GemStone®

GemStone/S 64 Bit Release Notes

Version 2.2.2

August 2007

GemStone S 64™

INTELLECTUAL PROPERTY OWNERSHIP

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

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

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

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

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

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

COPYRIGHTS

PATENTS

GemStone is covered by U.S. Patent Number 6,256,637 “Transactional virtual machine architecture”, Patent Number 6,360,219 “Object queues with concurrent updating”, and Patent Number 6,567,905 “Generational Garbage Collector”. GemStone may also be covered by one or more pending United States patent applications.

TRADEMARKS

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

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

Sun, **Sun Microsystems**, **Solaris**, and **SunOS** are trademarks or registered trademarks of Sun Microsystems, Inc. All **SPARC** trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. **SPARCstation** is licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

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

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

Microsoft, **MS**, **Windows**, **Windows 2000** and **Windows XP** 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.

AIX and **POWER4** are trademarks or registered trademarks of International Business Machines Corporation.

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. All terms mentioned in this documentation that are known to be trademarks or service marks have been appropriately capitalized to the best of our knowledge; however, GemStone 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.

GemStone Systems, Inc.
1260 NW Waterhouse Avenue, Suite 200
Beaverton, OR 97006

Preface

About This Documentation

These release notes describe changes in the GemStone/S 64 Bit version 2.2.2 release. We recommend that everyone migrating to this version read these release notes before beginning installation, testing or development.

No separate Installation Guide is provided with this release. For instructions on installing GemStone/S 64 Bit version 2.2.2, or upgrading or converting from previous products or versions, see the Installation Guide for version 2.2.

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

Terminology Conventions

This document uses the following terminology:

The term "GemStone" is used to refer both to the product, GemStone/S 64 Bit, or previous GemStone/S server products; and to the company, GemStone Systems, Inc.

Technical Support

GemStone provides several sources for product information and support. The product-specific manuals and online help provide extensive documentation, and should always be your first source of information. GemStone Technical Support engineers will refer you to these documents when applicable.

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

GemStone's Technical Support website provides a variety of resources to help you use GemStone products. Use of this site requires an account, but registration is free of charge. To get an account, just complete the Registration Form, found in the same location. You'll be able to access the site as soon as you submit the web form.

The following types of information are provided at this web site:

Help Request allows designated support contacts to submit new requests for technical assistance and to review or update previous requests.

Documentation for GemStone/S 64 Bit is provided in PDF format. This is the same documentation that is included with your GemStone/S 64 Bit product.

Release Notes and **Install Guides** for your product software are provided in PDF format in the Documentation section.

Downloads and **Patches** provide code fixes and enhancements that have been developed after product release. Most code fixes and enhancements listed on the GemStone Web site are available for direct downloading.

Bugnotes, in the Learning Center section, identify performance issues or error conditions that you may encounter when using a GemStone product. A bugnote describes the cause of the condition, and, when possible, provides an alternative means of accomplishing the task. In addition, bugnotes identify whether or not a fix is available, either by upgrading to another version of the product, or by applying a patch. Bugnotes are updated regularly.

TechTips, also in the Learning Center section, provide information and instructions for topics that usually relate to more effective or efficient use of GemStone products. Some Tips may contain code that can be downloaded for use at your site.

Community provides customer forums for discussion of GemStone product issues.

Technical information on the GemStone Web site is reviewed and updated regularly. We recommend that you check this site on a regular basis to obtain the latest technical information for GemStone products. We also welcome suggestions and ideas for improving and expanding our site to better serve you.

You may need to contact Technical Support directly for the following reasons:

- Your technical question is not answered in the documentation.
- You receive an error message that directs you to contact GemStone Technical Support.
- You want to report a bug.
- You want to submit a feature request.

Questions concerning product availability, pricing, keyfiles, or future features should be directed to your GemStone account manager.

When contacting GemStone Technical Support, please be prepared to provide the following information:

- Your name, company name, and GemStone/S license number
- The GemStone product and version you are using
- The hardware platform and operating system you are using
- A description of the problem or request
- Exact error message(s) received, if any

Your GemStone support agreement may identify specific individuals who are responsible for submitting all support requests to GemStone. If so, please submit your information through those individuals. All responses will be sent to authorized contacts only.

For non-emergency requests, the support website is the preferred way to contact Technical Support. Only designated support contacts may submit help requests via the support website. If you are a designated support contact for your company, or the designated contacts have changed, please contact us to update the appropriate user accounts.

Email: support@gemstone.com

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

Requests for technical assistance may also be submitted 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 that is non-operational. In these cases, please also submit your request via the web or email, including pertinent details such error messages and relevant log files.

If you are reporting an emergency by telephone, select the option to transfer your call to the technical support administrator, who will take down your customer information and immediately contact an engineer.

Non-emergency requests received by telephone will be placed in the normal support queue for evaluation and response.

24x7 Emergency Technical Support

GemStone offers, at an additional charge, 24x7 emergency technical support. This support entitles customers to contact us 24 hours a day, 7 days a week, 365 days a year, if they encounter problems that cause their production application to go down, or that have the potential to bring their production application down. For more details, contact your GemStone account manager.

Training and Consulting

Consulting and training for all GemStone products are available through GemStone's Professional Services organization.

- Training courses are offered periodically at GemStone's offices in Beaverton, Oregon, or you can arrange for onsite training at your desired location.
- Customized consulting services can help you make the best use of GemStone products in your business environment.

Contact your GemStone account representative for more details or to obtain consulting services.

Contents

Chapter 1. GemStone/S 64 Bit 2.2.2 Release Notes

Chapter 2. GemStone C Statistics Interface

Chapter 1

GemStone/S 64 Bit 2.2.2 Release Notes

Overview

GemStone/S 64 Bit 2.2.2 is a new version of the GemStone/S 64 Bit object server. This release provides several new features and fixes a number of bugs; we recommend everyone using or intending to upgrade to GemStone/S 64 Bit 2.x, upgrade to this new version. The details of these changes are provided in this document.

These release notes provide changes between the previous version of GemStone/S 64 Bit, version 2.2.1, and version 2.2.2. If you are upgrading from a version prior to 2.2.1, please also review the release notes for each intermediate release to see the full set of changes.

No separate Installation Guide is provided with this release. For installation instructions, use the Installation Guide for version 2.2, with the following changes and notes:

- Conversion now resets the DataCurator password to the default "swordfish", similarly to the way GcUser's password is handled. Following conversion, be sure to reset the DataCurator password as well as the SystemUser and GcUser passwords.
- Version 2.2.2 includes a new conversion option to collect instances of Float objects during conversion, for later processing. However, use of this option requires that certain features are available in the originating repository. These features are not currently available in any released product or version. Collecting instances of Float objects will be possible when converting from the upcoming GemStone/S 64 Bit 1.2 and GemStone/S 6.2 releases, to this version.
- The conversion environment variable $GEMSTONE_22 is unchanged for later 2.2.x releases.

Changes and New Features

DateTime Changes

DateTimes now support millisecond resolution

While DateTime has included internal millisecond values, the official support was to the seconds level. This has been modified so DateTime operations are fully supported to millisecond resolution.

The following methods have been added:

```
DateTime >> asStringMs
DateTime >> asStringGmtMs
DateTime >> addMilliseconds:
```

ByteArray methods for DateTimes now support cross-endian use

Previously, DateTimes that were stored in ByteArrays could only be extracted correctly on a platform which is the same byte order as the platform on which they were created. This is reported as bug #37125, see page 1-10.

The following methods have been deleted:

```
ByteArray >> dateTimeAsUnsigned32At:
ByteArray >> dateTimeAsUnsigned32At:put:
```

The following methods have been added:

`ByteArray >> dateTime32At:`
`ByteArray >> dateTime32At:put:`
Reads and stores the DateTime in big-endian byte order, truncating to seconds resolution.

`ByteArray >> dateTime32NativeAt:`
`ByteArray >> dateTime32NativeAt:put:`
Reads and stores the DateTime in the native byte order of the gem process, truncating to seconds resolution.

`ByteArray >> dateTime64At:`
`ByteArray >> dateTime64At:put:`
Reads and stores the DateTime in big-endian byte order, with millisecond resolution.

Existing uses of `dateTimeAsUnsigned32At:` should be changed to use `dateTime32At:` or `dateTime32NativeAt:`, depending on whether the ByteArrays were originally written on Linux, which is the only supported small-endian platform.

Code should be modified to use `dateTime64At:` to store the full millisecond precision of DateTime instances.

New inverse privileges

New inverse privileges have been added, which operate differently than normal privileges; the affected methods are permitted except if the user has the specified privilege. The privileges and affected behavior are listed in the table below.

The state of NoUserAction, NoGsFileOnServer, and NoGsFileOnClient are cached in the Virtual Machine at session login, and changes to a UserProfile for these 3 privileges will only be seen in sessions that login after those changes have been committed.

Privilege Name	Affected actions	Cached on login
NoPerformOnServer	performOnServer:	no
NoUserAction	Loading of any user action library.	yes
NoGsFileOnServer	Any GsFile operation which accesses a file on the server, i.e. access from the gem process to the file system.	yes
NoGsFileOnClient	Any GsFile operation which accesses a file on the client, i.e. access from the remote GCI process to the file system.	yes

RcQueue and continueTransaction

RcQueue has had optimizations in previous releases that circumvented the process of safely handling continueTransaction. To avoid risk, updating an RcQueue after executing a continueTransaction is no longer allowed.

After executing continueTransaction, if you attempt to update an RcQueue, the new error RT_ERR_RC_UPDATE_DISALLOWED is raised, and you will not be able to commit the transaction; you must abort. Attempting to commit will raise the new error RT_ERR_COMMIT_DISALLOWED_UNTIL_ABORT.

The following new method has been added:

```
System class >> inContinueTransaction
```

Cache Statistics Changes

GCSI

The ability to write C programs to read cache statistics, avoiding the need to log in a gem session, has been added in this release. See Chapter 2, ”GemStone C Statistics Interface” for more information.

Statmonitor and VSD changes

The way cache statistics are recorded has changed with this release.

Statmonitor files produced by version 2.2.2 cannot be read using versions of VSD that were provided with earlier releases. You must use the version of VSD (version 3) provided with 2.2.2 to read version 2.2.2 statmonitor files or monitor version 2.2.2 repositories.

VSD version 3 will read statmonitor files produced by earlier versions of GemStone/S 64 Bit or GemStone/S.

LostOT handling

The process of handling sessions that do not respond to SigAborts has changed in this release. In addition to the process changes, some special cases have been fixed to make handling of lostOT more robust, and several bugs have been fixed.

Following is a full description of the process, starting with the sigabort:

Sessions that are not in transaction and are referencing the oldest CR when there are more than STN_SIGNAL_ABORT_CR_BACKLOG commit records are sent a sigabort.

After sending the sigabort, the stone waits for STN_GEM_ABORT_TIMEOUT before further action. If all references to the CR are from transactionless sessions, only wait for STN_GEM_ABORT_TIMEOUT / 4.

If the session does not abort or commit within the STN_GEM_ABORT_TIMEOUT, lostOT processing is done, depending on the setting for STN_GEM_LOSTOT_TIMEOUT.

If STN_GEM_LOSTOT_TIMEOUT > 0, then a sigLostOT is sent to all sessions still referencing the oldest commit record, with a timeout of STN_GEM_LOSTOT_TIMEOUT.

If STN_GEM_LOSTOT_TIMEOUT = 0 (a timeout of 0), it does not wait, but immediately continues with the lostOT handling.

If the session does not respond (by aborting, begin transaction, or logging out) within the STN_GEM_LOSTOT_TIMEOUT, the stone does the following:

- Retracts the session's commit record
- Poisons its cache slot (the session will get a fatal error if it references the shared cache).
- Forcibly terminates the session by sending a stopSession, with a timeout of STN_GEM_TIMEOUT (if this is not set, a value of 5 minutes is used).
- If the gem does not stop within STN_GEM_TIMEOUT, the session is sent kill -TERM via the Stone or the remote netldi. Sessions are never sent a kill -9, to avoid the risk of fatal stuck spin locks.

If STN_GEM_LOSTOT_TIMEOUT = -1, then the session is not terminated, either via stopSession or kill. When the STN_GEM_ABORT_TIMEOUT limit expires, the stone immediately retracts the session's commit record and poisons its cache slot (the session will get a fatal error if it references the shared cache). Sessions logged in as Nameless are also handled in this way, regardless of the configuration settings used.

Collection Optimizations

A new instance method `removeKey:otherwise:` has been added to AbstractDictionary, with optimized versions that avoid the use of blocks added to KeyValueDictionary, GsMethodDictionary, AbstractCollisionBucket, IdentityCollisionBucket, IdentitySoftCollisionBucket, and SoftCollisionBucket.

A new instance method `inject:keysAndValuesInto:` has been added to Dictionary, KeyValueDictionary, GsMethodDictionary, IdentityDictionary, KeySoftValueDictionary, CanonStringDict, ObsoleteDictionary, ObsoleteSymbolListDictionary, RcKeyValueDictionary, and CanonStringBucket.

`AbstractDictionary >> addAll:` has been updated to use `inject:keysAndValuesInto:` if argument is an AbstractDictionary.

A new instance method `removeIdentical:otherwise:` has been added to SortedCollection.

`SortedCollection >> withAll:` and `SortedCollection >> resort` now use a merge sort for sizes larger than 2000. Also, the methods `SortedCollection >> select:` and `SortedCollection >> reject:` have had optimizations.

Topaz Changes

OUTPUT PUSHNEW behavior change

The semantics of this command has been changed. Now, up to 100 versions of the file are created (rather than 999, as previously). Once 100 files have been created, the oldest version will be overwritten. This is in response to bug #37101; see "Topaz OUTPUT PUSHNEW problems after 999 files" on page 1-9.

New commands

The following commands have been added to the topaz environment:

DOWN [<*anInteger*>**]**

DOWN -- Go down one activation in currently selected stack, and display the activation thus selected.

DOWN <*anInteger*> -- Go down anInteger activations and display the activation thus selected. See also STACK DOWN

UP [<*anInteger*>**]**

UP -- Go up one activation in currently selected stack, and display the activation thus selected.

UP <*anInteger*> -- Go up anInteger activations and display the activation thus selected. See also STACK UP

WHERE [<anInteger>**]**

WHERE -- Display current stack, one line per activation, same as STK.

WHERE <*anInteger*> -- Display anInteger activations of the current stack, starting at the current activation, one line per activation.

FRAME [<*anInteger*>**]**

FRAME displays current activation of current stack, without temps

FRAME <*anInteger*> go to specified activation in current stack, and make the specified activation the current activation and display that activation with temps. Similar to "STACK SCOPE <*anInteger*>".

Linux system stats now available

Operating System statistics are now available on Linux and on AIX when using statmonitor and VSD.

Logging changes

Improved formatting by addAllToStoneLog:

Sending the method `addAllToStoneLog:` now automatically adds a timestamp and additional empty lines before and after.

Timestamps now include milliseconds

Some message printed to the Stone log, and gem login messages, now include milliseconds as part of the timestamp.

Optimization of or: and and:

Previous version and products optimized `_and:` and `_or:` to simplify the block, while `and:` and `or:` were unoptimized message sends. Now, `and:` and `or:` are optimized the same as `_and:` and `_or:`; it's no longer useful to use the underscore versions of these methods.

New Errors

RT_ERR_COMMIT_DISALLOWED_UNTIL_ABORT 2424

Commits are disallowed because an attempt was made to continue a transaction after accessing an RcQueue. The session must abort.

RT_ERR_RC_UPDATE_DISALLOWED 2500

An attempt was made to update an RC object in a continued transaction. The transaction must abort.

RT_ERR_ROLLBACK_DL_FAIL 2425

Rollback of objects in a specific dirty list not allowed after a failed commit.

GS_ERR_SIGTERM 4052

Gem process terminated because SIGTERM was received. See gem log file for details. RPC GCI client will typically receive a network error 4137

Macintosh on Intel client libraries (unsupported)

Client libraries allowing connection from client Smalltalk running on the Macintosh platform on Intel based chips is now available as an unsupported goodie.

Seaside and GLASS

GemStone's support for Seaside and Monticello have undergone considerable development and changes, but are still not finalized and fully supported in this release. For more information, see:

```
http://seaside.gemstone.com/
```

The location of the distribution is now in the GemStone/S 64 Bit installation directory, in `$GEMSTONE/seaside/`.

To avoid seaside-related errors, multiple dirty lists have been implemented.

Multiple Dirty Lists

Multiple dirty lists are now supported. The following methods have been added:

```
dirtyListId
setDirtyList:
rollbackDirtyList:
enumerateDirtyList:
```

Bugs Fixed

The following bugs in GemStone/S 64 Bit 2.2.1 have been fixed in GemStone/S 64 Bit 2.2.2.

Topaz does not start on Windows

On Windows, topaz failed to start, with a gethostbyname() error. (#36884)

Conversion problems with Segments

During conversion from previous products or versions, it was possible for repositories with specific conversion histories to encounter Segment errors during conversion. (#37168)

In addition, large objects did not have their Segments handled properly during conversion, causing possible authorization errors following conversion. (#37357)

Keyfile CPU Affinity failures if running with more CPUs

A keyfile feature added in 2.2.x, CPU Affinity, restricts Stone startup to machines with that many or fewer CPUs. A logic error caused keyfile failures when fewer than the allowed number of CPUs were present. (37369)

Gems not terminated quickly enough on lost remote cache

If the stone or page manager loses connection to a remote cache, it terminates the logged in sessions attached to that cache. Under some circumstances it may not have terminated these sessions aggressively enough, with the risk that these session may have continued to operate on a stale cache. (#36849)

Gem not responsive to stopSession when hung sending GCI result

If a gem became hung waiting on a blocked socket when sending GCI results, it did not respond to stopSession or kill -TERM, and could cause commit Record backlog and related problems. (#37326)

Idle AIO page servers

Depending on the way AIO page servers and Free Frame page servers were configured, the distribution of page servers over the shared page cache regions allowed a situation in which half of the configured number of AIO page servers were idle. (#37418)

Execution of user actions failed during initialization

Executing client side user actions failed during session initialization. (#36841)

VSD display problems on Linux

Linux only

VSD running on Linux had multiple display problems, most noticeable the y-axis labels were incorrect by an amount equivalent to one labelling increment. (#35143)

DoubleByteString withAll: failed for large argument

If the argument to `DoubleByteString >> withAll:` was a large string (more than 8097 characters), this method would fail with an error that the object was corrupt. This error would disable further commits, so work done previously in the session was lost. (#36985)

Lock requests caused stone pause

When a request for a lock is made, the stone examines commit records later then the requesting session's view. Much of this work has been moved to the requesting session, and the processing has been made more efficient (#36831)

Missing details in timeToRestoreTo:

When using `Repository >> timeToRestoreTo:` to restore a repository to a specific time, the argument is localized using the current Repository TimeZone. information on transaction logs is localized differently, which could cause confusion when apparently valid restore to times returned errors. The error message now provides full details to allow easier diagnosis. (#36390, #36093)

OtherPassword privilege was required to modify own transient symbol list

Sessions have a transient SymbolList as well as the permanent SymbolList kept in the UserProfile. Privileges were checked correctly for the permanent SymbolList, but for the session's current transient SymbolList, OtherPassword privilege was incorrectly required. (#36822)

Application write locks 3 - 10 unavailable

Application write locks were increased from 2 to 10 in version 2.2.1. However, an internal an incorrect internal check prevented use of the added write locks. (#36702)

Create/remove indexes with uncommitted objects may result in errors

If indexes are created on uncommitted collections containing uncommitted objects, queries on these collections will create results that include OOPs for the objects. If the results were saved, but the collection or any objects in the collection were removed without committing or the commit failed, this may have resulted in incorrect answers or object does not exist errors. (#36892)

DST starts and ends a second late

Daylight Savings Time begins and ends at specified times, such as 2:00 am. GemStone was not applying the offset until the second following the specified time. (#37080)

Exception >> signal resumed on handled return

In the case where an signaled exception was handled by an exception handler, the exception was incorrectly being resumed, rather than returning to normal execution. This did not conform to the ANSI specification. Now, an exception handled by a handler block returns the value of the handler block (implicit 'ex return'). Resumable exceptions that do not have matching handlers are unchanged; they resume following execution of the default action (implicit 'ex resume'). (#36980)

For example, the following code used to assign 'bad' to x, but now assigns 'good' to x.

```
x := [
   Notification signal.
   'bad'.
] on: Notification do: [:ex |
   'good'.
].
```

Message details missing on Exception raised via error:

Using `Object >> error:` to raise an ANSI Exception did not set the Exception's description to the argument text. (#37219)

Passivate/activate sensitive to Locale changes

The passivation mechanism used the current decimal separator, resulting in errors if Floating point values were activated in a different Locale than the one in which they were passivated. (#36627)

PassiveObject obsolete OOP size limit

`PassiveObject >> readNamedIV` included obsolete code with separate handling for OOPs greater than 1 billion. (#37274)

Topaz OUTPUT PUSHNEW problems after 999 files

The command option OUTPUT PUSHNEW should create a new file, or return an error if files with the given name numbered up to 1000 already exist. However, the behavior when files 1..999 already exist was incorrect and varied by platform; either the file was not created, but no error returned, or it resulted in a coredump. (#37101)

This is fixed by limiting the number of files to 100, and overwriting older files when the limit is reached; see "OUTPUT PUSHNEW behavior change" on page 1-5.

raisedTo: failed for some results

For negative arguments, where the resulting denominator was a LargePositiveInteger close to the Integer range, this method would return a corrupt object error. This error would disable further commits, so work done previously in the session was lost. (#36939)

listInstances:toDirectory: failures when target final slash character missing

If a final slash was missing in the directory argument for the method `System >> listInstances:toDirectory:`, invalid file specification errors were likely. (#37100)

ByteArrays with DateTimes were not usable cross platform

ByteArray stored and read DateTime instances in native byte order, which meant the resulting ByteArrays could not be moved between Intel (Linux) and non-Intel platforms. (#37125)

The fix for this bug includes new methods to either read DateTimes from ByteArrays in the old native byte order, which is required if you have existing ByteArrays on Linux; or to store and read DateTimes from ByteArrays in the correct (always big endian) byte order. New protocol in this release also allows you to store with millisecond resolution. See “ByteArray methods for DateTimes now support cross-endian use” on page 1-2 for more information.

Risk of SIGSEGV on markSweep in low memory conditions

It is possible for internal memory markSweep operation to cause a gem SIGSEGV, if temporary object memory is heavily loaded and there are many dirty committed objects in pom-gem when the scavenge overflows and becomes a markSweep. (#37270)

methodsTooLargeFor64.topaz failures on iferr

The script `methodsTooLargeFor64.topaz` is provided to assist in the conversion process from GemStone/S or GemStone/S 2G to GemStone/S 64 Bit. This script included an abbreviation of the topaz command, which is no longer unambiguous following the enhanced topaz error handling functionality in version 2.2.1, causing failures. (#37179)

Chapter 2

GemStone C Statistics Interface

This appendix describes the GemStone C Statistics Interface (GCSI), a library of functions that allow your C application to collect GemStone statistics directly from the shared page cache without starting a database session.

Developing a GCSI Application

The command lines in this appendix assume that you have set the GEMSTONE environment variable to your GemStone installation directory.

Required header files

Your GCSI program must include the following header files:

- $GEMSTONE/include/shrpcstats.ht — Defines all cache statistics. (For a list of cache statistics, refer to the "Monitoring GemStone" chapter of the *GemStone/S 64 Bit System Administration Guide*.)
- $GEMSTONE/include/gcsi.hf — Prototypes for all GCSI functions.
- $GEMSTONE/include/gcsierr.ht — GCSI error numbers.

Your program must define a `main()` function somewhere.

The GCSI shared library

GemStone provides a shared library, $GEMSTONE/lib/libgcsi.so (on HP-UX, $GEMSTONE/lib/libgcsi.sl) that your program will load at runtime.

- Make sure that $GEMSTONE/lib is included in your LD_LIBRARY_PATH environment variable, so that the runtime loader can find the GCSI library. For example:

  ```
  export LD_LIBRARY_PATH=$GEMSTONE/lib:$LD_LIBRARY_PATH
  ```

- $GEMSTONE/lib/libgcsi.so is a multi-threaded library, so your program must also be compiled and linked as a multi-threaded program.

Compiling and linking

The $GEMSTONE/examples/gcsi/ directory includes the sample GCSI program `gsstat.cc`, along with a set of sample makefiles that show how to compile the sample GCSI program, using the compilers that are used to build the GemStone product.

NOTE

It may still be possible to build your program with another compiler (such as g++), so long as you specify the appropriate flags to enable multi-threading.

$GEMSTONE/lib/libgcsi.so (on HP-UX, $GEMSTONE/lib/libgcsi.sl) is a symbolic link that points to the actual shared library, which includes the GemStone version in the filename.

- When linking, *always* use the symbolic link libgcsi.so; do not link with the actual filename.

Whenever you upgrade to a new GemStone version, you must re-compile and re-link all your GCSI programs. This is because the internal structure of the shared cache may change from version to version. Assuming you've created a makefile, all you should need to do is change $GEMSTONE and rebuild.

Connecting to the shared page cache

The GCSI library allows your program to connect to a single GemStone shared page cache. Once the connection is made, a thread is started to monitor the cache and disconnect from it if the cache monitor process dies. This thread is needed to prevent your program from "holding on" to the shared cache after all other processes have detached from it. In this way, your program can safely sleep for a long time without preventing the operating system from freeing and recycling shared memory should the Stone be unexpectedly shut down.

The sample program

The sample program `gsstat` (in the $GEMSTONE/examples/gcsi directory) monitors a running GemStone repository by printing out a set of statistics at a regular interval that you specify. The program prints the following statistics:

- Sess — TotalSessionsCount; the total number of sessions currently logged in to the system.
- CR — CommitRecordCount; the number of outstanding commit records that are currently being maintained by the system.
- PNR — PagesNeedReclaimSize; the amount of reclamation work that is pending, that is, the backlog waiting for the GcGem reclaim task.
- PD — PossibleDeadSize; the number of objects previously marked as dereferenced in the repository, but for which sessions currently in a transaction might have created a reference in their object space.

- DNR — DeadNotReclaimedSize; the number of objects that have been determined to be dead (current sessions have indicated they do not have a reference to these objects) but have not yet been reclaimed.
- FP — The number of free pages in the Stone.
- OCS — OldestCrSession; the session ID of the session referencing the oldest commit record. Prints 0 if the oldest commit record is not referenced by any session, or if there is only one commit record.
- FF — FreeFrameCount; the number of unused page frames in the shared page cache.

To invoke `gsstat`, supply the name of a running Stone (or shared page cache, if running on a Gem server) and a time interval in seconds. For example:

```
% gsstat myStone 2
```

To stop the `gsstat` program and detach from the cache, issue a CTRL-C.

GCSI Data Types

The following C types are used by GCSI functions. The file `shrpcstats.ht` defines each of the GCSI types (shown in capital letters below). That file is in the `$GEMSTONE/include` directory.

ShrPcMonStatSType
: Shared page cache monitor statistics.

ShrPcStnStatSType
: Stone statistics.

ShrPcPgsvrStatSType
: Page server statistics.

ShrPcGemStatSType
: Gem session statistics.

ShrPcStatUnion
: The union of all four statistics structured types: shared page cache monitor, page server, Stone, and Gem.

ShrPcCommonStatSType
: Common statistics collected for all processes attached to the shared cache.

The Structure for Representing the GCSI Function Result

The structured type **GcsiResultSType** provides a C representation of the result of executing a GCSI function. This structure contains the following fields:

```
typedef struct {
   signed int processId;
   signed int sessionId;
   ShrPcCommonStatSType cmn;
   union ShrPcStatUnion u;
}    ShrPcStatsSType;

class GcsiResultSType {
public:
   char              vsdName[SHRPC_PROC_NAME_SIZE + 1];
   unsigned int      statType;
   ShrPcStatsSType   stats;
};
```

In addition, a set of C mnemonics support representation of the count of each process-specific structured type.

```
#define COMMON_STAT_COUNT
   (sizeof(ShrPcCommonStatSType)/sizeof(int))

#define SHRPC_STAT_COUNT (sizeof(ShrPcMonStatSType)/sizeof(int)
   + COMMON_STAT_COUNT)

#define GEM_STAT_COUNT (sizeof(ShrPcGemStatSType)/sizeof(int)
   + COMMON_STAT_COUNT)

#define PGSVR_STAT_COUNT
   (sizeof(ShrPcPgsvrStatSType)/sizeof(int) + COMMON_STAT_COUNT)

#define STN_STAT_COUNT (sizeof(ShrPcStnStatSType)/sizeof(int)
   + COMMON_STAT_COUNT)
```

GcsiAllStatsForMask

Get all cache statistics for a specified set of processes.

Syntax

int GcsiAllStatsForMask(*mask*, *result*, *resultSize*);
unsigned int *mask*;
GcsiResultSType * *result*;
int * *resultSize*;

Input Arguments

mask	Indicates what types or processes to collect statistics for.
result	Address of an array of kind GcsiResultSType where statistics will be stored.
resultSize	Pointer to an integer that indicates the size of the result in elements (not bytes). On return, indicates the number of that were stored into *result*. Indicates the maximum number of processes for which statistics can be returned.

Return Value

Returns 0 if successful; otherwise returns an error code, as defined in gcsierr.ht.

Example

Mask bits should be set by a bitwise OR of the desired process types. For example, to get statistics for the stone and Shared Page Cache Monitor:

```
unsigned int mask = SHRPC_MONITOR | SHRPC_STONE;
```

GcsiAttachSharedCache

Attach to the specified shared page cache.

Syntax

int GcsiAttachSharedCache(*fullCacheName*[], *errBuf*, *errBufSize*);

const char	*fullCacheName*;
char *	*errBuf*;
size_t	*errBufSize*;

Input Arguments

fullCacheName	Full name of the shared page cache, in the format *stoneName@stoneHostIpAddress*. To determine the full name of the shared cache, use the gslist -x utility.
errBuf	A buffer that will contain a string describing an error.
errBufSize	Size (in bytes) of *errBuf*.

Return Value

Returns 0 if successful; otherwise returns an error code, as defined in gcsierr.ht.

See Also

GcsiAttachSharedCacheForStone, page 2-7
GcsiDetachSharedCache, page 2-8

GcsiAttachSharedCacheForStone

Attaches this process to the specified shared page cache.

Syntax

int GcsiAttachSharedCacheForStone(*stoneName*[], *errBuf*, *errBufSize*);

const char	*stoneName*;
char *	*errBuf*;
size_t	*errBufSize*;

Input Arguments

stoneName	Name of the Stone process.
errBuf	A buffer that will contain a string describing an error.
errBufSize	Size (in bytes) of *errBuf*.

Return Value

Returns 0 if successful; otherwise returns an error code, as defined in gcsierr.ht.

Description

This function assumes that the cache name is `<stoneName>@<thisIpAddress>` where `thisIpAddress` is the IP address of the local machine. This function may fail if the host is multi-homed (has more than one network interface). In that case, use GcsiAttachSharedCache (page 2-6) to specify the full name of the shared cache.

See Also

GcsiAttachSharedCache, page 2-6
GcsiDetachSharedCache, page 2-8

GcsiDetachSharedCache

Detach from the shared page cache.

Syntax

int GcsiDetachSharedCache (void);

Return Value

Returns 0 if successful; otherwise returns an error code, as defined in gcsierr.ht.

See Also

GcsiAttachSharedCache, page 2-6
GcsiAttachSharedCacheForStone, page 2-7

GcsiInit (void)

Initialize the library. This function must be called before all other GCSI functions.

Syntax

GcsiInit(void);

GcsiFetchMaxProcessesInCache

Return the maximum number of processes that can be attached to this shared cache at any instant. The result may be used to allocate memory for a calls to the GcsiFetchStatsForAll* family of functions.

Syntax

int GcsiFetchMaxProcessesInCache(*maxProcesses*);
int * *maxProcesses*;

Input Arguments

maxProcesses	The maximum number of processes that can be attached to this shared cache at any instant.

Return Value

Returns 0 if successful; otherwise returns an error code, as defined in gcsierr.ht.

GcsiShrPcMonStatAtOffset

Get the SPC monitor cache statistic at the given byte offset within the ShrPcMonStatSType structure type.

Syntax

int GcsiShrPcMonStatAtOffset(*byteOffset*, *stat*);
size_t *byteOffset*;
unsigned int * *stat*;

Input Arguments

byteOffset	Offset (in bytes) of the desired statistic in the ShrPcStatUnion type.
stat	Value of the requested statistic.

Return Value

Returns 0 if successful; otherwise returns an error code, as defined in gcsierr.ht.

See Also

GcsiStnStatAtOffset, page 2-12

GcsiStnStatAtOffset

Get the Stone cache statistic at the given byte offset within the ShrPcStnStatSType structure type.

Syntax

int GcsiStnStatAtOffset(*byteOffset*, *stat*);
    size_t *byteOffset*;
    unsigned int * *stat*;

Input Arguments

byteOffset	Offset (in bytes) of the desired statistic in the ShrPcStatUnion type.
stat	Value of the requested statistic.

Return Value

Returns 0 if successful; otherwise returns an error code, as defined in gcsierr.ht.

See Also

GcsiShrPcMonStatAtOffset, page 2-11

GcsiStatsForGemSessionId

Get the cache statistics for the given Gem session id.

Syntax

int GcsiStatsForGemSessionId(*sessionId*, *result*);
int *sessionId*;
GcsiResultSType * *result*;

Input Arguments

sessionId	Session ID of the Gem for which statistics are requested.
result	Pointer to a GcsiResultSType structure.

Return Value

Returns 0 if successful; otherwise returns an error code, as defined in gcsierr.ht.

See Also

GcsiStatsForGemSessionWithName, page 2-14
GcsiStatsForPgsvrSessionId, page 2-15
GcsiStatsForProcessId, page 2-16
GcsiStatsForShrPcMon, page 2-17
GcsiStatsForStone, page 2-18

GcsiStatsForGemSessionWithName

Get the cache statistics for the first Gem in the cache with the given cache name.

Syntax

int GcsiStatsForGemSessionWithName(*gemName*, *result*);
    char * *gemName*;
    GcsiResultSType * *result*;

Input Arguments

gemName	The case-sensitive name of the Gem for which statistics are requested.
result	Pointer to a GcsiResultSType structure.

Return Value

Returns 0 if successful; otherwise returns an error code, as defined in gcsierr.ht.

See Also

GcsiStnStatAtOffset, page 2-12
GcsiStatsForPgsvrSessionId, page 2-15
GcsiStatsForProcessId, page 2-16
GcsiStatsForShrPcMon, page 2-17
GcsiStatsForStone, page 2-18

GcsiStatsForPgsvrSessionId

Get the cache statistics for the given page server with the given session id. Remote Gems have page servers on the Stone's cache that assume the same session ID as the remote Gem.

Syntax

int GcsiStatsForPgsvrSessionId(*sessionId, result*);
    int *sessionId*;
    GcsiResultSType * *result*;

Input Arguments

sessionId	Session ID of the page server for which statistics are requested.
result	Pointer to a GcsiResultSType structure.

Return Value

Returns 0 if successful; otherwise returns an error code, as defined in gcsierr.ht.

See Also

GcsiStatsForGemSessionId, page 2-13
GcsiStatsForGemSessionWithName, page 2-14
GcsiStatsForProcessId, page 2-16
GcsiStatsForShrPcMon, page 2-17
GcsiStatsForStone, page 2-18

GcsiStatsForProcessId

Get the cache statistics for the given process ID.

Syntax

int GcsiStatsForProcessId(*pid*, *result*);
int *pid*;
GcsiResultSType * *result*;

Input Arguments

pid	Process ID for which statistics are requested.
result	Pointer to a GcsiResultSType structure.

Return Value

Returns 0 if successful; otherwise returns an error code, as defined in gcsierr.ht.

See Also

GcsiStatsForGemSessionId, page 2-13
GcsiStatsForGemSessionWithName, page 2-14
GcsiStatsForPgsvrSessionId, page 2-15
GcsiStatsForShrPcMon, page 2-17
GcsiStatsForStone, page 2-18

GcsiStatsForShrPcMon

Get the cache statistics for the shared page cache monitor process for this shared page cache.

Syntax

int GcsiStatsForShrPcMon(*result*);
GcsiResultSType * *result*;

Input Arguments

result	Pointer to a GcsiResultSType structure.

Return Value

Returns 0 if successful; otherwise returns an error code, as defined in gcsierr.ht.

See Also

GcsiShrPcMonStatAtOffset, page 2-11
GcsiStatsForGemSessionId, page 2-13
GcsiStatsForGemSessionWithName, page 2-14
GcsiStatsForPgsvrSessionId, page 2-15
GcsiStatsForProcessId, page 2-16
GcsiStatsForStone, page 2-18

GcsiStatsForStone

Get the cache statistics for the Stone if there is a Stone attached to this shared page cache.

Syntax

int GcsiStatsForStone(*result*);
GcsiResultSType * *result*;

Input Arguments

result	Pointer to a GcsiResultSType structure.

Return Value

Returns 0 if successful; otherwise returns an error code, as defined in gcsierr.ht.

See Also

GcsiStnStatAtOffset, page 2-12
GcsiStatsForGemSessionId, page 2-13
GcsiStatsForGemSessionWithName, page 2-14
GcsiStatsForPgsvrSessionId, page 2-15
GcsiStatsForProcessId, page 2-16
GcsiStatsForStone, page 2-18

GCSI Errors

The following errors are defined for the GemStone C Statistics Interface.

Table 1 GCSI Errors

Error Name	Definition
GCSI_SUCCESS	The requested operation was successful.
GCSI_ERR_NO_INIT	GcsiInit() must be called before any other Gcsi functions.
GCSI_ERR_CACHE_ALREADY_ATTACHED	The requested shared cache is already attached.
GCSI_ERR_NOT_FOUND	The requested session or process was not found.
GCSI_ERR_BAD_ARG	An invalid argument was passed to a Gcsi function.
GCSI_ERR_CACHE_CONNECTION_SEVERED	The connection to the shared cache was lost.
GCSI_ERR_NO_STONE	Stone statistics were requested on a cache with no stone process.
GCSI_ERR_CACHE_NOT_ATTACHED	No shared page cache is currently attached.
GCSI_ERR_NO_MORE_HANDLES	The maximum number of shared caches are attached.
GCSI_ERR_CACHE_ATTACH_FAILED	The attempt to attach the shared cache has failed.
GCSI_ERR_WATCHER_THREAD_FAILED	The cache watcher thread could not be started.
GCSI_ERR_CACHE_WRONG_VERSION	The shared cache version does not match that of the libgcsi.so executable.