased	64
New commands	64
Added EXEC command	64
Support for setting transaction mode, add SET TRANSACTIONMODE	64
Added DEBUGRUN command	65
Added LITERALS command	65
DISPLAY and OMIT added PAUSEONWARNING	65
Changes in the ordering of lines within status output	65
topaz fileout uses protocol in Behavior	65

26. GCI changes	66
Added functions.	66
GciAll7Bit.	66
GciExecuteStrFetchBytes, GciNbExecuteStrFetchBytes	66
Changed functions	67
Deprecated functions	67
Removed Functions	67
Other Changes affecting the GCI interface	68
User actions that invoke Smalltalk nesting limit is now two	68
Linux Compile and Link Information.	68
Complier version	68
Debugger version	68
Compiling a user action or GCI application	68
Linking a user action library	68
Linking a GCI application	69
Solaris on SPARC Compile and Link Information.	69
Complier version	69
Debugger version	69
Compiling a user action or GCI application	69
Linking a user action library	69
Linking a GCI application	69
Solaris on x86 Compile and Link Information	69
Complier version	69
Debugger version	69
Compiling a user action or GCI application	70
Linking a user action library	70
Linking a GCI application	70
AIX Compile and Link Information.	70
Complier version	70
Debugger version	70
Compiling a user action or GCI application	70
Linking a user action library	70
Linking a GCI application	71
DARWIN Compile and Link Information	71
Complier version	71
Debugger version	71
Compiling a user action or GCI application	71
Linking a user action library	71
Linking a GCI application	71
Windows Compile and Link Information	71
Complier/Debugger version.	71
Compiling a GCI application.	72
Linking a GCI application	72

Chapter 2. Bugs Fixed

AIO page server fatal errors during startup may leave Stone hanging.	73
Multithreaded scans now usable from remote Gems.	73
On first use of a remote host, linked logins may fail	73
Garbage Collection issues	73
Symbol Garbage Collection not usable.	73
markGcCandidates issues	74
Exception Handling issues	74
Exception handler #resume from primitive operation returned nil.	74
Incorrect return from exception handler with non-local return in ensure: block	74
Blocks may return wrong values when nested variable is assigned in outer scope	74
Unable to trap soft break in linked topaz	74
#halt returned the result of resume, not receiver	74
A do: loop on an Interval does not behave correctly with step through	75
Some ANSI exception classes inherited error number	75
Improvements in Error reporting.	75
CompileError display	75
Improved error reporting for String and MultiByteString>>encodeUsing:	75
Improved error for attempt to store into internal class.	75
Error message referencing primitive -1/-2 does not exist is confusing.	75
midLevelCacheConnect: error could return empty string rather than error	75
OutOfRangeException errors in setting configuration values did not include details	75
Improved error when using a name that matches a system service.	76
Kernel methods not respecting transient changes to symbol list	76
Binary selectors /- and /-/ failed.	76
Index and Query issues	76
GsQuery readStream with equality does not correctly handle cases where comparable objects are not equal	76
removeAllIndexes could run out of memory	76
Index audit does not catch entries with equal keys, but OOPS out of order	76
IndexManager usageReport may get MNU #nextWord with Portable Streams installed	76
Linux network statistics incorrectly relative to start of statmonitor	77
GEM_PGSVR_COMPRESS_PAGE_TRANSFERS only compressed one direction	77
GciEnableFreeOopEncoding and GciGetFreeOopsEncoded broken	77
NetLDI socket leak on incompatible connection	77
StringKeyValueDictionary removeKey: may fail incorrectly	77
StringKeyValueDictionary>>at:ifAbsent: did not use block on nil argument	77
Risk of SEGV when VW on Linux invokes C code	77
Topaz issues.	78
Linked topaz retained some changes to internal globals over logins	78
Topaz nbstep command broken.	78
Topaz shell does not work well with lineeditor	78
Topaz filename expansion did not recognize ~ in path	78

Hot and Warm Standby issues.	78
Standby system may lag behind primary significantly after large reclaim	78
Hot standby connection may timeout with large number of tranlogs. . .	78
Possible for logreceiver to get tranlogs from multiple repositories	78
fork: did not handle single quotes correctly	78
Checkpoints not written during replay of possible dead objects	79
GsFile does not handle file names with extended characters.	79
startcachewarmer handling of NRS in stone name	79
Upgrade and Conversion issues.	79
Conversion problems with very large Symbols	79
Filein class definitions that fail in _equivalentSubclass:... did not provide details	79
ClassOrganizer instances persisted through upgrade had missing symbol list 79	
Upgrade issues with #GemStoneRCLock.	80
filein of older code could introduce comment class method	80
Improved handling of login when repository upgrade required	80
SymbolGem that is slow to exit is made invalid, not handled cleanly	80
Compiling method with Array SymbolDictionary required #OtherPassword privilege	80
Inconsistent lock status after failed commit on temporary locked object	80
SlotsTotalCount off-by-one.	80
gslis -x did not show startnetldi -D value	80

GemStone/S 64 Bit 3.3 Release Notes

Overview

GemStone/S 64 Bit 3.3 is a new version of the GemStone/S 64 Bit object server. This release provides further multithreading to improve performance, adds a number of new features, and fixes a number of bugs in v3.2.12; we recommend everyone using or planning to use GemStone/S 64 Bit upgrade to this new version.

These release notes provide changes between the previous version of GemStone/S 64 Bit, version 3.2.12, and version 3.3. If you are upgrading from a version prior to 3.2.12, 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.

For details about installing GemStone/S 64 Bit 3.3 or upgrading from earlier versions of GemStone/S 64 Bit, see the *GemStone/S 64 Bit Installation Guide* for v3.3 for your platform.

Upgrade

Upgrade is supported from GemStone/S 64 Bit v2.4.x, 3.1.x, and 3.2.x. For details on the conversion and upgrade, see the *GemStone/S 64 Bit Installation Guide* for v3.3.

New keyfiles required

New keyfiles are required for version 3.3. To obtain a new keyfile for GemStone/S v3.3, write to keyfiles@gemtalksystems.com. In your request, include your license information, platform and any updates to contact information. It may be helpful to include your current keyfile with the request.

Contact GemTalk Technical Support if you have issues or questions.

Application code recompile required

The upgrade process from 3.2.x or earlier to 3.3 requires all method source code be recompiled.

Supported Platforms

Platforms for Version 3.3

GemStone/S 64 Bit version 3.3 is supported on the following platforms:

- ▶ Solaris 10 and 11 on SPARC
- ▶ Solaris 10 on x86
- ▶ AIX 6.1 on POWER6 and POWER7 and AIX 7.1 on POWER7 and POWER8
- ▶ Red Hat Enterprise Linux Server 6.4, 6.5, 6.7, and 7.1; Ubuntu 12.04; and SUSE Linux Enterprise 12, all on x86
- ▶ OS X 10.9.5 (Mavericks), with Darwin 13.4.0 kernel, on x86

For more information and detailed requirements for each supported platforms, please refer to the GemStone/S 64 Bit v3.3 Installation Guide for that platform.

GemBuilder for Smalltalk (GBS) Versions

GemStone/S 64 Bit version 3.3 requires a new version of GBS, version 8.1. Older versions of GBS cannot login to v3.3. For details on the changes in GBS v8.1, see the *GemBuilder for Smalltalk Release Notes* for v8.1

The following versions of GBS are supported with GemStone/S 64 Bit version 3.3.

GBS version 8.1

VisualWorks 7.10.1 32-bit and 64-bit
<ul style="list-style-type: none"> ▶ Windows 8, Windows 2008 R2 and Windows 7 ▶ Solaris 10 on SPARC ▶ RedHat ES 6.4, 6.5, 6.7, and 7.1, SUSE ES 12, and Ubuntu 12.04

GBS version 5.4.3

VA Smalltalk 8.6.2	VA Smalltalk 8.5.2
<ul style="list-style-type: none"> ▶ Windows 8, Professional or above ▶ Windows 2008 R2 ▶ Windows 7, Professional or above 	<ul style="list-style-type: none"> ▶ Windows 2008 R2 ▶ Windows 7

For more details on supported GBS and client Smalltalk platforms and requirements, see the *GemBuilder for Smalltalk Installation Guide* for that version of GBS.

VSD Versions

The GemStone/S 64 Bit v3.3 distribution includes VSD version 5.1.1. The previous version of GemStone/S 64 Bit, v3.2.12, included VSD v5.1.

Changes between VSD v5.1 and v5.1.1 include :

- ▶ changes in the statistic type for a number of statistics, affecting combined graphs.

For more details, see the [Release Notes for VSD v5.1.1](#).

GemBuilder for Java (GBJ) Versions

GBJ users must upgrade to GBJ 3.1.2 along with the upgrade to GemStone/S 64 Bit v3.3. Earlier versions of GBJ are not compatible with v3.3.

The GemStone/S 64 Bit v3.3 distribution includes the shared libraries for GBJ v3.1.2.

Note that GBJ 3.1.2 now requires Java 1.8.0.

For more information on changes in GBJ v3.1.2, see the *Release Notes for GBJ v3.1.2*

GemConnect Versions

The current version of GemConnect, v.2.3, is compatible with GemStone/S 64 Bit v3.3, so upgrade is not needed if you are running GemConnect 2.3.

The exception is the Macintosh; recent versions of Darwin have introduced incompatibilities with GemConnect. Contact GemTalk Technical Support if you need to use GemConnect with v3.3 on the Mac.

Documentation Changes

All Documentation has been revised for this release, with modifications to incorporate new and changed features, as well as corrections and improvements.

- ▶ The *Topaz User's Guide* has been extensively updated, with Chapter 1 reorganized and largely rewritten. The *Topaz User's Guide* is now available in .html as well as .pdf
- ▶ The graphics for the *System Administration Guide* and the *Programmer's Guide* have been updated to be more informative and easier to read, as well as more attractive.
- ▶ The *System Administration Guide* has extensive updates to Chapter 3 on distributed configurations, and reorganized material on GemStone processes and log file management in Chapter 5.
- ▶ The GBS and GBJ *Release Notes* and *Installation Guide* are now available as .html as well as .pdf.

Enhancements and Changes

1. Internal and System Changes

Updated libraries

Several open-source libraries included with GemStone/S 64 Bit have been updated:

- ▶ OpenSSL continues to release security advisories. The version of OpenSSL used by GemStone/S 64 Bit v3.3 has accordingly been updated to 1.0.2f.
- ▶ OpenLDAP has been updated to 2.4.40.
- ▶ ICU has been updated to version 54.1.

Multithreaded page servers and pageaudit

The pgsvr page server processes, including AIO and Free Frame page servers, and page servers for remote gems, are now multithreaded. When multiple page servers were previously used, these are now threads within a single pgsvr process, rather than multiple pgsvr processes, and write to a single log file.

The configuration parameters SHR_NUM_FREE_FRAME_SERVERS and STN_NUM_LOCAL_AIO_SERVERS configure the number of threads, not the number of processes.

All Gems on a remote cache will share a single pgsvr process on the stone's machine.

The pageaudit utility has also been multithreaded, and will complete much more quickly. This is described under "pageaudit" on page 61.

Compression using LZ4

Internal compress and uncompress, particularly compression of pages transmitted to remote processes are now compressed using LZ4 instead of Zlib (gzip).

LZ4 provides much faster performance for these operations. Configurations in which interprocess compression did not provide performance improvements should reevaluate under version 3.3.

Externally visible compression continues to use Zlib; the changes to LZ4 should not impact normal operations, such as statmonitor/VSD, GsFile, compression of backups and transaction logs, and GciCompress/GciUncompress.

The GemStone distribution now includes LZ4 v1.3.1, and associated open-source license file.

New cache statistics have been added to track the performance of compression and decompression; see under "Added statistics" on page 53

Extent sizes multiple of 16MB

The minimum extent size is now 16MB, and extent now grow by units of 16MB. This is much more efficient for modern disk sizes. For details, see "Extent sizes always multiple of 16MB" on page 55.

Class name and instance variable limitations raised

Class names may now be up to 1024 characters in length, and may have up to 2030 instance variables.

Zoneinfo library updated

GemStone includes a distribution of the tz or zoneinfo Time Zone database, under `$GEMSTONE/pub/timezone`. The version distributed has been updated to version 2015d. For the latest versions, see <http://www.iana.org/time-zones>.

Note that Linux and Solaris systems normally have this Time Zone Database installed as part of the operating system, and many methods in GemStone TimeZone classes will use the OS version by default on these platforms.

Improvements in support for debugging/single stepping

The code in `GsProcess` and other methods that supports stepping through and into code, which is used in GBS and topaz debugging, has been updated and refactored, and the API has been changed. A number of private methods have been removed from `GsProcess`, and other methods added.

Related to this, several GCI stepping functions have been deprecated.

Block execution

Version 3.3 includes new bytecodes and code changes to fix issues with Block execution.

Note that `to:do:` and similar loops may not be optimized; moving temps inside the loop may help restore the optimization.

With tranlogs set to /dev/null, no records sent

Setting the tranlog directories (`STN_TRAN_LOG_DIRECTORIES`) to `/dev/null` avoids the overhead of writing tranlog records to disk, at the risk of being unable to recover from unexpected shutdown.

Previously, while the disk write itself was not done, the tranlog records were still created and sent. Now, the gems themselves do not send tranlog records in the case where the tranlogs are set to `/dev/null`.

There are additional restrictions on tranlog directories set to `/dev/null`; either all items in the tranlog directories must be null, or none. A mix is disallowed and will raise an error.

Support for 1GB huge pages on Linux

The shared page cache can now be configured to use 1GB memory pages on Linux. This is configured in the same way as 2MB pages are configured. See the *Installation Guide* for more information.

2. Distribution and File Changes

Slow and no-op libraries and executables

When diagnosing problems, it can be useful to run slow or no-op Gems, which are not optimized and provide more internal state information and, in the case of slow executable, assertion checks. This can now be done more easily by using new components in the product distribution.

Slow and not-optimized (no-op or noop) libraries are now provided as `libgcilnk_slow-NN-64.so` and `libgcilnk_noop-NN-64.so`, in the `$GEMSTONE/lib` directory. The directory `$GEMSTONE/libslow/` is no longer included.

The `$GEMSTONE/sys` directory includes the executables for `gem`, `pgsvrmain` and `stone`. In v3.3, both `.slow` and `.noop` executable variants of each of these are also included.

The following `gemnetobject` variants will cause a no-op or slow gem to be run:

```
$GEMSTONE/sys/gemnetobject_noop
$GEMSTONE/sys/gemnetobject_slow
```

To run a slow or noop gem, you can invoke the new variant in place of `gemnetobject` in your GemStone login parameters.

Process log files and deletion policy

Handling of log files on clean exit has been reviewed and made consistent. These are the updated rules:

- ▶ Stone, NetLDI, logsender and logreceiver logs are never automatically deleted.
- ▶ When a process terminates abnormally, from an error or kill -TERM, its log is not deleted, regardless of any environment settings. The SymbolGem is also preserved from deletion if an error occurred at any point in the process's lifetime.
- ▶ If the environment variable `GS_KEEP_ALL_LOGS` is set, no log files are deleted.
- ▶ If the environment variable `GS_FORCE_CLEAN_LOG_FILE_DELETE` is set, then all log files for processes that exit cleanly are deleted.
- ▶ The environment variable `GEMSTONE_KEEP_LOG` controls whether a particular process log file is deleted on clean shutdown. These are set in the process startup script `$GEMSTONE/sys/run*`. This can be modified this for individual process scripts for finer control.
 - ❑ SPC Monitor, PageManager, AdminGem and ReclaimGem logs are not deleted by default.
 - ❑ The logs for an RPC Gem, Symbol Gem, AIO Page Server, Free Frame Page Server, and most other processes are deleted by default.
 - ❑ Logs for remote processes are generally not deleted.

As a convenience to allow the RPC Gem log to be retained on clean exit, the following has been added:

```
$GEMSTONE/sys/gemnetobject_keeplog
```

Using this in place of `gemnetobject` in the gem login parameters allows the gem log to be retained on clean exit.

The names of log files for processes associated with remote processes has also been cleaned up to be easily identifiable and consistent with the naming for local process logs.

More specific run* scripts

Changes have been made to the set of `$GEMSTONE/sys/run*` scripts that are used to startup GemStone processes. New scripts have been added that are more specific to particular processes. This allows per-process type customization as needed, such as control over log file deletion.

The following scripts have been added:

```
runaiopgsvr
runffpgsvr
runcachepgsvr
rungempgsvr
runpageauditgem
runstatmonitor
```

32-bit GCI now available on AIX

In previous v3.1.x and 3.2.x releases, the 32-bit applications and libraries in `$GEMSTONE/bin32` and `$GEMSTONE/lib32` were not available on AIX. These have been restored.

Name change for Windows Client Distribution

The Windows client was previously provided with the name

```
GemBuilderCN.N.N-x86.Windows_NT.zip
```

This was not very descriptive; this has been renamed, and is now:

```
GemStone64BitClientN.N.N-x86.Windows_NT.zip
```

The Installation Guide to use with the Windows Client is *GS64-WindowsClient-3.3.pdf*; this new name was introduced in v3.2.6.

3. Improvements related to encryption

Encryption between remote gem and page server

To create an SSL connection between a remote Gem and its page server on the stone's host and the mid level cache, if used, a new configuration option has been added; `GEM_PGSRV_USE_SSL`. See "GEM_PGSRV_USE_SSL" on page 56. This option enables SSL between both the gem and the page server on the stone host and the page server mid cache, if any. The SSL tunnel is established in anonymous mode, meaning communications are encrypted but the sender and receiver are not authenticated.

Updated SSL protocol

The SSL protocol has been updated from TLS 1.1 to 1.2.

Querying for connection encryption status

The following methods have been added:

System class >> gciClientConnectionIsEncrypted

Answer a Boolean indicating whether the network connection to the gem's GCI client is encrypted (using TLS). Always answers false for linked gems and for gems running on the same host as the GCI client.

System class >> pageServerConnectionIsEncrypted

Answer a Boolean indicating whether the network connection to the gem's page server is encrypted (using TLS). Always answers false for gems without a page server, i.e., gems that run on the same host as the stone process.

4. Deprecation and Obsolete code changes

The following methods have been deprecated:

ProfMonitor class >> spyOn:

Use monitorBlock: or one of its variants.

System class >> stopGcSession:

Use stopReclaimGem or stopAdminGem.

SymbolDictionary >> textForError:args:

Repository>>cleanupFdcResultsForMgc:

Use the new method cleanupFdcResults

The following methods related to indexing have also been deprecated:

CollectionBasedPathTerm >> allDependencyListsFor:into:

EnumeratedPathTerm >> allDependencyListsFor:into:

PathTerm >> allDependencyListsFor:into:

PathTerm >> auditDirectNscCounts:for:on:count:

PathTerm >> auditNscCounts:for:on:count:

SetValuedPathTerm >> allDependencyListsFor:into:

UnorderedCollection >> _auditAndRepairDepListsLogging:

UnorderedCollection >> _getAllDependencyLists

The following obsolete methods have been removed:

System class >> hstLogCategory:setup:

System class >> hstLogCategoryDisable:

System class >> hstLogCategoryEnable:

System class >> hstLogOpcode:setup:

System class >> hstLogOpcodeDisable:

System class >> hstLogOpcodeEnable:

System class >> hstLogStatus

System class >> prvLogPrint

System class >> prvLogRecordX1:X2:X3:

System class >> shrLogPrint

System class >> shrLogRecordX1:X2:X3:

The following private methods have been replaced by public methods, with the same name excluding the leading underscore. For details, see section starting on page 39.

```
System class >> _commitRecordBacklog
System class >> _deadNotReclaimedCount
System class >> _possibleDeadSize
System class >> _possibleDeadSymbols
System class >> _reclaimedSymbols
System class >> _remoteSharedCacheCount
System class >> _scavengablePagesCount
System class >> _totalSessionsCount
```

The following removed methods are replaced by the equivalent public (non-underscored) method in System class:

```
Repository >> _shadowPagesCount
Repository >> _deadNotReclaimedCount
```

The following obsolete classes have been moved into Globals at: #ObsoleteClasses.

```
ObsoleteTimeZone           StackBuffer
Process                   StackSegment
OldRepository
```

5. Numerics changes

SmallFraction special added

SmallFraction is a new class in this release. SmallFraction is a special, an object for which the value is encoded in the OOP. Specials do not require a individual object in the repository, and so do not need to be loaded into the repository or into memory, making them much more efficient.

SmallFractions can hold fractional values with numerators between -536870912 and 536870911, and denominators greater than 0 and less or equal to 134217727 .

SmallFraction is related to Fraction similarly to how SmallDouble and Float or SmallInteger and Integer relate; Fraction creation or operations on Fractions that produce results within the SmallFraction range will return an instance of SmallFraction, and SmallFraction operations with results outside of the range will return an instance of Fraction.

The hierarchy for Fraction is changed in this release; the class AbstractFraction has been added as the superclass for Fraction as well as SmallFraction.

Change in printing

Float and SmallDouble no longer default to exponential notation

The default printing for Floats has changed. Previously, SmallDouble and Float by default printed in exponential notation, with 16 digits following the decimal point. Now, the default print format prints in decimal notation, with only as many digits as required; this uses the C printf format string `%.16g`.

So for example, a SmallDouble created from the literal 12.7 would previously print `1.2699999999999999E01`; now it prints as `12.7`.

The previous output format can be specified using the new `asStringLegacy` methods listed below, or by using `asStringUsingFormat`:

This change also applies to SmallFloat; instances of SmallFloat use the C printf format string `%.7g`. Note that SmallFloat remains deprecated in this release, although full support may be restored in a future release.

Exponent letter E is now lowercase e

Float exponents are now printed using `e` rather than `E`; `E` is not part of the ANSI standard. This affects printing with `asStringUsingFormat`. The previous behavior can be accessed via new methods, listed in the following section.

Added methods

The following methods have been added to provide the printing behavior from previous releases.

```
Float >> asStringLegacy
SmallFloat >> asStringLegacy
Float >> asStringLocaleCLegacy
Float >> asStringUsingFormatLegacy:
```

All Floats now understand #asStringLocaleC, #fromStringLocaleC:

The methods `asStringLocaleC` and `fromStringLocaleC` are now all provided in `ScaledDecimal`, `FixedPoint`, and `SmallFloat`, as well as `Float` and `SmallDouble`, to allow reading and printing to use \$. for the decimal separator regardless of the locale. This is useful for creating Locale-independent external output that will be read into GemStone again.

Parsing custom numeric literals

The GemStone compiler recognizes a number of numeric literals in addition to integers and floats; for example, `ScaledDecimals` can be specified using the literal syntax `1.23s2`. If your application has defined a new numeric type, is it now possible to define a new literal syntax, and have the compiler handle this correctly to create instances of your numeric class. The new literal syntax can be of the form `Nc` or `NcM`, where `N` and `M` are numbers and `c` is an alphabetic character.

The following message provides a way to instruct the compiler to recognize a new literal format for a numeric class:

```
NumberSubclass parseLiterals: aCharacter exponentRequired: aBoolean
```

NumberSubclass must be a subclass of `Number`. Once this is sent, when the compiler encounters a numeric value using *aCharacter*, it will invoke *NumberSubclass* >> `fromString:` to create an instance of *NumberSubclass*.

The subclass of `Number` must implement `fromString:` in such a way as to be able to read the literal format and create the new instance.

aCharacter must be an alphabetic Character with `codePoint <= 127`, and may not be an existing numeric literal character. The following are not allowed: `$e $E $d $D $f $F $p $s $q $r`.

aBoolean indicates if digits following *aCharacter* are required or not. If *aBoolean* is `nil`, it uninstalls a previously-installed literal class.

Once invoked, the new literal format will be recognized until the session logs out.

Note that for subsequent logins, compiled references to that literal will continue to be valid, but unless the method is invoked again, methods with that literal cannot be recompiled. Including the invocation in session initialization code (such as using `loginHook:`) is recommended.

to:do: and to:by:do: handling

The optimized selectors `to:by:` and `to:by:do:` are only applicable to classes that have a sequencable control value. Previously, these methods were provided in `Number`; however, this created issues for application classes that implemented sequenceable kinds of objects not under the `Number` hierarchy. These methods have been moved to `Object`; although, note that they will fail if the subclass does not implement minimal arithmetic protocol.

6. Changes related to Garbage Collection

Runtime configuration of ReclaimGem and AdminGem

Previously, the only way to update Reclaim and Admin Gem's configuration was to modify values for the appropriate keys in the GcUser's UserGlobals, and, provided #autoRefreshGcGemConfig was set to true, waiting for the GcGem to notice and update.

These can now be set at runtime without commit, and take effect immediately. A particular value of this feature is to configure reclaim in warm and hot standby systems, which previously were limited to the identical ReclaimGem configuration as required by their primary production system.

Runtime configuration changes made using these new methods are not persistent; to make a change that will persist after the Reclaim or Admin restart, you must also edit the value in GcUser's UserGlobals.

When you make persistent changes in GcUser's UserGlobals, these changes take effect some time subsequent to the commit, when the Reclaim or Admin Gem commits or aborts. On idle systems, there may be a delay before these changes take effect; the frequency of checking has changed in this release. Note that #autoRefreshGcGemConfig is no longer used, so changes to GcUser's UserGlobal settings will always take effect.

API to configure Reclaim Gem

System now has methods to fetch and set the Reclaim Gem configuration values.

```
System class >> getReclaimConfig: configSymbol
System class >> setReclaimConfig: configSymbol toValue: aVal
```

The configuration configSymbols that the reclaim methods accept are:

```
#deadObjsReclaimedCommitThreshold
#deferReclaimCacheDirtyThreshold
#maxTransactionDuration
#objectTableCleanupIdleTime
#objsMovedPerCommitThreshold
#reclaimDeadEnabled
#reclaimMinFreeSpaceMb
#reclaimMinPages
#reclaimVerboseLogging
#sleepTimeBetweenReclaimMs
#sleepTimeWithCrBacklogMs
```

API to configure Admin Gem

System now has methods to fetch and set the Reclaim Gem configuration values.

```
System class >> getAdminConfig: configSymbol
System class >> setAdminConfig: configSymbol toValue: aVal
```


The configuration configSymbols that the admin methods accept are:

```
#adminVerboseLogging
#epochGcMaxThreads
#epochGcPageBufferSize
#epochGcPercentCpuActiveLimit
#epochGcTimeLimit
#epochGcTransLimit
#saveWriteSetUnionToFile
#sweepWsUnionMaxThreads
#sweepWsUnionPageBufferSize
#sweepWsUnionPercentCpuActiveLimit
```

removed configuration options

The following parameters are obsolete in v3.3, and have been removed:

```
#autoRefreshGcGemConfig
#enableDebugging
```

verboseLogging replaced

In order to complete the separation of the set of parameters that apply to the ReclaimGem vs. to the AdminGem, #verboseLogging has been removed, and replaced by #adminVerboseLogging and #reclaimVerboseLogging. Both of these default to false.

New Reclaim Parameters

The following have been added:

#sleepTimeWithCrBacklogMs

Amount of time (in milliseconds) to sleep after a commit when the commit record backlog is larger than $1.25 * \text{the current setting for STN_CR_BACKLOG_THRESHOLD}$. For each 25 percent above the threshold the sleep time is increased so that the ReclaimGem does fewer commits the higher the number of commit records is above the threshold.

```
min 0
max 300000
default: 0
```

This configuration is designed to reduce the impact of application sessions that hold the oldest transaction for an extended period, which can create a commit record backlog. A busy ReclaimGem performs a large number of commits (depending on how it is configured), which makes the commit record backlog grow quickly. It can be a problem when there is a timing coincidence of a commit record being held and a large reclaim operation. With this configuration parameter set, when a commit backlog is developing, the Reclaim Gem commit rate is automatically slowed down. This reduces the potential impact of the timing of reclaim on a commit record backlog.

#objectTableCleanupIdleTime

Controls whether internal objectTableCleanup is enabled, and sets the time in seconds that system is idle (no commits) before the reclaim gem attempts to perform the cleanup. The default value, 0, means disabled.

```
default: 0
```

Mark/sweep by FDC/MGC no longer available

In previous releases, an alternative to `markForCollection` has been a `findDisconnectedObjects/markGcCandidates` set, generally used with FDC occurring on an offline copy of the repository. With the introduction of multi-threaded `markForCollection`, the offline FDC/MGC combination is no longer necessary for performance.

Changes in the multithreading scan may have introduced instability into the MGC, so this feature has been removed. The following methods are removed:

```
basicMarkGcCandidatesFromFile:forceOnError:readOnly:
cleanupBigArray:forMgc:
cleanupMgcResultsForMgc:
markGcCandidatesFromFile:
markGcCandidatesFromFile:forceOnError:
markGcCandidatesReadOnlyFromFile:
```

FDC remains available as an analysis tool.

The method `Repository >> cleanupFdcResults` has been added to replace the deprecated method `Repository >> cleanupFdcResultsForMgc:`.

Symbol Garbage Collection restored

Symbol garbage collection was disabled in the previous version, due to bug 45403. It is re-enabled in v3.3; you may now choose to set `STN_SYMBOL_GC_ENABLED` to `TRUE`.

stopGcSession: deprecated

Stopping garbage collection session by session number is deprecated; use `stopReclaimGem` or `stopAdminGem` instead.

7. Changes in Backup and Restore

Multi-threaded backup and restore now transparently adjust sessions

Multi-threaded backup and multi-threaded restore compute the number of sessions based on CPUs and extents. If this number of sessions is not available (e.g., when many other sessions are logged in), backup and restore now use fewer sessions, as available, and write a message to stone log.

Optimization in restoring backups

The code that performs the restore operation has been refactored to change the way the new object table is managed. This improves performance; testing indicates a 30% improvement in CPU time. Note that the actual performance improvement may also be limited by I/O. The changes are in the backup file; backups from earlier versions cannot be restored into 3.3, which is not a supported operation in any case.

8. Improvements in handling extended character strings

Kinds of `String` that contain `Characters` outside the ASCII range (those with `codePoints` above 127) require additional handling when interfacing to external systems, including the file system, operating system, etc. There are a number of improvements in this area in v3.3.

Writing to logs now writes as UTF8 as needed

Messages that are written to the stone log using `System class >> addAllToStoneLog:`, or to the GCI client or server logs using `gciLogClient:` and `gciLogServer:`, are now encoded as UTF8.

This allows all kind of `String` containing any characters to be written to the logs.

performOnServer: now accept input beyond ASCII range

`System >> performOnServer:` accepts a string argument; this argument can now be a traditional or `Unicode` string or a `Utf8`, and may include `Characters` outside the ASCII range.

Note that the results are always returned as a `String` containing byte characters, in the range up to code point 254. The application must determine if the result is 8-bit or UTF8, and send `decodeFromUTF8` to the result as needed.

Converting between Strings, UnicodeStrings, and Utf8s

Handling strings containing compositional UTF-8 bytes

A legacy `String` may hold bytes that are the compositional bytes of a UTF-8 encoded string. This is not valid state for `Unicode Strings`.

To convert a legacy `String` containing compositional bytes of a UTF-8 encoded string into a decoded legacy or `unicode` string, the following methods have been added:

```
String >> decodeFromUTF8ToString
```

The receiver's bytes are the UTF-8-encoding; decode the bytes and return a new instance of `String`, `DoubleByteString`, or `QuadByteString`.

```
String >> decodeFromUTF8ToUnicode
```

The receiver's bytes are the UTF-8-encoding; decode the bytes and return the `Unicode7`, `Unicode16`, or `Unicode32` representation.

`String >> decodeFromUTF8` has been un-deprecated.

The method `String class >> withBytes:` now accepts a `Utf8` argument, and will create a `String` with the still-encoded bytes.

Utf8 and ByteArray

`ByteArray` and `Utf8` instances are collections of bytes (integers in the range 0-255). `ByteArray` places no limitation or interpretation on its contents, while `Utf8` requires that they be legal sequences of UTF-8 encodings.

This creates an ambiguity in converting to a `String`, since the intended result may or may not be the decoded version.

The following methods are available for both ByteArray and Utf8, with the behavior:

`decodeToString`

Returns the decoded legacy string.

`decodeToUnicode`

Returns the decoded Unicode string.

`bytesIntoString`

Returns an instance of String containing the bytes of the receiver without doing any decoding.

`bytesIntoUnicode`

Returns an instance of a Unicode string containing the bytes of the receiver without doing any decoding. Since a Unicode string should never hold encoded bytes, this method is disallowed for Utf8. For ByteArray, it is assumed that the contents are not UTF-8 encoded.

It is recommended to use these methods rather than `asString` or `asUnicodeString`.

Utf8 at:put: now disallowed

It is disallowed to directly manipulate an instances Utf8 using `at:put:`; Utf8 instances should be constructed as strings and then encoded. Utf8 previously inherited the `at:put:` implementation. Attempting to use this method will now raise an error.

Changing Unicode Comparison Mode

Unicode Comparison Mode makes it easier to work when there are combinations of Legacy and Unicode String data. In Unicode Comparison Mode, all legacy string as well as Unicode string comparisons use the ICU libraries. This avoids inconsistencies and disallowed operations in comparison rules between Legacy and Unicode strings.

Note that applications with existing Legacy String data in dictionaries and indexes should not change the Unicode Comparison Mode without performing manual resorts/rebuilds, since it is possible, although rare, for ordering to change and lookups to fail.

The following methods provide an interface for changing the setting and testing for Unicode Comparison Mode.

`CharacterCollection class >> disableUnicodeComparisonMode`

Disable Unicode Comparison Mode; restore legacy handling for Legacy String classes. This method may only be executed by SystemUser. This method performs a commit; the change takes effect on subsequent logins. This method returns the previous value of `isInUnicodeComparisonMode`.

`CharacterCollection class >> enableUnicodeComparisonMode`

Enable Unicode Comparison Mode, so all String comparison use ICU library comparison functions. This method may only be executed by SystemUser. This method performs a commit; the change takes effect on subsequent logins. This method returns the previous value of `isInUnicodeComparisonMode`.

`CharacterCollection class >> isInUnicodeComparisonMode`

Return if the current session operates in Unicode Comparison Mode. This does not reflect changes to the Unicode Comparison Mode status that were made subsequent to this session's login.

These methods are implemented in `CharacterCollection`, so can be sent to any `String` class. It is recommended that you send these messages to the class returned by the Global reference `#StringConfiguration`. For example:

```
StringConfiguration isInUnicodeComparisonMode
```

`StringConfiguration` is a key in `Globals` that currently may refer to `String` or to `Unicode16`, but may in the future provide more ability to manage strings. Using `StringConfiguration` will accommodate future changes to this reference.

9. Enhancements to Indexing/Querying

There have been a number of changes and improvements in support for the new Indexing API.

ReversedReadStream

Rather than returning indexing results as a collection, they can be returned as a kind of `Stream`, which avoids the need to create a large result set in memory. This is done via `GsQuery >> readStream`, which streams over the query results in sorted order.

Now, you can also stream over the results in reverse order using the new method `GsQuery >> reversedReadStream`, or by sending `reversed` to the result of `GsQuery >> readStream`.

A new class, `ReversedRangeIndexReadStream`, has been added to implement this feature.

OptionalPathTerm indexes specification moved

`OptionalPathTerms` allow indexes to be built and queries run that ignore the presence of objects that do not implement an instance variable that is specified on a path. For building indexes, this feature is specified using protocol in `GsIndexOptions`.

To set `optionalPathTerms` on a query, previously `GsQuery` methods were used; now, methods on `GsQueryOptions` have been added, and the `GsQuery` methods are legacy.

The following methods have been added:

```
GsQueryOptions >> optionalPathTerms  
GsQueryOptions >> optionalPathTerms:  
GsQueryOptions >> setOptionalPathTerms  
GsQueryOptions >> clearOptionalPathTerms  
GsQueryOptions class >> optionalPathTerms
```

Introductory support for automatic management of selector path indexes

Selector path indexes are indexes on a computed value returned by a unary method selector, rather than the usual index that is directly on an instance variable value. These were introduced in v3.2 with the new indexing API, and have been usable for static data. However, you must manually manage any changes to the values that would affect the indexes.

This has been made somewhat easier in this release. Modification tracking support has been improved, and an example is now included with the distribution. This can be found under `$GEMSTONE/examples/smalltalk/SelectorPathTermExample.gs`.

The following method has been added:

```
Object >> aboutToModifyObject:atOffset:to:
```

Note that this example of automatic modification tracking is not fully tested or supported in this release, but is provided as an example for your own development.

Error handling

The updates to the PetitParser have improved query parsing error handling. Invalid formatting in a GsQuery string now provides useful messages for many kinds of errors.

The following classes have been added:

```
GsQueryParseError  
GsQueryErrorHandlingParser
```

Warning on reduced conflict for identity indexes

GsIndexSpec allows you to specify reduced conflict when creating an identity index. This is meaningless, since IdentityIndexes are by definition reduced conflict; this option was previously quietly ignored. Now, a warning is raised to ensure no other unintentional error exists in your code.

Indexing Stream Classes modified

To fix bug #44079, the class definition for BtreeReadStream has been modified. After upgrade, any persistent instances of this class, and its subclass RangeIndexReadStream, will be invalid.

It is not expected that customer applications create persistent instances of streams on index results. If any exist, they should be recreated after upgrade.

Set valued paths may not begin with asterisk

When creating an query or index, it has been disallowed for the first term on the path to be a set-valued collection. This disallows index paths starting with '*' and 'each.*', depending on the index and query API being used.

CharacterCollection >> asArrayOfPathTerms no longer accepts initial *

The method `CharacterCollection >> asArrayOfPathTerms` is used to partition a String containing the terms of an indexed path. This method no longer accepts an asterisk (*) as the first element of the path.

10. PetitParser changes

The GemStone port of PetitParser has been refactored and updated. The previous version of PetitParser was based on `PetitParser-1r.226`; in version 3.3, it corresponds to `PetitParser-JanKurs.275`.

As well as changes in the existing classes, a number of new PP-prefixed classes have been added.

The following PetitParser related methods have been added to kernel classes:

```
Interval>>asParser  
Collection>>asParser  
Character>>to:
```

`Character>>-` has been removed; it is replaced by `Character>>to:`.

Cached instances require recompile

The upgrade to 3.3 will create new versions of a number of PetitParser classes. The older classes will remain, but will not have methods. Any existing instance of PetitParser classes are **not** automatically upgraded by the GemStone upgrade process.

PetitParser classes are used by the indexing system, but such instances are not persisted.

If you have created persistent instances of any of these classes, they should be rebuilt after upgrade.

Affected classes are:

GsQueryParser	PPMemoizedParser	PPFailure
GsQueryGrammar	PPCompositeParser	PPStream
PPToken	PPMemento	

11. Hot Standby Changes

Add continuousRestoreFromArchiveLogs: variant with delay

The following method has been added:

```
Repository >> continuousRestoreFromArchiveLogs:withDelay:
```

This method is similar to `continuousRestoreFromArchiveLogs:`, with the additional option to specify a delay before the restore. Records are transferred immediately to the standby system but the processing of commits and checkpoints are delayed by the specified number of seconds to allow the system to be stopped to avoid restoring an operation on the database that is not desired.

Methods to get information about Hot Standby sessions

The following methods have been added, to determine if a hot standby logsender or logreceiver are running. Note that since a logsender or logreceiver may be started without passing in the associated master or slave stone, a zero does not prove that the given process is not present.

```
System class >> logreceiverSessionId
```

Return the session ID of the logreceiver process, or zero if no logreceiver is connected to the stone process.

```
System class >> logsenderSessionId
```

Return the session ID of the logsender process, or zero if no logsender is connected to the stone process.

Smalltalk reporting on logsender and logreceiver

A logsender or logreceiver uses a session slot. The return value from invoking `System >> currentSessionNames` now includes these by name. In addition, the results from `System >> descriptionOfSession:` now provides information on a logsender or logreceiver session; the session's `UserProfile` slot is `nil`.

Other changes

The logsender will now detect and shut down if the master stone is in partial logging mode.

12. GsFile changes

Improved performance for server files

Additional primitives and C level code changes improve performance when working with files on the server.

Improvements in handling extended characters

In filenames

Issues related to filenames containing extended characters have been fixed; see “GsFile does not handle file names with extended characters” on page 79.

In code filed out to GsFile

The fileout methods in Behavior have been enhanced so filing out to a GsFile does not require manual encoding or concern about 8-bit file formats.

Now, if you file out source code to an instance of GsFile using methods such as `fileOutClassOn:` or `fileOutMethodOn:`, the output will always be UTF-8 encoded, and the output will include **fileformat utf8**. This allows correct handling of all non-ASCII as well as ASCII text. See “Filing out to GsFile now automatically handles extended characters” on page 40.

GsFile managing file locks

GsFile now has an interface to allow read or write locking a portion of a file. File locking services are provided by the operating system, so behavior is highly dependent on the operating system performing the lock operation. Consult the documentation for your host system to learn about the behavior of file locking for your system.

For processes running on UNIX based systems, file locks are advisory, which means locks are not enforced by the operating system and applications may freely ignore them.

For processes running on Microsoft Windows-based systems, file locks are mandatory, which means locks are enforced by the operating system and applications cannot ignore them. Also note that `getStatusForLockKind: *` is not valid on Windows, as an underlying system function is not available.

On UNIX-based systems, closing any handle referencing a given physical file forces all locks on that file held by the process to be released. For this reason, locks should not be used on any file which the process opens more than once.

Some systems do not support read locks if the file is opened in write-only mode or write locks if the file is open in read-only mode. Attempting to lock files opened in these manners may raise an error indicating a bad file descriptor.

The following methods have been added:

`getStatusForLockKind: kind atOffset: offset forBytes: bytes`

Tests if the receiver could be locked with the given arguments but without actually acquiring the lock. Returns information about a conflicting lock, if any. The `kind` argument must be a symbol that indicates the lock kind requested, and must be either `#readLock` or `#writeLock`. `offset` indicates the position within the receiver in bytes. `bytes` indicates the number of bytes starting at `offset` (0 means to the end of the file).

Returns an Array of 5 elements as described below. If the first element is false, then elements 2 through 5 contain information about a conflicting lock (there may be more than one) which prevented the lock from being available. If the first element is true, then element 2 will be nil and elements 3 through 5 will be -1.

- 1 - a Boolean: true if the lock would have been granted, otherwise false.
- 2 - a Symbol (`#readLock` or `#writeLock`): conflicting lock kind.
- 3 - a SmallInteger: start position of the conflicting lock.
- 4 - a SmallInteger: number of bytes held by the conflicting lock.
- 5 - a SmallInteger: process ID of the owner of the conflicting lock.

This method cannot be used for client files on Windows.

`readLockAtOffset: offset forBytes: bytes`

Requests a read lock on a portion of the receiver without blocking.

`readLockAtOffset: offset forBytes: bytes waitTime: milliseconds`

Requests a read lock on a portion of the receiver and blocks for up to milliseconds to acquire the lock.

`readLockContents`

Requests a read lock on the entire contents the receiver without blocking.

`unlockAtOffset: offset forBytes: bytes`

Releases a previously acquired lock on a portion the file referenced by the receiver. `offset` indicates the position within the receiver in bytes and `bytes` indicates the number of bytes to unlock (0 means unlock the file from `offset` to the end of the file).

`unlockContents`

Releases a previously acquired lock on contents of the receiver .

`writeLockAtOffset: offset forBytes: bytes`

Requests a write lock on a portion of the receiver without blocking.

`writeLockAtOffset: offset forBytes: bytes waitTime: milliseconds`

Requests a write lock on a portion of the receiver and blocks for up to milliseconds to acquire the lock.

`writeLockContents`

Requests a write lock on the entire contents the receiver without blocking.

nextPutAsUtf8 extended

`nextPutAsUtf8`: now accepts a Character as well as a String.

GsFile atEnd now returns true for empty file

When `GsFile` opened an empty file, previously the `atEnd` message returned false. Now, it will return true.

Other added method

`serverCurrentDirectory`

Returns a String, the current directory of this topaz -l or gem process.

13. ProfMonitor new features and improvements

ProfMonitor has been enhanced with the ability to monitor memory-related activities.

In addition the `downTo`: keyword can now specify a percentage by using a `SmallDouble`, as well as an absolute number.

Profile sampling report headers now include a line with memory summary information, for example:

```
0 pageFaults 0 objFaults 0 gcMs 191496 edenBytesUsed
```

New profiling options

ProfMonitor can now profile some memory-related activities, including:

- ▶ Object faults
- ▶ Page faults
- ▶ Time spent in in-memory garbage collection
- ▶ Use of temporary memory eden space

This allows you to correlate memory impact with specific method execution.

To specify these kinds of profiling, use the `ProfMonitor >> setOptions`: method, or a class method with the `options`: keyword, specifying an array with the type of profiling. The options are listed in Table 1.1. The options for type of profiling are mutually exclusive.

For example,

```
ProfMonitorTree
  monitorBlock: [ code to profile ]
  intervalNs: 100000
  options: { #objFaults }
```

When these memory monitoring options are used, in the resulting report, the sampling sections provide the per-method totals (leftmost columns) in units of the tallying option.

Sampling interval CPU or real

The `options`: or `setOption`: argument array also provides the way to specify the sampling frequency interval. Ordinarily, the profiling interval specifies the amount of CPU time between samples. Now, you may instead specify the samples should be taken in the interval in real time. This may be useful in finding performance issues that are not CPU related, such as a sleep.

The default sampling frequency is generally CPU, but for object faults, page faults, and eden space, the sampling frequency is real time.

You can specify the intended sampling interval by including `#real` or `#cpu` in the `options`: or `setOption`: argument array. For example,

```
ProfMonitorTree
  monitorBlock: [ code to profile ]
  intervalNs: 100000
  options: { #objFaults . #cpu }
```

Note that the sampling timing is not limited to memory profiling; you can use this option with ordinary method time profiling.

Object creation profiling

Object creation can now be more easily invoked, using `#objCreation` in the `options:` or `setOption:` argument array. This produces the same result as using `traceObjectCreation:`.

Note the behavior is slightly different from the memory profiling operations controlled by `options:`; rather than producing modified reports, the object creation report is appended to an ordinary method profile.

Table 1.1 ProfMonitor memory profiling options

Type of Profile	Options Keyword	Tally units for sampling reports	Default time profiling
Object faults	<code>#objFaults</code>	faults	<code>#real</code>
Page faults	<code>#pageFaults</code>	faults	<code>#real</code>
Time in in-memory garbage collection	<code>#gcTime</code>	milliseconds	<code>#cpu</code>
Use of temporary memory eden space	<code>#edenUsage</code>	bytes	<code>#real</code>
Object creation	<code>#objCreation</code>	n/a (additional report appended)	<code>#cpu</code>

The following new class methods provide quick profiling access to the new options:

```
ProfMonitor class >> monitorBlock:intervalNs:options:
    Similar to monitorBlock:intervalNs:, but includes the options keyword to specify VM operation profiling.
```

```
ProfMonitor class >> monitorBlock:downTo:intervalNs:options:
    Similar to monitorBlock:intervalNs:, but includes the options keyword to specify VM operation profiling and a percent argument to limit output details.
```

Profiling reports down to a percentage

Profiling methods with the `downTo:` argument allow you to limit the amount of result lines, making the results more readable. This keyword accepted an integer, the lower limit of the number of samples to report. However, if it was not clear before execution how many total samples would be taken, then this was of limited value.

Now, the `downTo:` keyword may be specified as a percentage, by using a `SmallDouble` between 0.01 and 0.99, as an alternative to an absolute number specified by an integer.

For example, to generate a profile report that includes samples for methods taking 10% or more of the total time:

```
ProfMonitorTree
    monitorBlock: [ code to profile ]
    downTo: 0.1
```

14. FFI Changes

C function that accepts an array of callback function addresses

You may now pass callback functions to a C callout as an Array, rather than individually. The following methods have been added to support this:

```
CCallin >> copyBindTo: aBlock
    Return a copy of the receiver which is bound to the specified block. The result
    would typically be used as second argument to CByteArray >>
    ccallinAt:put: when building an Array of function pointers.

CCallout >> bind
    Attempt to resolve the C function described by the receiver, using dlopen() if
    needed, and dlsym().

CByteArray >> ccallinAt: zeroBasedOffset put: aCCallin
    Store the C address for a bound CCallin. Returns receiver.

CByteArray >> ccalloutAt: zeroBasedOffset put: aCCallout
    Store the C address for a CCallout. Returns receiver.
```

Removed methods

The following methods, which were internal to the respective classes, have been removed during code refactoring:

```
CDeclaration >> readAlignedAttribute
CDeclaration >> readDeprecatedAttribute
CDeclaration >> readFormatAttribute
CDeclaration >> readNonNullAttribute
CHeader >> add:for:to:
CHeader >> createFunctionsInClass:library:select:
CHeader >> createInitializerInClass:forFunctions:library:
CHeader >> initializeFunctionsCodeForFunctions:library:
```

GemStone version no longer hardcoded in image

Previously, the GemStone shared library name in image methods including the specific version number. Now, this is computed from the current environment.

15. Other added and changed methods

System Class

New Hidden Set protocol

To iterate a hidden set without converting to an in-memory collection, the following method has been added.

```
System class >> hiddenSet: hiddenSetSpecifier do: aBlock
  Executes the one argument block aBlock for each argument in the specified hidden set.
  Does a destructive enumerate of the hidden set. Should only be used with public
  hidden sets 41-45. Returns the number of objects enumerated.
```

To check if two hidden sets are the same, without performing any other work, the following method has been added.

```
System class >> compareHiddenSet: firstSetSpecifier to: secondSetSpecifier
  Compares two hidden sets to determine if they contain the same objects. Returns true
  if the hidden set contents are identical. Otherwise returns false. Neither hidden set is
  modified.
```

Change in return value range from timeNs

The method `timeNs` has been documented to return a `SmallInteger`, but actually has returned a `LargeInteger`. As of version 3.3, the highest bits are discarded to ensure the result is a `SmallInteger`.

Given the `SmallInteger` range and nanosecond resolution, this creates a result range of over 36 years. However, values returned from earlier versions will be larger than values returned in v3.3 and later. For example, at a particular specific time, `timeNs` could return 1454362338115845960 in v3.2.12, and 301440833508998984 in v3.3.

Improved clock resolution, new profiling method

The millisecond resolution of the profiling methods is insufficient for modern systems. The clock operations have been updated to microsecond resolution, and the following methods have been added. Note that the clock resolution is dependent on the operating system

```
System class >> microsecondsToRun: aBlock
  Returns the number of CPU microseconds used while evaluating aBlock. The
  argument aBlock must be a zero-argument block.
```

```
System class >> readClockNano
  Returns an Integer indicating the amount of CPU time used by the current process, in
  units of nanoseconds. The resolution of the result is operating system dependent.
```

Object >> subclassResponsibility

The method `Object >> subclassResponsibility` has been added, for compatibility with other Smalltalk dialects. This method looks up the selector on the stack to preserve the Gemstone approach.

System gemVersionReport and gemVersionAt: include buildType

The methods `System class >> gemVersionReport` now includes the key 'gsBuildType', and this is understood by `gemVersionAt:`. Possible build types are FAST, SLOW, PROFILE and NOOP. Normal customer distributions and installations will be FAST.

Convenience methods to get commonly needed statistics

The following public methods have been added to fetch useful information about the repository, and the equivalent private methods have been removed.

`System class >> commitRecordBacklog`

Return the number of commit records in the system. Replaces `System class >> _commitRecordBacklog`, which has been removed.

`System class >> deadNotReclaimedCount`

Return the number of dead not reclaimed objects in the system. Replaces `System class >> _deadNotReclaimedCount`, which has been removed.

`System class >> possibleDeadSize`

Return the number of possible dead objects in the system. Replaces `System class >> _possibleDeadSize`, which has been removed.

`System class >> possibleDeadSymbols`

Return the number of possible dead Symbols in the system. Replaces `System class >> _possibleDeadSymbols`, which has been removed.

`System class >> reclaimedSymbols`

Return the number of Symbols reclaimed since stone was started. Replaces `System class >> _reclaimedSymbols`, which has been removed.

`System class >> remoteSharedCacheCount`

Return the number of remote shared page caches on the system. Replaces `System class >> _remoteSharedCacheCount`, which has been removed.

`System class >> scavengablePagesCount`

Return the number of scavengable pages in the system. Replaces `System class >> _scavengablePagesCount`, which has been removed.

`System class >> shadowPagesCount`

Return the number of shadow pages in the system, same as `scavengablePagesCount`.

`System class >> totalSessionsCount`

Return the number of sessions logged into the system. Replaces `System class >> _totalSessionsCount`, which has been removed.

`System class >> userSessionsCount`

Return the number of sessions logged into the system.

Collection classes

Added methods

`SequenciableCollection >> second`

This method returns the second element of the collection.

`PositionableStreamPortable >> peekTwice`

Answer an array containing the two elements that would be returned if the message next were sent to the receiver twice. If the receiver is at or reaches the end, the array will include one or two nils.

This method is not in `PortableStreamLegacy`. It is available for `PositionableStreams` only if `PortableStreams` are installed in your system.

removeAllSuchThat: now returns removed elements

The method `SequenceableCollection >> removeAllSuchThat:` previously returned the receiver; now, it returns the collection of elements that were removed.

Class information and Fileout changes

Default class comment now reports hierarchy

For classes that did not have a class comment defined, previously the comment method reported an empty String. Now, text similar to the following is produced:

```
No class-specific documentation for <myClass>, hierarchy is:
Object
  <myClass>( ivar1 ivar2 ...
```

If no comment is defined, the fileout form will not include the comment. This avoids filein explicitly defining the default comment.

Fileouts now include expectvalue /Class

When filing out a class, the fileout form now includes a line `/Class` before the Class definition and before the Class comment. This avoids problems with topaz error checking.

Filing out to GsFile now automatically handles extended characters

When filing out source code that included extended characters to a `GsFile`, previously it was necessary to first encode the fileout using `encodeAsUTF8`, and manually include the line **fileformat UTF8** to ensure filein would be correct. Now, when using the Behavior methods that file out to a stream, such as `fileOutClassOn:`, `fileOutMethodOn:`, etc., and using an instance of `GsFile` as the stream argument, the output will automatically be encoded as UTF-8 and the **fileformat UTF8** automatically included.

Fileout using these same methods to instances of `Streams` does not apply these changes.

Class definition for old class versions now provides SymbolDictionary

Previously, the definition (a class creation expression) produced by `Class >> definition`, when sent to a version of a class other than the current version, provided "(class not in your dictionaries)" as the `inDictionary:` argument. Now, if the given class is itself not in a `SymbolDictionary`, it provides the `SymbolDictionary` of the most recent version in its `classHistory`.

Added method instancesInMemory

The method `Class >> instancesInMemory` has been added, which returns instances of the class that are in temporary object memory.

Socket classes

`GsSocket class >> fromFileHandle: aSmallInteger`

Create a non-blocking instance of the receiver which will use the specified file handle. The fileHandle must be the file descriptor of an open socket that was inherited from the parent when the gem or topaz -l process was forked, or a socket created by non-Gemstone C code. in this gem or topaz -l process. The non-Gemstone C code may be C application code that calls the GCI or C code called from a user action or through FFI.

The address family of the socket is determined by the process doing the fork when it accepts a connection, or by the non-Gemstone C code. The socket will be set to non-blocking and close-on-exec by this method.

`GsSecureSocket class >> sslLibraryVersionString`

Answer an instance of String which describes the version of OpenSSL library loaded by this session.

The following methods were added to GsSecureSocket, overriding the inherited version from GsSocket which was not correct for GsSecureSocket.

`GsSecureSocket >> lastErrorString`

`GsSecureSocket class >> lastErrorString`

`GsSecureSocket >> acceptTimeoutMs:`

Date and DateTime changes

Date class creation methods now error on out-of-range day

The Date class creation methods that accept a day of month input now will raise an ArgumentError if the day specified is larger than the number of days in the given month. Previously, the date wrapped to the following month.

Affected methods on Date class include `newDay:month:year:`, `newDay:monthNumber:year:`, `fromString:*`, and `fromStream:*`.

Date today now respects TimeZone

The method `Date Class >> today` previously returned the OS date, regardless of a TimeZone set in the repository. Now, Date today invokes DateAndTime to compute the Date in the repository. For the old behavior, see `Date Class >> _today`.

DateTime creation performance improvement

Creating an instance of DateTime invokes code to check the TimeZone for daylight savings time. The objects that support these TimeZone operations are now cached for better performance.

In addition, the method `TimeZone >> yearStartDst` will return `SmallInteger maximumValue` rather than raising an error if there is no DST set.

ClassOrganizer changes

ClassOrganizer references to literals

The following method has been added, allowing you to list all references to a literal value (such as a Symbol, String, or numeric):

```
ClassOrganizer >> literalsReport: aLiteral
    Returns a String describing the methods whose source contains a literal reference to the
    specified literal. The argument may be any object that is legal as a literal, including a
    String, Symbol, Number, Array, Boolean, nil.
```

sendersOf:* and optimized selectors

Previously, optimized selectors were not returned by `ClassOrganizer >> sendersOf:` or `sendersOf:in:.` Now, senders of optimized selectors are found.

However, due to the optimization, the index into the source code of the first use of an optimized selector within a method is not available.

Repository

Repository createExtent:withMaxSize: minimum is 16MB

Extent sizes are now always in multiples of 16MB. This method will now raise an error if the size argument is less than 16. If it is greater than 16MB and not a multiple of 16MB, the multiple of 16MB that is lower than the argument is used as the maximum size.

This is also true of the deprecated variant
`createExtent:withMaxSize:startNewReclaimGem:.`

findReferencePath excludes internal implementation instances

The results of methods such `findReferencePathToObject:` produces a list of object representing the objects chain of references. Previously, this included objects such as `LargeObjectNodes` that are internal implementation details that cannot be accessed normally. Now, these are excluded from the result set, though they are still visible in the scan details.

GsExternalSession API for host login

Methods have been added to `GsExternalSession` to allow the host login information to be set in the parameters. The following instance methods have been added:

```
hostUsername:
hostPassword:
```

An instance creation method that includes host login information has been added:

```
GsExternalSession >> gemNRS:stoneNRS:username:password:
hostUsername:hostPassword:
```

An instance method to return the username has also been added,
`GsExteranalSession>>username`

GsProcess stack report that includes method source code

A method has been added so programmatically generated stack reports can include the source code of the method:

```
GsProcess >> stackReportToLevel:withArgsAndTemps:  
andMethods:includeSource:
```

New Features

16. Transient Methods

A new feature has been added to allow non-persistent override of existing methods for a single session, such that the transient methods are not affected (neither lost nor made persistent) by commit or abort.

To define transient methods, the method must already exist for the given class, and you must have write permission for the object security policy of the methods's class.

Invoke the following method to compile a transient method:

```
Behavior >> compileTransientMethod: dictionaries: environmentId:
    This method is similar to compile:dictionaries:environmentId:, except that
    the compiled method is installed into the transient method dictionary for the receiver.
```

The transient methods disappear on logout. You may remove them during the life of the session by invoking one of the following:

```
Behavior >> removeTransientSelector: environmentId:
    Remove the method with the selector argument from the transient method dictionary
    for the receiver.
```

```
Behavior >> removeTransientSelector: environmentId: ifAbsent:
    Remove the method with the selector argument from the transient method dictionary
    for the receiver. If the given selector is not present, execute the result of evaluating the
    zero-argument ifAbsent: block.
```

17. Object Canonicalization Framework

For objects that have a finite set of possible values, and which are used frequently within an application, it can be very useful to reduce the number of duplicate instances. One example is GemStone DateTime instances.

The Object Canonicalization Framework provides an way to reduce these duplicate instances in your application. It allows you to register a class and use this for object creation.

Added classes supporting this framework are:

```
CanonicalObjectManager
CanonicalObjectPolicy
CanonicalObjectRegistry
AbstractReferencingObjectPolicy
ReferencingObjectPolicy
```

This framework does not guarantee that instances of the given class are entirely canonical. If multiple users create instances at the same time, duplicate instances are created (this is intentional, to avoid commit conflicts); and it is not restricted from creating instances that do not use the framework. Comparisons of instances of the given class should use = rather than == .

To create canonical instances, first register a class as being canonicalizable, using `CanonicalObjectManager>>createRegistryFor:..`. Then, to request a canonical instance of that class, use `canonicalObjectFor:..`

For example, to create canonical instances of `DateTime`,

```
CanonicalObjectManager default createRegistryFor: DateTime.  
CanonicalObjectManager default canonicalObjectFor:  
    (DateTime fromString: '01/01/2015 00:00:00')
```

As mentioned, to avoid commit conflicts, if a canonical object is not yet in the registry, the framework allows overlapping transactions to see different objects as the canonical one. This means that two overlapping transactions that create an equivalent object will result in two equivalent instances. Subsequent transactions will see one of the earlier ones. To ensure that only one object is used, send `ultimateCanonicalObjectFor:` and be prepared to handle commit conflicts.

An alternate way to canonicalize objects is to provide a set of objects to the manager and have it scan them for references to canonicalizable objects, using the method `CanonicalObjectManager >> cleanupAll:..` This would typically be used with objects found using a repository scan, such as the results of `Repository>>listReferencesToInstancesOfClasses:toDirectory:..` Using this, you can canonicalize existing objects in your application.

Because some objects should not have references arbitrarily replaced (e.g., `IdentitySets`), the framework uses a pluggable policy to decide which referencing objects can be modified. The default `ReferencingObjectPolicy` bases the decision on the referencing object class.

To fully implement a transparent canonicalization scheme, applications may want to use `CanonicalObjectManager>>cleanupWriteSet` to scan the write set for the current transaction and replaces references as indicated by the policy. A properly configured manager could do appropriate cleanup before each commit.

By default, the default instance of `CanonicalObjectManager` has world write permission so that canonical objects can be created by any session and shared. You can modify this using `CanonicalObjectManager class>>#securityPolicy:..`

18. Support for LDAP authentication without anonymous binds

Previously, LDAP authentication relied on anonymous binds to the LDAP server, prior to authentication. It is now supported to make authenticated binds to an LDAP server, to support installations that disallow anonymous binds.

To store the information required to make an authenticated bind to LDAP, a new class has been added, `LdapDirectoryServer`. For each LDAP server, an instance of this class is created containing the authentication information, and the instances are stored in a class variable, `AllLdapDirectoryServers`, which has its own reserved OOP for performance.

To create an `LdapDirectoryServer`, and add it to the global list, use the following method:

```
LdapDirectoryServer class >> newWithUri: uri bindDN: aBindDn  
    password: password
```

If LDAP authentication is setup, using

```
enableLDAPAuthenticationWithAlias: aString baseDn: baseDn
filterDn: filterDn
```

then either anonymous or authenticated binds are performed to authenticate the user password.

- ▶ If there are existing instances of `LdapDirectoryServer` in `AllLdapDirectoryServers`, then these are used, in order; normally this would include a primary LDAP server and an alternate. If and only if the bind to the first LDAP server fails, the second is tried, and so on. If no binds are successful, then the login will fail.
- ▶ If `AllLdapDirectoryServers` is empty - if no `LdapDirectoryServer` have been added - then an anonymous bind is performed, as in previous versions of GemStone.

For example, to set up a user to use LDAP authentication, and setup a single LDAP server to authenticate binds, the following code would be executed.

```
aUserProfile
  enableLDAPAuthenticationWithAlias: nil
  baseDn: 'ou=Users,dc=gemtalksystems,dc=com'
  filterDn: '(uid=%s)' .

LdapDirectoryServer
  newWithUri: 'ldaps://ldap.gemtalksystems.com'
  bindDN: 'uid=bindUser,ou=Users,dc=gemtalksystems,dc=com'
  password: 'swordfish' .
```

Users are individually configured to use LDAP authentication, but LDAP directories are configured for the repository as a whole, so this step is only done once. All users use the same bind authentication, though they use their own individual passwords to perform the actual account authentication.

Note that the instances of `LdapDirectoryServer` store the encrypted bind password. This is not the actual user password to login to GemStone; this bind password allows you to perform the user password authentication.

To determine if a particular `LdapDirectoryServer` exists, use the following method, which returns either the `LdapDirectoryServer` or `nil` if it is not found:

```
LdapDirectoryServer >> findServerWithUri: aUriString
```

to remove an `LdapDirectoryServer`, use the method:

```
LdapDirectoryServer >> removeServerWithUri: aUriString
```

19. Interface to write to syslog

Syslog is an OS facility to log messages from a variety of software, using a standard API. GemStone now includes an interface, allowing GemStone processes to send messages to this log, on the stone's and/or gem's host. The syslog log file is normally located at `/var/log/syslog`, and is only readable by root.

Support is provided by a new class, `GsSysLog`. Logging is provided by class methods, and this class is not expected to be instantiated.

The core functionality is provided by the following method:

```
GsSysLog class >> writeLogMessage: messageString
    prefix: prefixString
    priority: prioritySymbol
    facility: facilitySymbol
    flags: flagsInt
```

This method writes a message to the system log by calling UNIX functions `openlog()`, `syslog()` and `closelog()`.

messageString must be an instance of `String` or a instance of a subclass of `String`. It also must have a size between 1 and 4905 characters. Note the host system may have further restrictions, consult operating system documentation for more information.

prefixString must be an instance of `String`, an instance of a subclass of `String`, or `nil`. If it is not `nil`, then it also must have a size between 1 and 1023 characters. A value of `nil` causes the default prefix of 'GemStone' to be used for the prefix. Note the host system may have further restrictions, consult operating system documentation for more information.

prioritySymbol indicates the priority of the message, one of the following:

```
#LOG_EMERG   system is unusable
#LOG_ALERT   action must be taken immediately
#LOG_CRIT    critical conditions
#LOG_ERR     error conditions
#LOG_WARNING warning conditions
#LOG_NOTICE  normal, but significant, condition
#LOG_INFO    informational message
#LOG_DEBUG   debug-level message
```

facilitySymbol indicates what type of program is logging the message (called the facility) and must be one of the following:

```
#LOG_USER    generic user-level messages
#LOG_LOCAL0  reserved for local use
#LOG_LOCAL1  reserved for local use
#LOG_LOCAL2  reserved for local use
#LOG_LOCAL3  reserved for local use
#LOG_LOCAL4  reserved for local use
#LOG_LOCAL5  reserved for local use
#LOG_LOCAL6  reserved for local use
#LOG_LOCAL7  reserved for local use
```

flagsInt is a SmallInteger which is either 0 or the bitwise-OR of any of the following values:

- 1 - Write the message to stderr of this process as well as the system logger.
- 2 - Do not call `closeLog()` after writing the message.
- 4 - Write directly to system console if there is an error while sending to system logger (LOG_CONS option).
- 8 - Include the PID with each message (LOG_PID option).
- 16 - Write the message to the system logger on the stone's host. Has no effect unless the gem is running on a host remote from the stone.
- 32 - Write the message on the system logger gem's host. Has no effect unless the gem is running on a host remote from the stone.

For sessions that are not on same host as the stone, at least one of the bits corresponding to the values 16 and 32 must be set.

Signals an error if the privilege `NoPerformOnServer` is true.

The following shortcut class methods are provided:

`writeInfoMessage: messageString`
Writes an informational message to the syslog using default settings. This is priority LOG_INFO, to the facility LOG_USER, echoing to stderr and the console. This is restricted to gems on the same host as the stone.

`writeWarningMessage: messageString`
Writes a warning message to the syslog using default settings. This is priority LOG_WARNING, to the facility LOG_USER, echoing to stderr and the console. This is restricted to gems on the same host as the stone.

`writeErrorMessage: messageString`
Writes an error message to the syslog using default settings. This is priority LOG_ERR, to the facility LOG_USER, echoing to stderr and the console. This is restricted to gems on the same host as the stone.

The following methods provide convenient ORing of flags, and other flag information:

`setFlagForStderr: anInt`
ORs *anInt* with the bit indicating enable echo to stderr.

`setFlagNoCloseLog: anInt`
ORs *anInt* with the bit indicating do not close the log.

`setFlagLogToConsoleOnError: anInt`
ORs *anInt* with the bit indicating log to console if there is an error writing to syslog.

`setFlagIncludePid: anInt`
ORs *anInt* with the bit indicating include PID in the log message.

`setFlagLogOnStoneHost: anInt`
ORs *anInt* with the bit indicating log to stone host.

`setFlagLogOnGemHost: anInt`
ORs *anInt* with the bit indicating log to gem host.

`allFlagBits`
Returns Integer with all flag bits.

defaultFlags

Returns an Integer with the default bits, which are setFlagLogToConsoleOnError and setFlagIncludePid.

For example, a remote gem could write a message to the syslog on the remote host using the following:

```
GsSysLog writeLogMessage: 'my message'  
  prefix: nil  
  priority: #LOG_ERR  
  facility: #LOG_USER  
  flags: (GsSysLog setFlagLogOnGemHost: GsSysLog defaultFlags)
```

Utility and Environment Changes

20. Changes in Statmonitor and Cache Statistics

Millisecond monitoring

Previously, statmonitor recorded the time as an Integer value, and if sampling rates smaller than 1 second were requested, these were recorded in incremented units rather than real time to accommodate VSD.

In VSD v4.0, the ability to read millisecond timestamps was added. The format is *s.ms* where *s* is seconds from January 1, 1970 and *ms* is the milliseconds within that second.

With v3.3, when statmonitor is specified with samples intervals of less than 1 second, timestamps are recorded using millisecond timestamps. VSD v4.0 and later will correctly graph and label these timestamps.

When sub-second time stamps are requested, statmonitor will initiate collecting stats on an rounded interval, to allow easier statistics reading. So for example, if 10 ms statistics are requested (-I10) and the actual system time milliseconds is .734, statmonitor will delay the collecting the first statistic until the system time milliseconds is .740.

With recording real times for millisecond timestamps, there are some cases where samples may be skipped. The actual sample time is deemed "close enough" to the scheduled sample time if the actual sample time is within +/- 10% of the scheduled sample time. Samples are skipped only if this difference exceeds 10% of the sample interval. To allow tracking, statmonitor has two additional statistics:

SamplesSkipped (Statmonitor)

Number of times statmonitor did not record statistics as a scheduled point in time.

SystemClockStuckCount (Statmonitor)

Number of time the system returned the same timestamp consecutively.

VSD version

Statmonitor in version 3.3 records millisecond timestamps when intervals of less than a second are used; this is a change in the statmonitor file format. You must use VSD version 4.0 or later to read statmonitor data files with subsecond timestamps generated by statmonitor in version 3.3. VSD v4.0 and later can read statmonitor files produced by older versions of GemStone/S 64 Bit and 32-bit GemStone/S.

64-bit statistics

Previously, some statistics values could roll past the 32-bit unsigned integer range, which caused them to roll over to invalid values.

For statistics in which this occurred, in GemStone and for host statistics, the code now records 64-bit values. In VSD v4.0.2 and later, recorded statistics values that are outside the 32-bit range are automatically promoted to 64-bit, and will be displayed correctly.

Statmonitor automatic startup for Stones and remote caches

Configuration options have been added to allow you to specify that statmonitor should always be run when the stone is running. Configuration options allow monitoring Stone, remote shared page cache, and mid level shared page cache. These configuration options are:

STN_STATMONITOR_ARGS (page 58)

GEM_STATMONITOR_ARGS (page 56)

GEM_STATMONITOR_MID_CACHE_ARGS (page 57)

These options are enabled by setting the configuration parameter to a valid set of statmonitor arguments, not including the name of the stone or cache. For example,

```
STN_STATMONITOR_ARGS = "-i5 -r -h24";
```

To allow the statmonitor output file location and names to be more easily managed for these automatically generated statmonitor data files, new options have been added to statmonitor, allowing much more control over output file names. The new options are described in the following sections.

If statmonitor fails to start, it does not prevent the stone or remote session from starting up.

File patterns specification with new add -F option

Previously, you could specify a filename explicitly, or allow the system to produce a default name based on the process PID. Now, flexible file names can be specified.

The new -F option specifies a file pattern to use. This may include a directory specification, strings and characters, and variables that are resolved when the filename is generated.

There are two kinds of variables that can be used.

- ▶ Invocation-static variables include data from the specific invocation of statmonitor. These variables start with %% characters; the following are accepted:

- %%C - name of the shared page cache

- %%H - name of the host

- %%i - sample interval in seconds

- %%I - sample interval in milliseconds

- %%P - process ID of statmonitor

- %%S - name of the stone

- ▶ Timestamp variables starting with a single % which have the meaning described in the strftime(3) function call. For example, %d-%m-%y-%H:%M:%S provides the numeric day and month, and two-digit year, and hours, minutes

So for example, to define a file with stone name, host name, and the current date and time:

```
statmonitor -F '%%S_%%H_%d-%m-%y_%H:%M:%S' gs64stone
```

Change in behavior when existing filename specified

Previously, attempting to write to a specific filename would error if a file with that name already existed. Now, if *filename* already exists, *filename-n* is created instead, where *n* is the lowest number for which no file with that name already exists.

So, for example, if `statmon_gs64stone.out` exists, `statmon_gs64stone-1.out` is created, and if both these files exist, `statmon_gs64stone-2.out` is created.

Change in filename numbering of first file for -r restart filenames

Using the -r option with statmonitor, along with -t or -h, allows a new statmonitor file to be started at regular intervals. The file names of these files have the same base, and are suffixed with a number for uniqueness.

Previously, the file names were *filename.out*, *filename-2.out*, *filename-3.out*...

Now, the file names are *filename-1.out*, *filename-2.out* out, *filename-3.out*...

Directory specification; add -d

The new -d option allows you to specify a directory in which to place statmonitor files. This only applies when statmonitor determines the filename, and cannot be used with the -F or -f options. To specify a directory with the -f or -F option, include the desired directory name within the specified file pattern.

Filenames on automatic restart; add -R

The option -R has been added in this release, similar to the existing -r feature.

The existing -r option (along with -h or -t) causes statmonitor to be restarted automatically, using the same filename for each restart. With -R, the filename is recomputed for each restart. This is useful when using the -F option for filenames.

When the -F pattern does not include any timestamp variables, it will be the same for each new filename generated. These files will each be numbered, starting with -1 (the same file numbering as with the -r option).

However, if the -F pattern includes a timestamp variable, then the resulting filenames may or may not be unique. It is most useful if the timestamp pattern specified includes units that are smaller than the -h; -t restart frequency, for example, using the day, month and year with a 24-hour restart would mean each filename is unique. To optimize file naming for files that include timestamps, the first file of a sequence does not have the "-n" suffix. If the filename is not unique, on restart, the second file will have a -1 suffix.

Changes related to multi-threaded page servers and remote page servers

Multi-threaded page servers have been added in v3.3; each gem on a remote cache shared the same page server process on the Stone's host. Statmonitor will record these as multiple entries with the same ProcessId, with each thread named "PgsvrClientThdN", where N is the cache slot; previously these were named "PgsvrN".

In addition, a new method has been added to allow you to set the name of the page server in Statmonitor, similar to setting the name of the Gem.

```
System >> pageServerCacheName: aString
```

Sets the cache name for the page server thread serving the current session. Also sets the cache name for the page server on the mid-level cache, if it exists.

Sampling persistent shared counters; add -B

The added -B option allows sampling of persistent shared counters. Up to 1536 persistent shared counters can be samples.

Added -G option

The added -G option specified to sample only the stone, shared cache monitor, AdminGem and ReclaimGem.

-h option

Statmonitor now understands the -h option (with no argument) and reports help; the existing option for -h <number> (with an argument) works as in previous releases.

Added statistics

The following new statistics are collected in v3.3:

CompressionCount (Pgsvr)

Number of times the compression library was called to compress data.

CompressionKbIn (Pgsvr)

Total size of data compressed before compression, expressed in kilobytes.

CompressionKbOut (Pgsvr)

Total size of data compressed after compression, expressed in kilobytes.

CompressionTimeCpu (Pgsvr)

CPU time in milliseconds spent performing data compression.

CompressionTimeReal (Pgsvr)

The real time in milliseconds spent performing data compression.

DecompressionCount (Pgsvr)

Number of times the compression library was called to decompress data.

DecompressionKbIn (Pgsvr)

Total size of data decompressed before decompression, expressed in kilobytes.

DecompressionKbOut (Pgsvr)

Total size of data decompressed after decompression, expressed in kilobytes.

DecompressionTimeCpu (Pgsvr)

CPU time in milliseconds spent performing data decompression.

DecompressionTimeReal (Pgsvr)

The real time in milliseconds spent performing data decompression.

ExtentGrowTimeMs (Stone)

Total amount of real time in milliseconds the stone has spent increasing the size of the repository by growing the extent (s).

ExtentGrowTimePerGrow (Stone)

Average amount of real time in microseconds it takes the stone to perform a single grow operation on an extent.

ExtentGrowTotal (Stone)

Number of times the stone has increased the size of the repository by growing the extent(s).

FreePagesPoolSize (Stone)

The number of pages in the freePool that the Stone maintains for its own use.

LogSenderBlockId (Stone)

The record ID of the last transaction log record read by the log sender process.

LogSenderId (Stone)

The file ID of the last transaction log read by the log sender process.

RecoverTimeLag (Stone)

The difference in commit time (`restoreLogCommitTime - originalCommitTime`), gives a relative idea of how hot a hot standby is.

SamplesSkipped (Statmon)

Number of times statmonitor did not record statistics as a scheduled point in time.

SystemClockStuckCount (Stone)

Number of time the system returned the same timestamp consecutively.

TotalObjectsCommitted (Stone)

Total number of new objects committed by all gems.

UserSessionsCount (Stone)

The total number of user sessions currently logged in to the system.

Renamed Statistics

The Gem statistic **WorkingSetInvalidateAllCount** has been renamed to **WorkingSetClearedCount**.

The Linux-only statistic **CommittedAsKB** has been renamed to **CommittedASKB**, to avoid the misleading name; this is the Linux system stat `Committed_AS`.

The Solaris-only statistic **Recoverable** has been renamed to **RecoverableErrors** for clarity.

the 32-bit statistic **TotalNewObjsCommitted** is replaced by a 64-bit statistic **TotalNewObjectsCommitted**.

Removed statistics

The following statistics have been obsolete, but were still recorded. They have been removed in this release:

- AsyncWritesInProgress
- BackupRecordPagesWrittenByGem
- BackupRecordPagesWrittenByStone
- CompressedLogPagesWrittenByGem
- CompressedLogPagesWrittenByStone
- CountMapLeafPagesWrittenByGem
- CountMapLeafPagesWrittenByStone
- DeadDataPagesWrittenByGem
- DeadDataPagesWrittenByStone
- EmptyBmLeafPagesWrittenByGem
- EmptyBmLeafPagesWrittenByStone
- LogRecordPagesWrittenByGem
- LogRecordPagesWrittenByStone
- LostOtPagesWrittenByGem
- LostOtPagesWrittenByStone
- MultObjPagesWrittenByGem
- MultObjPagesWrittenByStone
- RepBkupRestPagesWrittenByGem
- RepBkupRestPagesWrittenByStone

21. Changes in Configuration Parameters

Extent sizes always multiple of 16MB

The minimum extent size is now 16MB, and the size must be a multiple of 16MB.

This affects `DBF_EXTENT_SIZES` and `DBF_PRE_GROW`. Entries must be at least 16MB, and should be a multiple of 16MB. If your configuration file specifies a value that is not a multiple of 16MB, the next lowest multiple will be used.

For raw partitions, the `DBF_EXTENT_SIZES` entry should now be 16MB less than the size of the partition for optimal performance.

Additional option to manage commit record backlog

When a commit record backlog occurs, the oldest session is sent a `sigAbort`; after this session updates, then the next-oldest session becomes the oldest and will in turn get a `sigAbort`. When many sessions may be holding onto old commit records, and if the sessions timeout rather than respond immediately to the `sigAbort`, this can make resolving commit record backlogs difficult.

A configuration parameter has been added, `STN_SIGNAL_ABORT_AGGRESSIVE`, that instructs the system to send a `sigAbort` to a configurable number of sessions that would be eligible for `sigAbort`; that is, the sessions reference the oldest commit records that are older than `STN_SIGNAL_ABORT_CR_BACKLOG`. This allows a number of sessions to be sent `sigAbort` at once. This setting is off by default.

See “`STN_SIGNAL_ABORT_AGGRESSIVE`” on page 58.

Additional options for handing configuration parameters

In v3.3, there are additional features that provide flexibility in managing configuration options.

- ▶ The `INCLUDE` directive parameter has been added. This allows configuration files to be nested. It can be used multiple times to include multiple configuration files, and these can in turn nest further files. Headers in the log files include information on the order in which included configuration files are read. See “`INCLUDE`” on page 57.
- ▶ Topaz has a new `-C` option that allows you to pass in the actual configuration parameters settings on the command line, overriding settings in the configuration files specified by `-e` or `-z` options. This applies to linked sessions only. For more details, see “`Add -C option to pass in arbitrary configuration parameter values`” on page 64.
- ▶ The Gem process now accepts additional parameters, `-e execonf` and `-C confParamsText`, which may be provided via the NRS used when invoking the `gemnetobject` or other scripts.

For example, with topaz:

```
set gemnetid '!@santiam.gemtalksystems.com!gemnetobject -e
    $GEMSTONE/my.conf '
set gemnetid '!@santiam.gemtalksystems.com!gemnetobject -C
    GEM_TEMPOBJ_OOMSTATS_CSV=TRUE '
```

Runtime changes no longer write to system configuration file

Updates to the runtime configuration parameters #StnEpochGcEnabled or #StnSymbolGcEnabled previous added settings for STN_EPOCH_GC_ENABLED and STN_SYMBOL_GC_ENABLED to the system configuration. This was inconsistent with other runtime configuration parameters, which apply only for the running system. Changes are no longer appended to the system configuration file; to make persistent changes, you must manually edit the configuration file.

Use of units in configuration values

In v3.2, for configuration parameters that controls file and memory sizes, a feature was added to allow you to specify units of KB, MB, and GB, as well as omitting units to use the default size.

To avoid confusion and ambiguity, the default, maximum, and minimum values described in the configuration file now always include units, and the values printed in the Stone, Gem, and other logs now include the units.

Configuration values without units are still understood, as the default units of that parameter; it is not required that you update your configuration files.

Added configuration parameter features

The following configuration parameters have been added:

Encrypted connection between remote gem and page server

GEM_PGSVR_USE_SSL

Controls whether a remote gem uses a secure socket layer (SSL) connection to converse with its page server(s), both the page server on the stone's host and the page server on the mid-level cache, if any. This option has no effect local gems (i.e., gems running on the same host the stone process.).

The SSL tunnel is established in anonymous mode, meaning communications are encrypted but the sender and receiver are not authenticated.

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

When enabling this option, consider also enabling the GEM_PGSVR_COMPRESS_PAGE_TRANSFERS option. SSL encodes no more than 16 KB into a single packet and some messages sent between the gem and page server may exceed this limit, causing SSL to send multiple encrypted packets to convey a single message.

Default: FALSE

Automatically start statmonitor for remote cache

GEM_STATMONITOR_ARGS

List of arguments to be passed to the statmonitor executable. If the list is not an empty string, statmonitor will be started automatically by a remote gem which creates a remote shared page cache; this setting is ignored other than by remote gems that create a remote shared page cache.

Example: start statmonitor with a 5 second sample interval with a date and timestamp included in the output file name and place the output file in the directory /statmonFiles:
GEM_STATMONITOR_ARGS = "-i5 -u5 -d /statmonFiles -F'statmon_%%S_%%P_%%d-%%m-%%y-%%H:%%M:%%S'";

Automatically start statmonitor on mid-level cache

GEM_STATMONITOR_MID_CACHE_ARGS

List of arguments to be passed to the statmonitor executable. If the list is not an empty string, statmonitor will be started automatically by the remote gem which creates a mid-level shared page cache; this setting is ignored other than by remote gems that create a mid-level shared page cache.

Example: start statmonitor with a 5 second sample interval with a date and timestamp included in the output file name and place the output file in the directory /statmonFiles:
GEM_STATMONITOR_MID_CACHE_ARGS = "-i5 -u5 -d /statmonFiles -F'statmon_%%S_%%P_%%d-%%m-%%y-%%H:%%M:%%S'";

Limit number of unsuccessful mark-sweeps

GEM_TEMPOBJ_CONSECUTIVE_MARKSWEEP_LIMIT

Controls in-memory garbage collection; if there are more consecutive in-memory mark sweeps than this value, without any intervening successful in-memory scavenges, then an OutOfMemory error will occur.

Runtime equivalent: #GemTempObjConsecutiveMarkswEEP_Limit
Default: 50
Min: 20
Max: 5000

Create CSV file with memory statistics on out-of-memory error

GEM_TEMPOBJ_OOMSTATS_CSV

If TRUE, then when an Error 4067/OutOfMemory occurs, the detailed stats of instances of classes in temporary memory are written to a `gemnetobjectpid.csv` file in CSV format. For a topaz -l process, CSV data is written to a `topazpid.csv` file in the current directory.

The configuration parameter GEM_TEMPOBJ_OOMSTATS_CSV and the environment variable GS_DEBUG_VMGC_VERBOSE_OUTOFMEM operate independently. The environment variable GS_DEBUG_VMGC_VERBOSE_OUTOFMEM controls writing statistics in standard text to the gem log or stdout.

Runtime equivalent: #GemTempObjOomstatsCsv
Default: FALSE

Add ability to nest configuration files

INCLUDE

Specifies the name of a file containing configuration parameters, that should be included in this configuration file. The value must be a single path to a file, and may include environment variables which will be expanded. INCLUDE directives may be nested up to 100 deep.

Unlike other configuration parameters, the INCLUDE directive parameter may be present more than once in a configuration file, and each file specified by an INCLUDE line, not just the last one, is read. When other configuration parameters settings are found in more than one file, however, the last one read is the value that is used.

For example:

```
INCLUDE=config2.conf;    # will look in current directory
INCLUDE="$GEMSTONE/data/config3.conf";
```

Define port range for pageservers for remote gems

STN_PGSVR_PORT_RANGE

A list of two integers specifying the minimum and maximum port range to be used for listening sockets in pgsvr processes on the stone machine. Such pgsvr processes will listen on the first available port from this range, using the address that was used by the remote gem when it contacted the netldi to start the pgsvr, i.e. one of the addresses from the -A argument to that netldi's startnetldi command.

Entries must be valid port numbers (between 1 and 65535), and the second element must be greater than the first. If empty, random ports above 10000 will be used.

Default: empty

On commit record backlog, send sigAborts to multiple sessions

STN_SIGNAL_ABORT_AGGRESSIVE

If the number of outstanding commit records exceeds STN_SIGNAL_ABORT_CR_BACKLOG, then STN_SIGNAL_ABORT_AGGRESSIVE is the maximum number of sessions which will receive SignalAbort when those sessions reference a commit record beyond STN_SIGNAL_ABORT_CR_BACKLOG and newer than the oldest commit record.

Requires GarbageCollection privilege.

Runtime equivalent: #StnSignalAbortAggressive

Default: 0

Min: 0

Max: 500

Automatically start statmonitor on Stone startup

STN_STATMONITOR_ARGS

List of arguments to be passed to the statmonitor executable. If the list is not an empty string, statmonitor will be started automatically by the stone when it starts. The list of possible arguments can be obtained by invoking statmonitor with only the "-h" argument. Note that the stone name is automatically passed to statmonitor and must not be included here.

Example: start statmonitor with a 5 second sample interval with a date and timestamp included in the output file name and place the output file in the directory /statmonFiles:

```
STN_STATMONITOR_ARGS = "-i5 -u5 -d /statmonFiles -F'statmon_%%S_%%P_%%d-%%m-%%y-%%H:%%M:%%S'";
```

Default: "" (Statmonitor will not be started)

Configure the startup timeout on shared page cache

STN_STONE_CACHE_STARTUP_TIMEOUT

Time in seconds allowed for the startup of the shared cache on the stone machine. This is time from when the stone forks the page cache monitor process until it is able to complete its connection to that process.

A scaledTimeout is also computed based on the size of the shared cache.

```
scaledTimeout := cacheSize <= 32GB
  ifTrue:[ (cacheSize / 32GB) * 120]
  ifFalse:[ 120 + ((cacheSize - 32GB / 268GB) * 300) ]
```

If the scaledTimeout exceeds the configured value of STN_STONE_CACHE_STARTUP_TIMEOUT, the scaledTimeout is used for that cache.

Default: 60

Min: 10

Max: 1800

22. Changes in Errors

The following errors have been added.

RT_ERR_UPGRADE_WARNING

Number 2486

This error is triggered when an attempt is made to login to a repository that has not been upgraded to 3.3.

STDB_ERR_BAD_NUMERIC_LIT

Number 1080

This error relates to a new feature, see “Parsing custom numeric literals” on page 23.

23. Utility Changes

For statmonitor changes, see “Changes in Statmonitor and Cache Statistics” on page 50, and for topaz changes, see “Topaz changes” on page 63.

copydbf

Added -u option

The -u option specified to uncompress output records. When not used, the default is to preserve record-level compression from input.

Record-level compression is used by the hotstandby implementation. By default logsender does record-level compression before sending tranlog records to the log receiver. logreceiver does record-level compression if not already compressed before writing to the slave filesystem.

Record-level compression of tranlogs (-u and -c) are distinct from the file-level compression that is used for an entire extent, tranlog, or backup file (-C).

disallowed for backups on raw partitions

copydbf cannot be used to copy backup files to or from for raw partitions. Full backups have been disallowed to raw partitions, so there should be no case where this is needed. Extents and transaction logs on raw partitions can still be accessed with copydbf.

Backups now report position and sequence

Copydbf -i now includes information on the number of backup files in the sequence. The line providing the file type is now similar to:

```
File type: backup  fileId: 0 in a backup set with 2 files
```

Note that the fileId, as in previous releases, indicates the sequence number of this file within the backup.

gslist

Sorted results

The results of gslist are now sorted, by default, by name.

Using the new arguments -s and -S with a sort key, you can specify that the gslist results are sorted by name, owner, time started, or version. -s sorts ascending, and -S sorts descending.

Additional information

gslist -v will now also report if the Stone is in restoreBkup, restoreLogs, recovery, or contRestore (hot standby).

pageaudit

Multithreaded

Pageaudit is now multi-threaded. The number of threads is calculated as the number of extents + the number of CPUs, which will use the most resources to produce the fastest possible completion. The impact of pageaudit on other system resources can be managed by using fewer sessions, with the added `-n` option.

The internal invocation and behavior of pageaudit has been refactored with the multithreading. Now, pageaudit invokes the script `$GEMSTONE/sys/runpageauditgem`. This creates a system gem log file; and the pageaudit run, which starts a GemStone server, produces the usual set of log files. These are deleted on clean exit, but may be retained depending on and customizations to your system log file deletion.

Changes in output

Previously, pageaudit wrote either to a log file specified by the `-l` argument, or to stdout. This output consisted of the stone log, ending with audit details and results. Now, this output does not go to stdout; if no `-l filename` is specified, output goes to a file named `stonename-pageAudit.log`, in the default stone log directory.

Both with and without specification of a log file, the stdout output will include information on the pageaudit status, e.g.:

```
pageaudit[info]: pageaudit gs64stone-pageAudit has been started,  
process 8500  
Page Audit of Repository completed successfully - no issues found
```

The statistical information on the repository that is reported by pageaudit has also been extensively updated, with more information and better details.

Added `-n` option to set number of threads

To specify another number of threads to use for pageaudit, use the added `-n numSessions` option.

Default name for pageaudit stone

Pageaudit starts a stone repository monitor, which can be specified on the command line. If this is not specified, the default name used has changed, from `gs64stone-audit` to `gs64stone-pageAudit`.

pstack

pstack now supports the `-h` usage query.

Added options to collect C stacks only

To collect only C stacks, rather than both C and Smalltalk stacks, use the new `-c` or `-C` options. `-c` prints out full C stacks, `-C` prints out brief C stacks.

removedbf

removedbf is no longer supported for remote use over a network connection. The -n and -p options, which did not function correctly in past releases, have been removed.

The -h option to return help text has been added.

startcachewarmer

startcachewarmer now reports a message in the stone log when the warming completes successfully.

Cache warming also now transparently handles the case where not enough user sessions are available for the specified or default number of cache warmer threads. Provided at least two sessions are available, the cachewarmer will reduce the requested number of sessions to match what is available.

It is now not an error to specify the stone name using NRS (see bug 45604 on page 79)

startnetldi

added -r option

If a netldi with the same name as the argument is already running, the netldi is restarted. This is useful for changing options, and for restarting with a different version. Note that it will not work on running NetLDI versions earlier than 3.3.

waitstone

Waitstone now supports -h usage query.

Waitstone also now will wait for the SymbolGem to be ready; this is to support upgrade in which the SymbolGem requires extra time to convert Symbols.

24. Environment variable changes

GS_DEBUG_SSL_LOG_DIR disallowed in a fast build for security reasons

This environment variable causes diagnostic output to be sent to a file. To avoid a potential security risk, this has been disallowed; the environment variable is available in slow and no-op executables only.

Additional variables to manage log deletion

As described under “Process log files and deletion policy” on page 18, the variables GS_KEEP_ALL_LOGS and GS_FORCE_CLEAN_LOG_FILE_DELETE have been added.

25. Topaz changes

Single-line command serializing

A number of changes in this release are designed to improve readability of topaz code. It is now possible to enter topaz commands on a single line, by separating with a semicolon (;) character. For example:

```
topaz 1> time ; send SystemRepository fileSizeReport ; time
```

These behave as if they were each entered on a separate topaz line, with the following exceptions:

- ▶ SPAWN and SHELL pass the remainder of the line to the OS shell.
- ▶ IFERR, IFERROR are disallowed with semicolon
- ▶ REMARK, RUN, PRINTIT, and DOIT ignore the rest of the line.
- ▶ EXEC will treat as Smalltalk source code anything up to the % character.

UTF-8 related changes

fileformat UTF8 in .topazini now applies to subsequent input

When code is input into topaz, the decoding of characters above the ASCII range depends on the setting of **fileformat**, which may be 8BIT (legacy strings) or UTF8 (input is decoded from UTF-8). This can now be set to UTF8 in .topazini, allowing applications to reliably use UTF-8 for files.

The following order of precedence is applied:

1. A **fileformat** setting within the input file. This does not affect any subsequent input.
2. Setting **fileformat** on the topaz command line, which affects subsequent input.
3. The repository in Unicode comparison mode sets the fileformat to UTF8 on login, which affects subsequent input.
4. Setting **fileformat** to utf8 in .topazini, which affects subsequent input. This is new in v3.3.
5. The topaz default is 8bit.

topaz output files now UTF-8 encoded

When topaz produces an output file such as using output push, the resulting file will always be UTF-8 encoded. If your system is in 8-bit mode (the default, and usual in legacy applications), and you want to use this file as a base for a script by editing for later input, you may need to edit the file to include the line:

- ▶ fileformat UTF8

Topaz executable argument changes

Add -C option to pass in arbitrary configuration parameter values

You can now pass Gem configuration parameters to linked topaz directly on the command line, avoiding the need to create a custom executable configuration file. The -C argument takes a string in standard configuration file syntax.

For example:

```
topaz -l -C 'GEM_TEMPOBJ_CACHE_SIZE = 1GB; GEM_TEMPOBJ_OOMSTATS_CSV = TRUE;'
```

-e, -z, -C and -T disallowed with -r

The RPC mode of topaz do not use configuration files passed in with the -z or -e options, nor temp object cache size passed in with the -T; these only apply in linked mode. Now, -e, -z, and -T, as well as the new option -C, are disallowed in the absence of the -l argument.

Warning on configuration file not found

When topaz starts up, it searches for a configuration file based on the command line arguments -z and -e, otherwise for the environment variables GEMSTONE_SYS_CONF and GEMSTONE_EXE_CONF, and finally on the default name for topaz, gem.conf.

Previously, the name of the configuration files and any errors in finding these files was reported in the first part of the topaz header. Now, an additional warning is printed at the end of the headers section, if either of these cannot be resolved to a valid file. This is particularly intended to catch errors in -z and -e arguments.

Line limit increased

The maximum command line is now 64K characters

New commands

Added EXEC command

A new command has been added, **exec**.

This is similar to **run**, but allows the text of the Smalltalk expression to be on the same line, including the terminating %. For example:

```
topaz 1> exec SystemRepository markForCollection %
```

If the % is not on the same line as the **exec**, all input up until a line containing % as the first character is part of the execution source.

Support for setting transaction mode, add SET TRANSACTIONMODE

When topaz logs in, by default it is in automatic transaction mode and in transaction. This is convenient for making modifications, but runs the risk of idle sessions causing a commit record backlog.

GemStone supports changing modes using System transactionMode:, with one of #autoBegin, #manualBegin, and #transactionless.

To allow topaz sessions to default to other transaction modes, the SET command now includes TRANSACTIONMODE. This accepts (case insensitive) **autobegin**, **manualbegin**, and **transactionless**. The default remains **autobegin**. This setting is for the topaz instance, and will apply to each setting on login.

For more information on transaction modes, see the *System Administration Guide*.

Added DEBUGRUN command

The **debugrun** command is similar to run, but execution stops at the first source point within the source text to allow stepping through code for debugging.

Added LITERALS command

A new command has been added, **literals** *anObject*.

This returns the methods containing *anObject* as a literal reference; equivalent to

```
ClassOrganizer new literalsReport: anObject
```

DISPLAY and OMIT added PAUSEONWARNING

display pauseonwarning causes topaz processing to pause on a compiler warning, and **omit pauseonwarning** turns off that behavior.

When a compiler warning occurs, and if the topaz stdin is from a terminal, then this writes a message "Pausing after warning ..." to stdout, and waits for an end of line on stdin before continuing topaz execution. A Control-C on stdin will terminate the pause and terminate further processing of input files.

Changes in the ordering of lines within status output

The status command outputs information on the current state of topaz variables. There are changes in the ordering of lines and what information is included.

The sourcestringclass and fileformat information is now under the current settings section.

Lines have also been added for interactive pause on warnings and transaction mode.

topaz fileout uses protocol in Behavior

The FILEOUT command in topaz now uses the Behavior protocol to generate fileout text.

26. GCI changes

The thread-safe GCI now implements keep-alive functionality using the configuration parameter `GEM_RPC_KEEPALIVE_INTERVAL`, as the standard GCI does.

Added functions

GciAll7Bit

```
(BoolType) GciAll7Bit(
    const char* str,
    size_t *len
);
```

`GciAll7Bit` returns `TRUE` if contents of `str` are all characters `<= 127`. `strlen(str)` is returned in `*len`.

GciExecuteStrFetchBytes, GciNbExecuteStrFetchBytes

```
(int64) GciExecuteStrFetchBytes(
    const char* source,
    int64 sourceSize,
    OopType sourceClass,
    OopType contextObject,
    OopType symbolList,
    ByteType *result,
    int64 maxResultSize
);
```

```
(void) GciNbExecuteStrFetchBytes(
    const char* source,
    int64 sourceSize,
    OopType sourceClass,
    OopType contextObject,
    OopType symbolList,
    ByteType *result,
    int64 maxResultSize,
);
```

`GciExecuteStrFetchBytes` executes `source` and assuming the execution result is a byte format object, returns that String in `*result`. `GciNbExecuteStrFetchBytes` is a non-blocking implementation of this function.

The function result is the number of bytes returned in `*result`, or `-1` if an error is available to be fetched with `GciErr`.

`source`

The string to be executed.

`sourceSize`

the number of bytes in the 'source' array, usually `strlen(source)`. if `sourceSize == -1`, `strlen(source)` is used.

sourceClass

The OOP of the class that 'source' should be converted to. Examples are OOP_CLASS_STRING, OOP_CLASS_Utf8, OOP_CLASS_Unicode7 .

contextObject

The OOP of any GemStone object. The code to be executed is compiled as if it were an instance method in the class of contextObject. A value of OOP_NO_CONTEXT means no context.

symbolList

The OOP of a GemStone symbol list (that is, an Array of instances of SymbolDictionary). The compiler uses the symbolList to resolve symbolic references in the code in source. A value of OOP_NIL means to use the default symbol list for the current GemStone session (that is, System myUserProfile symbolList).

Execution is in environment 0 using GCI_PERFORM_FLAG_ENABLE_DEBUG .

Changed functions

```
(int64) GciFetchUtf8Bytes_(
    OopType aString, int64 startIndex, ByteType *buf,
    int64 bufSize, OopType *utf8String, int flags);
```

In the flags argument, you may now pass 2, indicating to put a generated description in *buf instead of signalling an error.

Deprecated functions

The following functions have been deprecated in v3.3:

GciClampedTraverseObjs
use GciClampedTrav

GciFetchObjInfo
use GciFetchObjectInfo

GciNbClampedTraverseObjs
use GciNbClampedTrav

GciNbPerformTraverse, GciNbPerformTraverse_
use GciNbPerformTrav

GciStep, GciStep_, GciNbStep, GcNbStep_
Debuggers should use instance methods in GsProcess.

GciPathToStr

GciStrToPath

Removed Functions

The following obsolete functions have been removed:

GciInitNoSharedCache

GciUseSharedCache

The following GSCI functions have been removed; they are no longer valid with 64-bit stats:

```
GcsiShrPcMonStatAtOffset
GcsiStnStatAtOffset
```

Other Changes affecting the GCI interface

User actions that invoke Smalltalk nesting limit is now two

User actions may invoke Smalltalk code that itself calls other user actions. This is now limited to a nesting depth of two.

Linux Compile and Link Information

Compiler version

```
Red Hat 6.x: gcc/g++ 4.4.7
Red Hat 7.x: gcc/g++ 4.8.5
Ubuntu Linux 12.04: gcc/g++ 4.6.3
SUSE Linux 12: gcc/g++ 4.8.5
```

Debugger version

```
Red Hat 6.x: GNU gdb 7.2-83.el6x
Red Hat 7.1: GNU gdb 7.6.1-64.el7
Ubuntu Linux 12.04: GNU gdb 7.7
SUSE Linux 12: GNU gdb 7.9.1
```

Compiling a user action or GCI application

```
g++ -fmessage-length=0 -fcheck-new -O3 -ggdb -m64 -pipe
-D_REENTRANT -D_GNU_SOURCE -pthread -fPIC -fno-strict-aliasing
-fno-exceptions -I$GEMSTONE/include -x c++ -c userCode.c
-o userCode.o
```

The following warn flags are recommended for compilation:

```
-Wformat -Wtrigraphs -Wcomment -Wsystem-headers -Wtrigraphs
-Wno-aggregate-return -Wswitch -Wshadow -Wunused-value
-Wunused-variable -Wunused-label -Wno-unused-function
-Wchar-subscripts -Wmissing-braces -Wmultichar -Wparentheses
-Wsign-compare -Wsign-promo -Wwrite-strings -Wreturn-type
-Wuninitialized
```

Linking a user action library

```
g++ -shared -Wl,-Bdynamic,-hlibuserAct.so userCode.o
$GEMSTONE/lib/gciualib.o -o libuserAct.so -m64 -lpthread
-lcrypt -ldl -lc -lm -lrt -lpam -lpam_misc -Wl,-z,muldefs
-Wl,--warn-unresolved-symbols
```

Linking a GCI application

```
g++ userCode.o $GEMSTONE/lib/gcirtlobj.o -Wl,-traditional
-Wl,--warn-unresolved-symbols -m64 -lpthread -lcrypt -ldl -lc
-lm -lrt -lpam -lpam_misc -Wl,-z,muldefs -o userAppl
```

Solaris on SPARC Compile and Link Information

There are corrections in the compile line, but otherwise there is no change from v3.2; for completeness, all information is included here.

Compiler version

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

Debugger version

Sun Dbx Debugger 7.5 Patch 121023-07 2010/09/22

Compiling a user action or GCI application

```
CC -xO4 -xcode=pic32 -xarch=v9 -mt -xchip=ultra2
-I$GEMSTONE/include -features=no%except -D_REENTRANT
-D_POSIX_PTHREAD_SEMANTICS -features=no%anachronisms
-c userCode.c -o userCode.o
```

Linking a user action library

```
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 -lpam -lCrun -znodefs
```

Linking a GCI application

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

Solaris on x86 Compile and Link Information

There are corrections in the compile line, but otherwise there is no change from v3.2; for completeness, all information is included here.

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
-features=no%except -c userCode.c -o userCode.o
```

Linking a user action library

```
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 -lpam -lCrun -z nodefs
```

Linking a GCI application

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

AIX Compile and Link Information

Compiler version

AIX 6.1, and 7.1 on POWER7: IBM XL C/C++ for AIX, V11.1

AIX 7.1 on POWER8: IBM XL C/C++ for AIX, V13.1.2

Debugger version

dbx

Compiling a user action or GCI application

```
xlC_r -O3 -qstrict -qalias=noansi -q64 +- -qpic
-qthreaded -qarch=pwr6 -qtune=balanced -D_LARGEFILE64_SOURCE
-DFLG_AIX_VERSION=version -D_REENTRANT -D_THREAD_SAFE
-qminimaltoc -qlist=offset -qmaxmem=-1 -qsuppress=1500-010:1500
-029:1540-1103:1540-2907:1540-0804:1540-1281:1540-1090 -qnoeh
-I$GEMSTONE/include -c userCode.c -o userCode.o
```

Depending on your version of AIX, you need to include either `-DFLG_AIX_VERSION=61` or `-DFLG_AIX_VERSION=71`.

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

Linking a user action library

```
xlC_r -G -Wl,-bdatapsize:64K -Wl,-btextpsize:64K
-Wl,-bstackpsize:64K -q64 userCode.o $GEMSTONE/lib/gciualib.o
-o libuserAct.so -e GciUserActionLibraryMain -L/usr/vacpp/lib
-lpthreads -lc_r -lC_r -lm -ldl -lbsd -lpam -Wl,-berok
```

Linking a GCI application

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

DARWIN Compile and Link Information

Compiler version

Apple LLVM version 6.0 (clang-600.0.56)

Debugger version

GNU gdb 6.3.50.20050815-cvs (Thu Nov 21 15:33:19 UTC 2013)

Compiling a user action or GCI application

```
g++ -fmessage-length=0 -O3 -ggdb -m64 -pipe -fPIC
-fno-strict-aliasing -D_LARGEFILE64_SOURCE -D_XOPEN_SOURCE
-D_REENTRANT -D_GNU_SOURCE -I$GEMSTONE/include -x c++
-c userCode.c -o userCode.o
```

The following warn flags are recommended for compilation:

```
-Wformat -Wtrigraphs -Wcomment -Wsystem-headers -Wtrigraphs
-Wno-aggregate-return -Wswitch -Wshadow -Wunused-value
-Wunused-variable -Wunused-label -Wno-unused-function
-Wchar-subscripts -Wconversion -Wmissing-braces -Wmultichar
-Wparentheses -Wsign-compare -Wsign-promo -Wwrite-strings
-Wreturn-type
```

Linking a user action library

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

Linking a GCI application

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

Windows Compile and Link Information

This information is unchanged from v3.2

Compiler/Debugger version

Microsoft Visual Studio 2010 Version 10.0.30319.1 RTMRel

Microsoft Visual C++ 2010 01021-532-2002102-70611

Compiling a GCI application

```
c1 /W3 /Zi /MD /O2 /Oy- -DNDEBUG /TP /nologo /D_LP64 /D_AMD64_  
/D_CONSOLE /D_DLL /DWIN32_LEAN_AND_MEAN  
/D_CRT_SECURE_NO_WARNINGS  
/DNATIVE /I 'VisualStudioInstallPath\atlmfc\include'  
/I 'VisualStudioInstallPath\VC\include'  
/I 'C:\Program Files (x86)\Microsoft  
SDKs\Windows\v7.0A\Include'  
/I '%GEMSTONE%\include' -c userCode.c -FouserCode.obj
```

Linking a GCI application

```
link /LIBPATH:"VisualStudioInstallPath\VC\lib\amd64"  
/LIBPATH:"C:\Program Files (x86)\Microsoft  
SDKs\Windows\v7.0A\Lib\x64"  
/OPT:REF /INCREMENTAL:NO /MAP /nologo /MANIFEST  
/MANIFESTFILE:userAppl.exe.manifest  
/MANIFESTUAC:"level='asInvoker'" userCode.obj  
%GEMSTONE%\lib\gcirpc.lib ws2_32.lib netapi32.lib advapi32.lib  
comdlg32.lib user32.lib gdi32.lib kernel32.lib winspool.lib  
/out:userAppl.exe
```


Bugs Fixed

The following bugs have been fixed in this release.

AIO page server fatal errors during startup may leave Stone hanging

During startup, the sockets that connect the Stone and AIO page servers do not detect disconnects. If the AIO page servers encountered an error such as a write failure, it exited (correctly); but the Stone continued to wait for a response. (#44587)

Multithreaded scans now usable from remote Gems

In previous 3.x versions, the multi-threaded scans used by operations such as markForCollection could only be run multithreaded on Gems on the Stone's host. When executed from a remote Gem, these operations would run single-threaded. This restriction has been removed, and now remote Gems can run multithreaded scans. (#41178, #41177)

On first use of a remote host, linked logins may fail

Remote connections require that each host have a `/opt/gemstone/locks/gemstone.hostid` file to identify themselves. On a remote host, if this file is not found, it is created when a netldi starts up. However, if there has never been a netldi started, and the first GemStone process is a linked session which no longer requires a netldi, the login failed with a file not found error. (#45111)

Garbage Collection issues

Symbol Garbage Collection not usable

There were bugs in the primitive code supporting garbage collection of unused Symbols, as enabled by `STN_SYMBOL_GC_ENABLED`; it did not complete correctly with a non-trivial number of symbols or with corner cases of symbol references. Symbol GC was disabled in v3.2.7 and later. (#45403)

markGcCandidates issues

The internal phases in markGcCandidates were numbered incorrectly with respect to the object table sweep phase number, resulting in some phases sweeping pages that incorrectly include scavengeable pages. (#45227).

Since markGcCandidates is no longer of significant benefit, this function has been removed in v3.3.

Exception Handling issues

Exception handler #resume from primitive operation returned nil

If an exception handler was triggered from within low-level C code in a primitive, nil was incorrectly returned as the result of the Smalltalk method that called the primitive. (#44375)

Incorrect return from exception handler with non-local return in ensure: block

A non-local return within an ensure block did not correctly override the on:do: handling. (#45056)

For example, the following previously returned #ok, while the correct return should be 7.

```
[ [3 zork] ensure: [^7].
  ] on: Error do: [:ex | ex return: #ok]
```

Blocks may return wrong values when nested variable is assigned in outer scope

A block that is nested within another block can, in some cases, have the wrong value for a variable that is declared in an outer scope. An assignment to the variable performed in the outer scope is not always propagated to the inner block when it should be. (#44478).

For example, the following previously returned 1, while the correct result should be 100.

```
[ | temp myBlock |
  temp := 1.
  myBlock := [ temp ].
  temp := 100.
  myBlock value ] value
```

Unable to trap soft break in linked topaz

Soft breaks can normally be caught by a handler. While this worked in RPC topaz, in linked the soft break was treated like a hard break. (#43502)

#halt returned the result of resume, not receiver

Object halt returned the result of the signal, rather than the receiver. This created problems continuing after a halt. (#45126)

A do: loop on an Interval does not behave correctly with step through

When stepping through (using GBS or topaz **step thru** commands) a `do: loop` on an instance of `Interval`, on reaching the end of the block, the step through behaves as a `continue`. (#45320)

Some ANSI exception classes inherited error number

The classes `CompileWarning`, `Deprecated`, `InterSessionSignal`, `TestFailure` and `ReusableTestFailure` inherited the error number from their superclass. (#43411)

Improvements in Error reporting

CompileError display

Compilation errors that occur in some block contexts; for example, In code sent to External sessions; did not display useful information, only a message such as 'a `CompileError` occurred (error 1001), compilation errors -- parameter 1 is error descriptor'. Now, the error text is provided. (#45626)

Improved error reporting for String and MultiByteString>>encodeUsing:

When the primitive failed, no useful information was provided. (#44140)

Improved error for attempt to store into internal class

Working with instance of an internal class (such as `LargeObjectNode`), the error returned was "Unable to store forward reference to object that is not allocated to this session", which is not clear; it now says "Attempt to store a reference to a hidden object". (#44232)

Error message referencing primitive -1/-2 does not exist is confusing

An error message saying "Primitive number -1 does not exist in the virtual machine" is incorrectly worded. These values are error indicators, with -1 meaning it does not exist, and -2 meaning it should not be used. The most likely cause is incompatible Stone image and executable versions. (#45692)

midLevelCacheConnect: error could return empty string rather than error

For certain kinds of errors from an attempt to connect to a mid-level cache using `midLevelCacheConnect: *`, the connection could fail, but the return value was an empty string rather than an error message. (#45142)

OutOfRange errors in setting configuration values did not include details

The error returned when setting a runtime configuration parameter did not include details (#44107).

Improved error when using a name that matches a system service

Service names that require ports, such as `gs64stone` and `gs64ldi`, look for assigned port numbers in the OS services database. If a name is used that is already defined by the system, such as `"auth"`, it will find this port, which may not be usable by the GemStone process. Now, the log files will indicate how the port was determined to distinguish this case. (#45268)

Kernel methods not respecting transient changes to symbol list

Some kernel methods referenced `System myUserProfile symbolList`, which did not respect any transient changes to the symbol list. This has been changed to use `GsSession currentSession symbolList`. (#45236)

Binary selectors `/-` and `/-/` failed

These legal binary selectors were not compilable. (#45062)

Index and Query issues

GsQuery readStream with equality does not correctly handle cases where comparable objects are not equal

Since `Strings` and `Symbols`, for example, shared a superclass and can be ordered using `<=`, etc., it is allowed to mix instances of each together as objects that are in an index. However, a `String` and a `Symbol` with the same contents are not equal, in other words, `a <= b` and `a >= b` does not mean `a = b`; so a query using `=` must only return instances of the appropriate class. `GsQuery readStream` incorrectly returned objects of any class in which `a >= b` and `a <= b`, rather than requiring `a = b`. (#44079)

removeAllIndexes could run out of memory

`removeAllIndexes` bypassed the collections and updated the dependency maps for each index element directly. On a large collection, this could result in `AlmostOutOfMemory` errors. Now, the `DependencyMaps` are put into a hidden set for processing, where they do not affect temporary object memory.

Index audit does not catch entries with equal keys, but OOPS out of order

An equality index creates a `btree` that may include many keys with equal values. There were conditions in which the `btree` entries were not ordered deterministically; this was not caught by the `indexAudit` code. (#44669).

IndexManager usageReport may get MNU #nextWord with Portable Streams installed

The code invoked by the `IndexManager >> usageReport` method calls `PositionableStream >> nextWord`, which was missing from portable streams (`PositionableStreamPortable` and subclasses). (#45807)

Linux network statistics incorrectly relative to start of statmonitor

The Linux network statistics InputPackets, InputKBytes, MulticastInputPackets, OutputPackets, and OutputKBytes were relative to when statmonitor started recording, rather than absolute values. (#45052)

GEM_PGSRV_COMPRESS_PAGE_TRANSFERS only compressed one direction

The Gem configuration option GEM_PGSRV_COMPRESS_PAGE_TRANSFERS, when true, compresses page transfers from the Page Server to the Gem. However, it did not compress transfers in the other direction, from the Gem to the Page Server. (#44783)

GciEnableFreeOopEncoding and GciGetFreeOopsEncoded broken

The encoding functions in the GCI were not working correctly. GciGetFreeOops failed if called after GciEnableFreeOopEncoding, and GciGetFreeOopsEncoded always failed. (#44738)

NetLDI socket leak on incompatible connection

When the NetLDI opened a connection to an incompatible version of GemStone, there was a code path in which the socket connection was not closed, resulting in a socket leak in the NetLDI. (#45902)

StringKeyValueDictionary removeKey: may fail incorrectly

Dictionaries locate keys using hash, and may lookup Strings and Symbols interchangeably, or (in Unicode comparison mode) mix Unicode and traditional Strings. However, removeKey: used = to compare keys. It was therefor possible for at : to return a value, but removeKey: to fail. (#44097)

StringKeyValueDictionary >>at:ifAbsent: did not use block on nil argument

The methods StringKeyValueDictionary >> at:ifAbsent: and StringKeyValueDictionary >> at:otherwise: returned the error #rtErrNilKey (error 2090) when the specified key was nil, rather than executing the block supplied to handle nil keys. (#45367)

Risk of SEGV when VW on Linux invokes C code

When C functions are invoked from the VisualWorks executable, the stack pointer is not aligned correctly 16 bit. This is related to changes in the stack alignment rules in recent versions of gcc, vs. the compiler versions used for VisualWorks executables. The potential for problems is when using GBS on Linux with application user actions (which are deprecated in GBS), and depending on the compiler version and optimization used for the user actions. (#42632)

Topaz issues

Linked topaz retained some changes to internal globals over logins

A customized value for IcuCollator default, IcuLocale default, and Locale, were retained in a linked topaz process, and persisted through logout and login. (#45532)

Topaz nbstep command broken

The topaz command **nbstep** returned a message not understood error on `#_stepOverInFrame:mode:replace:tos:` (#44590)

Topaz shell does not work well with lineeditor

Using the topaz **shell** command may not work with the lineeditor enabled. The lineeditor is enabled by default. (#45249)

Topaz filename expansion did not recognize ~ in path

A tilde, representing the home directory, is now recognized by topaz commands that use a filename, such as **output push** and **fileout tofile:**. (#44018)

Hot and Warm Standby issues

Standby system may lag behind primary significantly after large reclaim

When a large reclaim operation occurs on the primary system, the same reclaim operation will need to be completed on the standby system. This could take significant time, during which period the standby lagged behind the primary. This was difficult to address on the standby, since the configuration of reclaim could only be identical to that on the primary (changes required commit, which is disallowed during restore). (#44265)

In v3.3, the code that coordinates between the transaction log restore and the ReclaimGem has been made more efficient. You may now configure reclaim on the standby differently than on the primary, using the run-time configuration options described under "Runtime configuration of ReclaimGem and AdminGem" on page 24

Hot standby connection may timeout with large number of tranlogs

When there are a large number tranlogs in the master systems tranlog directory, the logsender-logreceiver connection may timeout. (#45473)

Possible for logreceiver to get tranlogs from multiple repositories

In hotstandby systems that are incorrectly configured, it was possible for the logreceiver to reconnect to a logsender that is associated with a different repository than the one previously connected to. Further information has been added to tranlogs so that such inconsistencies are detected. (#43124)

fork: did not handle single quotes correctly

`GsHostProcess class >> fork:` accepts a String argument containing a command line. This string correctly handled double quotes, but not single quotes. (#45386)

Checkpoints not written during replay of possible dead objects

Stopping and restarting the Stone during the restore of transaction logs that include a reclaim of possible dead, required that the reclaim restart from the beginning. Now, checkpoints are written during the replay. The main impact of this would be on warm and hot standby systems. (#45901)

GsFile does not handle file names with extended characters

When passing a string that includes characters with codepoints outside the ASCII range to a GsFile file open method, the resulting filename may be truncated or contain invalid characters. (#45326, #44786)

startcachewarmer handling of NRS in stone name

In some earlier releases, using NRS syntax in the stone name argument to startcachewarmer results in a crash; in later version, the startcachewarmer would not crash or error, but it did not perform the warming operation. (#45604)

Upgrade and Conversion issues

Conversion problems with very large Symbols

GemStone/S 64 Bit does not allow Symbols larger than 1024 bytes. However, repositories converted from 32-bit may contain Symbols that are larger, including ones larger than 8K that require LargeObjectNodes for internal storage.

During the conversion to 3.x, startstone -C resorts AllSymbols; this sort operation will fail if any symbols over 8K are present, causing the startstone -C to fail. (#45923)

As of v3.3, the startstone -C conversion will not fail; any symbols over 8K are removed from AllSymbols. These symbols remain as uncanonical symbols and can be removed or, if unreferenced, will be removed by garbage collection. It is recommended that before conversion, you examine your repository for large symbols and convert appropriately.

Filein class definitions that fail in `_equivalentSubclass:...` did not provide details

When a class definition is filed in, it invokes the method

```
Class>>_equivalentSubclass:superCls:name:newOpts:newFormat:
      newInstVars:newClassInstVars:newPools:newClassVars:inDict:
      constraints:isKernel:
```

to determine if a new version of the class should be created, or if the definition is the same in all ways. When this method determined that the classes were not the same, it did not provide details about the specific cause of the failure, which made analysis of upgrade/conversion issues difficult. (#45836)

ClassOrganizer instances persisted through upgrade had missing symbol list

Persisted instances of ClassOrganizer from a pre-3.2 repository, when upgraded to 3.2 or later, were not fully initialized and encountered errors on use. (#44199)

Upgrade issues with #GemStoneRCLock

There were code paths in `upgradeImage`, in which the handling of `#GemStoneRCLock` was incorrect and could result in upgrade failure. (#45857)

filein of older code could introduce comment class method

In version earlier than 3.1, classes comments could be implemented by defining individual `#comment` class methods. In v3.1 and later, the comment for a class is in a separate field in the `Class`, and user classes should not reimplement class methods `comment` or `comment : .` Now, filing in older code that includes `comment` or `comment : methods` will get a compiler warning. (#44566)

Improved handling of login when repository upgrade required

When attempting to login to a repository that has been started on extents from an older version, when `upgradeImage` is required and has not been run, previously one of a number of unspecified errors would occur. Now, this is now handled explicitly with the new error 2486, `RT_ERR_UPGRADE_WARNING`, and a message that repository upgrade is needed. (#44183)

SymbolGem that is slow to exit is made invalid, not handled cleanly

When the `SymbolGem` was requested to stop and does not shutdown within the timeout, the Stone sets its cache state to invalid. The terminating `SymbolGem` reports this as a `lostOT`, and does not correctly handled the exit. (#45265)

Compiling method with Array SymbolDictionary required #OtherPassword privilege

When compiling a method (using `compileMethod:dictionaries:category:`), with the target `dictionaries:` argument being an instance of `Array`, the `#OtherPassword` privilege was required. This is not correct; this should require `#CodeModification`, but not `#OtherPassword`. (#44433)

Inconsistent lock status after failed commit on temporary locked object

If a commit fails when there is a lock on a temporary object, after an abort the lock status becomes inconsistent. The lock cannot be removed, and methods to access lock status are inconsistent. (#45744)

SlotsTotalCount off-by-one

The value recorded by the `ShrPcMonitor` cache statistics `SlotsTotalCount` was always one higher than the actual value. (#45713)

gslist -x did not show startnetldi -D value

The `-D` option was added in recent versions of GemStone. The value of this argument was not reported in the results of `gslist -x`. (#45935)