

---



# **GemStone/S 64 Bit<sup>TM</sup>** **Release Notes**

**Version 3.6.6**

May 2023



## 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.

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 © 1986-2023 GemTalk Systems LLC. All rights reserved by GemTalk Systems.

## PATENTS

GemStone software has been covered by U.S. Patent Number 6,256,637 "Transactional virtual machine architecture" (1998-2018), Patent Number 6,360,219 "Object queues with concurrent updating" (1998-2018), Patent Number 6,567,905 "Generational garbage collector with persistent object cache" (2001-2021), and Patent Number 6,681,226 "Selective pessimistic locking for a concurrently updateable database" (2001-2021).

## 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.

**UNIX** is a registered trademark of The Open Group.

**Intel** is a registered trademarks of Intel Corporation.

**Microsoft**, **Windows**, and **Windows Server** are registered trademarks of Microsoft Corporation.

**Linux** is a registered trademark of Linus Torvalds and others.

**Red Hat**, **Red Hat Enterprise Linux**, **RHEL**, and **CentOS** are trademarks or registered trademarks of Red Hat, Inc..

**AlmaLinux** is a trademark or registered trademark of AlmaLinux OS Foundation.

**Rocky Linux** is a trademark or registered trademark of Rocky Enterprise Software Foundation.

**Ubuntu** is a registered trademark of Canonical Ltd., Inc.

**AIX**, **Power**, **POWER**, **Power8**, **Power9**, and **VisualAge** are trademarks or registered trademarks of International Business Machines Corporation.

**Apple**, **Mac**, **macOS**, and **Macintosh** are trademarks of Apple Inc.

**Instantiations** is a registered trademarks of Instantiations, Inc.

**CINCOM**, **Cincom Smalltalk**, and **VisualWorks** are trademarks or registered trademarks of Cincom Systems, Inc.

**Raspberry Pi** is a trademark of the Raspberry Pi Foundation.

**RabbitMQ** is a trademark of VMware, Inc.

**Prometheus** is a registered trademark of The Linux Foundation.

**Grafana** is a registered trademark of Raintank, Inc. dba Grafana Labs.

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 LLC

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 3.6.6 release. Read these release notes carefully before you begin installation, upgrade, or development with this release.

No separate Installation Guide is provided with this release. For instructions on installing GemStone/S 64 Bit version 3.6.6, or upgrading or converting from previous products or versions, see the Installation Guide for version 3.6.2.

## 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 LLC, previously GemStone Systems, Inc. and a division of VMware, Inc.

## Technical Support

### Support Website

**[gemtalksystems.com](http://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.
- ▶ **Supplemental Documentation and TechTips**, providing information and instructions that are not in the regular documentation.
- ▶ **Compatibility matrices**, listing supported platforms for GemTalk product versions.

We recommend checking this site on a regular basis for the latest updates.

## Help Requests

GemTalk Technical Support is limited to customers with current support contracts. Requests for technical assistance may be submitted online (including by email), or by telephone. We recommend you use telephone contact only for urgent 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](http://techsupport.gemtalksystems.com)

**Email:** [techsupport@gemtalksystems.com](mailto:techsupport@gemtalksystems.com)

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

Please include the following, in addition to a description of the issue:

- ▶ The versions of GemStone/S 64 Bit and of all related GemTalk products, and of any other related products, such as client Smalltalk products, and the operating system and version you are using.
- ▶ Exact error message received, if any, including log files and statmonitor data 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. Release Notes for 3.6.6*

Overview . . . . .	9
Supported Platforms . . . . .	9
Platforms for Version 3.6.6 . . . . .	9
GemBuilder for Smalltalk (GBS) Versions . . . . .	10
VSD Version. . . . .	10
Rowan . . . . .	10
Open Source Library Versions . . . . .	10
Changes in this version . . . . .	11
Change in objectAudit behavior . . . . .	11
Version mismatch warnings. . . . .	11
Cache warming during upgrade . . . . .	11
AbstractDictionary >> at:ifPresent: restored . . . . .	12
String compare to ByteArray now returns false for matching bytes . . . . .	12
Time support for microseconds. . . . .	12
copydbf -i for backups now reports if in partial logging mode . . . . .	12
largememorypages improved and usable for remote caches . . . . .	12
stopnetldi performance improved . . . . .	12
Linux transparent huge pages not used as expected . . . . .	13
Topaz strings containing &nbsp; now print hex value rather than space . . . . .	13
Failure to get a stack trace for GEM_HALT_ON_ERROR values. . . . .	13
STN_GEM_TIMEOUT default login timeout reduced . . . . .	13
Determining status of cache warming . . . . .	13
Statmon data file header timestamp in UTC . . . . .	13
SharedCounters and PersistentCounters renumbered . . . . .	14
Removed methods . . . . .	14
Hot Standby Improvements and Fixes. . . . .	14
commitRestore now also stops continuous restore . . . . .	14
Added methods. . . . .	14

Upgrade via hot standby . . . . .	15
Hot standby bugs fixed. . . . .	16
Hot standby logreceiver errors if there are other logreceivers for the same logsender . . . . .	16
Problems after circular failOverToSlave . . . . .	16
Starting hotstandby continuous restore with obsolete tranlogs may crash	17
Bugs Fixed . . . . .	18
Gem crash recovery may cause Stone to SEGV . . . . .	18
Shrpcmon crash during crashed client frame lock recovery . . . . .	18
Protocol error on incomplete LGC_PAD_N read over socket . . . . .	18
TransactionBacklog signaled incorrectly . . . . .	18
listInstances may miss objects . . . . .	18
Risk of out of memory in voteNotDead in busy system . . . . .	18
Race condition results in Gem SEGV in copyFrom:to: . . . . .	18
Gems in login during stone shutdown may fail to exit . . . . .	18
Gems may be slow to shutdown due to memory checks . . . . .	19
Stone restart recovery may not correctly recover new epoch objects and write set union . . . . .	19
Epoch GC can declare a class as dead while an instance remained . . . . .	19
Multithreaded scan operations clear all Gem session stats . . . . .	19
Some Time stats incorrectly zero . . . . .	19
Unnecessary commit conflicts . . . . .	19
With Indexing operations. . . . .	19
On RciIdentitySet/Bag . . . . .	19
Slow timeout failure on login when network connection table is full . . . . .	19
Cache warming could hang . . . . .	20
Issues with thread-safe nbLogin/nbLogout . . . . .	20
After nbLogin, GsTsExternalSession wait methods SIGSEGVed or errored	20
Memory leak in GciTsNbLogin . . . . .	20
GsTsNbLogout could signal Network error . . . . .	20
Upgrade Issues . . . . .	20
Nonstandard decimalPoint Locale with GsPackagePolicy . . . . .	20
Upgrade errors on changes in definitions in ObsoleteClasses . . . . .	20
Backup and restore failed to check OOP upper bound . . . . .	20
SymbolGem printed excessive messages to its log on GC . . . . .	21
topaz set cachename did not work for linked topaz. . . . .	21
SecurityError on objectSecurityPolicy: for a class with no instance variables . .	21
Issues related to Numeric operations . . . . .	21
Division by SmallInteger minimumValue threw error . . . . .	21
Number parseLiterals:exponent: handling nil character argument . . . . .	21
ScaledDecimal, Fraction >> asFloat could return incorrect results. . . . .	21
Login log did not record failed logins. . . . .	21
GsFile primitives not interrupted by SIGTERM . . . . .	21
Improved reporting on client disconnect errors . . . . .	22
Checking for vote state incorrect . . . . .	22
Unicode Compares causes strings to not be returned in ExportedDirtyList . . .	22
performOnServer: failed for configurations with client s-bit set. . . . .	22

Object >> subclassResponsibility error was not reported correctly. . . . . 22  
Transaction log debug level change . . . . . 22  
Private method could SEGV. . . . . 22





# Release Notes for 3.6.6

---

## Overview

GemStone/S 64 Bit™ 3.6.6 is a new version of the GemStone/S 64 Bit object server. Version 3.6.6 provides new features and fixes for a number of significant bugs. We recommend everyone using or planning to use GemStone/S 64 Bit upgrade to this new version.

These Release Notes include changes between the previous version of GemStone/S 64 Bit, v3.6.5, and v3.6.6. If you are upgrading from a version prior to 3.6.5, review the release notes for each intermediate release to see the full set of changes.

The Installation Guide has not been updated for this release. For installation, upgrade and conversion instructions, use the Installation Guide for version 3.6.2.

## Supported Platforms

### Platforms for Version 3.6.6

GemStone/S 64 Bit version 3.6.6 is supported on the following platforms:

- ▶ Red Hat Enterprise Linux Server, CentOS Linux, Rocky Linux, and AlmaLinux 7.9, 8.7, and 9.1; Ubuntu 20.04 and 22.04; all on x86.  
Ubuntu 20.04 on ARM (Ubuntu on ARM is supported for development only)  
GemStone performs testing on a mixture of Red Hat, CentOS, Rocky Linux and AlmaLinux servers; these are all fully certified platforms. Any reference to Red Hat applies to all these distributions.
- ▶ AIX 7.1 and 7.2
- ▶ macOS 13.1 (Ventura) with Darwin 22.2.0 kernel, and macOS 12.6 (Monterey) with Darwin 21.6.0 kernel on x86; and macOS 11.6 (Big Sur) with Darwin 20.6.0 kernel on Apple silicon.  
(Mac is supported for development only)

Distributions for Solaris/x86 and Solaris/SPARC are no longer available.

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

## GemBuilder for Smalltalk (GBS) Versions

The following versions of GBS are supported with GemStone/S 64 Bit version 3.6.6:

### GBS/VW version 8.6

<b>VisualWorks 9.1.1 32-bit and 64-bit</b>
<ul style="list-style-type: none"> <li>▶ Windows 10</li> <li>▶ RedHat ES 7.9 and 8.7; Ubuntu 20.04</li> </ul>

### GBS/VA version 5.4.6

VAST Platform 11.0.1	VAST Platform 10.0.2	VA Smalltalk 8.6.3
▶ Windows Server 2016 and Windows 10	▶ Windows Server 2016 and Windows 10	▶ Windows Server 2016 and Windows 10

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

## VSD Version

The GemStone/S 64 Bit v3.6.6 distribution includes VSD version 5.6. The previous version of GemStone/S 64 Bit, v3.6.5, included VSD v5.5.4. VSD version 5.6 includes several bug fixes and new features. For details on the changes, see the [Release Notes for VSD v5.6](#).

VSD v5.6 is included with the GemStone distribution, and can also be downloaded as a separate product. For details or to download, go to <https://gemtalksystems.com/vsd/>.

## Rowan

The GemStone/S v3.6.6 distribution includes Rowan v 2.3.1.

## Open Source Library Versions

The version of OpenSSL has been updated to 1.1.1t

The version of MIT Kerberos has been updated to 1.19.4

## Changes in this version

### Change in objectAudit behavior

The previous implementation of `Repository >> objectAudit` could in some cases miss problems (#50419). The earlier implementation aborted as needed after an initial scan, to avoid creating a commit record backlog. While it could safely run while the repository is in use, it could not perform a definitive scan to ensure there were no invalid references.

As of v3.6.6, `objectAudit` will now run in transaction. While running in transaction allows `objectAudit` to perform a complete audit, it does risk a commit record backlog if it is run on a system that is in use. The earlier behavior is available using new methods.

**If you use `objectAudit` on active systems, you should review your scripts or monitor for commit record backlog.**

The following methods provide the old `objectAudit` behavior (aborting if necessary during the scan), now described as partial.

`fastObjectAuditPartial`

Similar to `objectAuditPartial` except executes as fast as possible.

`objectAuditPartial`

Similar to `objectAudit`, except that it aborts after processing the scavengable pages and audits the rest of the repository aborting as needed to avoid causing a commit record backlog. Does not check all object table references for validity, but will detect most references to non-existent objects.

`objectAuditPartialWithMaxThreads: numThreads`

Similar to `objectAuditPartial` except that it allows specifying the number of threads to use for the scan.

### Version mismatch warnings

Normally, you should always ensure that GemStone image and executables are the same version; that is, you should always run **upgradeImage** immediately after starting a stone using executables from a new version. However, in the hotstandby upgrade process, you may temporarily run the slave system on a mismatched system.

In previously releases, no error nor warning was raised at login time if the first two version digits matched, e.g. there was no error or warning message on login if 3.6.5 executables were started with a 3.6.4 image; errors would occur only over major releases, e.g. a 3.5.x image and 3.6.x executables.

Now, this error is always signalled to avoid accidentally using the wrong executable/image combination. This is a fatal error for any user other than `SystemUser`. With a mismatch between executables and image version, you may only login as `SystemUser`.

### Cache warming during upgrade

During the upgrade process, after the stone has started but before **upgradeImage** has completed, it is helpful for cache warmers to be able to log in, since cache warming may improve upgrade performance. It is now allowed for cache warmers to login before `upgradeImage` is complete, for major version upgrades as well as minor version upgrades.

## AbstractDictionary >> at:ifPresent: restored

The method `AbstractDictionary >> at:ifPresent:` was added in v3.5 with code to support X509 logins, and inadvertently removed from the image during code cleanup for v3.6. This method has been restored.

## String compare to ByteArray now returns false for matching bytes

If a `String` is compared to a `ByteArray` using `=`, and they both contain the same bytes (e.g. `' ' = #[32]`), in previous releases, the comparison returned `true`; while a similar comparison between a `ByteArray` and a `String` (e.g. `#[32] = ' '`) returned `false`. (#49031) Both cases now return `false` for this comparison.

## Time support for microseconds

In previous releases, `Time` has included microsecond resolution; however, methods to support using microsecond times were limited.

The following methods have been added:

`asStringUs`

Returns a `String` that expresses the receiver in local time in the format `HH:MM:SS.ssssss` where `sss` are microseconds.

`addMicroseconds: anInteger`

Returns a `Time` that describes a time of day `anInteger` microseconds later than that of the receiver.

`subtractMicroseconds: anInteger`

Returns a `Time` that describes a time of day `anInteger` milliseconds earlier than that of the receiver.

## copydbf -i for backups now reports if in partial logging mode

`copydbf -i` on a programmatic backup now includes an additional line if the repository was in partial logging mode or has `tranlogs` set to `/dev/null`:

```
Repository in partial logging mode or writing tranlogs to
/dev/null
```

## largememorypages improved and usable for remote caches

The calculations of the number of large memory pages was not entirely accurate, and could produce too low values for very large caches. (#50216).

In addition, the `-r` option has been added to `largememorypages`, which calculates the page requirements for a remote cache, which is somewhat smaller than for the Stone's cache.

## stopnetldi performance improved

The multithreaded `stopnetldi` utility did not poll and wait efficiently between threads, and is now much faster.

## Linux transparent huge pages not used as expected

The handling of the madvise kernel parameters setting from GemStone code was insufficient to reliably allow a temporary object cache to use transparent huge pages. This has been adjusted so that transparent huge pages will be used with TOC sizes of 400MB or greater, if `/sys/kernel/mm/transparent_hugepage/enabled` is set to "madvise."

## Topaz strings containing `&nbsp;` now print hex value rather than space

Character codePoint: 16rA0 is the non-blocking space `&nbsp;`. If this character is included in a string, previously topaz displayed this the same as a standard space, code point 16r20. Now, non-blocking spaces will be displayed as `'\ua0'` to avoid confusion.

White space characters outside of the ASCII range, which includes codePoint 16rA0 and others, are not considered whitespace by the GemStone compiler.

## Failure to get a stack trace for GEM\_HALT\_ON\_ERROR values

In some cases, GEM\_HALT\_ON\_ERROR may be set in a Gem's configuration, but when the error occurs, the gem log reports the error but does not prints stacks.

Client logins invoke the GCI login function `GciLoginEx_()`, which includes an argument, `haltOnError`. If this is set to 0 it disables the GEM\_HALT\_ON\_ERROR from the gem configuration; it should be set to -1 to specify using the configuration value.

The `GciLogin()` call incorrectly invoked `GciLoginEx_()` with a 0. (#50275)

GCI applications or client applications such as GBS and Jadeite may be calling `GciLoginEx_()` with a 0 for the `haltOnError` argument. This requires a change in the application code.

## STN\_GEM\_TIMEOUT default login timeout reduced

If the value of STN\_GEM\_TIMEOUT is zero, the login timeout is now 1 minute rather than 5 minutes.

## Determining status of cache warming

The following methods have been added, which report the `SessionId` and other information for active cache warmers. This allows you to detect when startup cache warming is complete.

```
System class >> cacheWarmerSessions
Returns an Array of sessionIds of cache warmer sessions.
```

```
System class >> cacheWarmerSessionsReport
Returns a String describing cache warmer sessions.
```

## Statmon data file header timestamp in UTC

The timestamp printed in the statmonitor data file header is now formatted as ISO 8601, rather than the GemStone's legacy `DateTime` format.

## SharedCounters and PersistentCounters renumbered

SharedCounters (AppStats) and PersistentCounters are customer-usable cache statistics which are recorded directly in the cache, unlike SessionCacheStats. These are collected by specifying the **statmonitor -n** and **-B** flags, respectively.

In previous releases, the numbering of SharedCounters in smalltalk methods was off by one for the numbering in the statmonitor/VSD. This has been corrected; e.g., for an expression such as

```
System sharedCounter: 1 setValue: 0.
```

Previously, this statistic would have the name **sharedCounter2** (under the process AppStat) now it is **sharedCounter0001**.

The number of shared counters is defined in the configuration file, and defaults to 1900; the leading 0 allows correct sorting.

SharedCounters are now recorded correctly as 64 bit signed integers.

The offset for Persistent counters was correct, but these have also been reformatted for sorting; names that were previously similar to **PersistentCounter001** (under the process PersistentCounters) are now **PersistentCounter0001**. Since the maximum number of persistent counters is 1536, this ensures sorting in VSD will be correct.

## Removed methods

The following private methods have been removed:

```
IdentityBag >> _basicAdd:
IdentityBag >> _rcIncludes:
IdentityBag >> _rcIncludesValue:
Integer >> _floatParts
Repository >> _objectAuditWithMaxThreads:waitForLock:
    pageBufSize:percentCpuActiveLimit:csvFile:repair:
SmallInteger >> _floatParts
System class >> _sessionsReportExcluding:
```

## Hot Standby Improvements and Fixes

### commitRestore now also stops continuous restore

Previously, when a slave system was being converted into normal mode, you had to perform a `stopContinuousRestore`, and then login again to perform the `commitRestore`.

Now, sending `commitRestore` will stop continuous restore if it is running, and then perform the commit restore. It is unnecessary to perform a separate `stopContinuousRestore`.

## Added methods

The following methods have been added to make hot standby failover simpler.

```
Repository >> commitRestoreForFailoverAfterWaitingUpTo: seconds
    On a slave stone, waits up to seconds for the failover record from the master stone
```

to be replayed. If *seconds* is 0, this method checks for the fail over record once and does not block. If *seconds* is -1, this method waits forever for the fail over record. If the failover record is detected, then a `commitRestore` is executed and the session is terminated with a `RestoreLogSuccess` error (4048). If the failover record is not detected within the timeout, returns false.

`Repository >> failoverFromMasterFinished`

On a slave stone, returns a boolean indicating if the failover record from the master has been replayed. Returns true if the failover record has been replayed or false otherwise.

`Repository >> waitForFailoverFromMasterUpToSeconds: seconds`

On a slave stone, waits up to *seconds* for the fail over record from the master stone to be replayed. If *seconds* is 0, this method returns a status immediately and does not block. If *seconds* is -1, this method waits forever for the failover record. Returns true if the fail over record has been replayed or false if the timeout expires.

## Upgrade via hot standby

It is now possible to perform a GemStone upgrade of a hot standby system, by upgrading the slave, failing over to make this the master, and allowing the former master, new slave to be upgraded via transaction logs.

This does not require changes in the originating version, and this process can be used to upgrade from earlier versions to v3.6.6. However, some methods that make this process easier are not present in the earlier version, and thus may not be available at the point where they will be useful.

This upgrade process has been recently developed and may need refinement, and has not been verified in a multiple slave system. It has been tested for upgrade from 3.6.5 and 3.6.4 to 3.6.6.

To upgrade via hot standby:

1. Stop the slave system's logreceiver and the slave stone. Restart both using the v3.6.6 binaries, and restart the continuous restore. Do not perform the **upgradeimage** yet.

The slave will continue to restore transaction from the master, although logins will report version mismatch errors. Login as `SystemUser` is allowed. These errors are temporary until the upgrade process is complete.

You should not attempt to execute other code on the slave; the version mismatch means that methods (other than those required for running as a slave) may fail unexpectedly. This include user errors; if for example you enter an incorrect path for continuous restore, the error message cannot be handled or provide details.

2. On the master, which is still on the original version, perform `failOverToSlave`. This stops further commits, checkpoints, and puts the now former master into restore mode.

Commits are now disallowed on the master.

3. Wait for the master's failover transaction to be replayed on the slave. You can determine this using `SystemRepository restoreStatusInfo at: 13`, which

returns a non-zero timestamp when the failover transaction is replayed. On the slave system, execute:

```
[ 0 == (SystemRepository restoreStatusInfo at: 13) ]
  whileTrue: [ System _sleepMs: 500 ].
```

In future upgrades, `Repository >>`

`waitForFailoverFromMasterUpToSeconds:` can be used here; but since the slave's image has not yet been upgraded, this method is not available.

Once the failover transaction has been replayed, stop the master stone and the logsender. The former-master stone may be left running to service read-only operations, until Step 6.

4. On the slave, stop the logreceiver, and perform the `stopContinuousRestore` and do `commitRestore`. This makes the now former slave into the new master.
5. On the new master (the former slave), perform the `upgradeImage` and any other required upgrade steps.

The new master is now available for logins.

Start the logsender on this new master.

6. On the former master, now slave system restart the stone and logreceiver using the 3.6.6 binaries, and start continuous restore (`continuousRestoreFromArchiveLogs:`). Do not perform `upgradeImage`.

As the transactions coming from the new master are replayed on the new slave, the new slave will be upgraded to v3.6.6. There will be a lag after the transactions are restored before next checkpoint completes, and the version is updated. Until this time, logins to the new slave system will continue to see a version mismatch error.

## Hot standby bugs fixed

### Hot standby logreceiver errors if there are other logreceivers for the same logsender

A master stone's logsender did not correctly service multiple slave systems logreceivers. If multiple logreceivers were connected to a single logsender, the data sent to the logreceivers was out of sync, and the logreceivers reported validation errors. (#50064)

### Problems after circular failOverToSlave

After a `failOverToSlave` from NodeA to NodeB and a subsequent second `failOverToSlave` from NodeB back to NodeA, there were issues on the new master NodeA.

NodeA's transient free oop list could have contained the oops of committed objects, which could result in corruption; this issue was cleared by a stone restart. (#50147)

The master stone NodeA was also left with commits suspended, which required executing `resumeCommits`. (#50155)



### **Starting hotstandby continuous restore with obsolete tranlogs may crash**

When the STN\_TRAN\_LOG\_DIRECTORIES directory of the slave system in a hotstandby system contains obsolete tranlogs, starting continuous restore may crash when attempting to read logs. In addition, now if a previously running continuous restore (hotstandby) has stopped due to a failure, the restoreStatus includes "continuous restore failed". (#49863).

## Bugs Fixed

### Gem crash recovery may cause Stone to SEGV

When a Gem crashes, the Shrpcmon performs recovery to clean up spin locks. In versions before v3.6.5, the Shrpcmon process could SEGV if an address into the process table was out of range (bug #49988, fixed in v3.6.5). Additional codepaths outside of the Shrpcmon, such as the Stone, which could also SEGV on an attempt to access in invalid offset; these are fixed in this version. (#50005)

### Shrpcmon crash during crashed client frame lock recovery

The shared page cache monitor frame lock recovery may crash under some circumstances with ongoing page reads, resulting in the message "Freezing shared page cache" (#50082)

### Protocol error on incomplete LGC\_PAD\_N read over socket

Gems may encounter a GCI protocol error, due to socket read returning bytes that contain an incomplete LGC\_PAD\_N packet (#50045)

### TransactionBacklog signaled incorrectly

When the Gem configuration parameter #GemAutoServiceSigAbort is set to false, sessions could still receive signal #3007/TransactionBacklog. (#50142)

### listInstances may miss objects

The multithreaded scan operation that executes listInstances scan may miss objects, if one or more objects are shadowed (that is, they are updated during the scan) and the scan aborts its view. (#50407)

### Risk of out of memory in voteNotDead in busy system

During the voteNotDead stage of garbage collection, the revote requires keeping a closure of candidate dead objects; this could grow to consume excessive memory on a system that had many changes, such as after upgrade or index rebuild. (#50362)

### Race condition results in Gem SEGV in copyFrom:to:

The primitive that supports Array and OrderedCollection >> copyFrom:to: (prim 817), contains an unsafe object allocation. There is a race condition, if a scavenge is triggered by an object faulting into memory after the new object is allocated; the new object could be initialized to zero rather than OOP\_NIL, causing a Gem SEGV (#50277)

### Gems in login during stone shutdown may fail to exit

A timing condition when the Stone shuts down during a window in the Gem login process may leave the Gems alive after the Stone is shut down. (#50204)

## **Gems may be slow to shutdown due to memory checks**

With very large temporary object memory, a Gem may take considerable time to verify memory before shutting down. The verification is not critical, and is now skipped in the standard (fast) environment. (#50314)

## **Stone restart recovery may not correctly recover new epoch objects and write set union**

After a crash, it was possible for restart to have incorrect state for new epoch objects and write set union. (#50470)

## **Epoch GC can declare a class as dead while an instance remained**

There is a code path in which a class can be found dead by MFC or epoch, when there is only one object referencing that class, due to non-thread-safe update. (#50460)

## **Multithreaded scan operations clear all Gem session stats**

The multithreaded scan operations update values for a number of session stats; it was unnecessarily also clearing cache statistics that it did not update. (#40411)

To see session statistics that are in use for a specific scan operation, open the statistics in VSD and apply the relevant aliases; this provides internally-useful labels for stats that are updated by that scan.

## **Some Time stats incorrectly zero**

The statistics TimeInUpdateUnionsCommit, TimeProcessingCommit, and TimeStoneCommit may be incorrectly understated or reported as zero on a fast system; these were rounded to ms prior to summing, and thus lost elapsed times in microsecond ranges. (#50422)

## **Unnecessary commit conflicts**

### **With Indexing operations**

BTreePlus indexes may encounter commit conflicts from internal structures; that is, operations that update an indexed collection and should succeed fail with a commit conflict. (#50146)

### **On RcIdentitySet/Bag**

If internal leaf balancing code executes during an add operation, the RcRead sets will not be correct and could result in commit conflicts. (#50220)

## **Slow timeout failure on login when network connection table is full**

If the network connection table was full, including with sessions that were zombies or in login or logout status, new sessions attempting to log in could fail login with the error (depending on version), "the maximum number of users is already logged in". Now, the Stone is more aggressive about processing zombies when the table is full. (#49927)

## Cache warming could hang

There is a race condition between the multiple threads of the cache warmer, which could result in a state flag being set incorrectly, such that all threads were waiting and no progress was made. (#50258)

## Issues with thread-safe nbLogin/nbLogout

### After nbLogin, GsTsExternalSession wait methods SIGSEGVed or errored

After invoking `GsTsExternalSession >> nbLogin`, either `waitForReadReady`, `waitForReadReadyTimeout:`, `waitForResult`, or `waitForResultForSeconds:`, should be called, to detect when the login has completed.

`waitForResult` resulted in a SIGSEGV.

`waitForReadReady`, `waitForReadReadyTimeout:`, `waitForResultForSeconds:` errored with call not in progress, but did not crash. (#50203)

### Memory leak in GciTsNbLogin

The function `GciTsNbLogin`, which is invoked when using `GsTsExternalSession`, contains a small memory leak. (#50215)

### GsTsNbLogout could signal Network error

`GsTsNbLogout` could signal error 4137, `#netErr`. (#50425)

## Upgrade Issues

### Nonstandard decimalPoint Locale with GsPackagePolicy

If a repository has `GsPackagePolicy` enabled (generally a Seaside or `GsDevKit` application), and the `Locale` does not specify a period (US-style) `decimalPoint`, `previousVersion` testing failed, and `upgradeImage` reported an error. (#50063)

### Upgrade errors on changes in definitions in ObsoleteClasses

The `ObsoleteClasses` dictionary contains classes that are no longer part of the GemStone kernel, but that are retained so that upgraded images that contain objects of these classes continue to work. These classes were unnecessarily being updated during `upgradeImage`, which for some upgrade paths produced errors in upgrade. (#50327)

## Backup and restore failed to check OOP upper bound

It was possible for programmatic backup, possibly due to hardware issues, to produce a backup file that included OOPs that were higher than the OOP high water for the backup file (which is stored in the backup when the backup is initiated). Restoring this backup produced errors and potentially crashed or corrupted the Gem and/or Stone. Now, OOPs higher than the high water mark are detected and rejected. (#50081)

## SymbolGem printed excessive messages to its log on GC

The SymbolGem's log file could excessively bloat with log messages, particularly commit messages related to symbol garbage collection. (#50112)

## topaz set cachename did not work for linked topaz

Setting the name recorded in statmonitor data using the topaz command `set cachename`, in linked topaz, did not actually update the name of the process as recorded in statmonitor. (#50053)

## SecurityError on objectSecurityPolicy: for a class with no instance variables

If a class has no instance variables, `instVarNames` is an empty Array, which is canonicalized to the object with oop 233217, which is in `SystemObjectSecurityPolicy`. If this class is sent `objectSecurityPolicy`, the code was incorrectly attempting to reassign the object security policy of this empty Array, which reported a `SecurityError`. (#49921)

## Issues related to Numeric operations

### Division by SmallInteger minimumValue threw error

Dividing an Integer by `SmallInteger minimumValue` resulted in an `IntenalError`, `StoreSmallInt out of range`. (#50111)

### Number parseLiterals:exponent: handling nil character argument

If the `parseLiterals:argument` to `Number class >> parseLiterals:exponent:` was nil, rather than a valid character, it resulted in an error. Now, the method completes without effect as documented. (#49817)

### ScaledDecimal, Fraction >> asFloat could return incorrect results

For some input values, the method `ScaledDecimal >> asFloat` and `Fraction asFloat` could return slightly incorrect values. These method have been reimplemented, and return the closest floating point number, rounding to nearest even in case of a tie. (#50071, #50170)

## Login log did not record failed logins

The login log, enabled by the configuration parameter `STN_LOGIN_LOG_ENABLED`, was intended to record failed logins as well as logins that succeeded. However, failed logins were not being recorded. (#50169)

## GsFile primitives not interrupted by SIGTERM

When a Gem is executing `GsFile` read primitives, and the Stone was shutdown, the Gem was not interrupted, which could leave the shared memory allocated. (#50186)

## Improved reporting on client disconnect errors

The errors reported when a Gem is unexpectedly disconnected has been improved to make debugging easier: now the specific code and the affected socket are included in the error messages. (#50242)

## Checking for vote state incorrect

The methods `Repository >> reclaimAllWait:` and `waitForVoteStateIdleSecs:` check if voting is occurring. This was checking the wrong voteState result. This bypassed reporting on specific sessions that were holding up voting. (#50114).

## Unicode Compares causes strings to not be returned in ExportedDirtyList

The `ExportedDirtyList` is used in GBS to correctly handle the state of objects replicated to the client, independent of having references on the client. However, the process of performing an ICU-based unicode comparison (regardless of whether the server is in Unicode Comparison Mode), caused a bit to be unset and the object was not returned in the `ExportedDirtyList`. (#50266)

## performOnServer: failed for configurations with client s-bit set

When the client executable (linked topaz or Gem) has the s-bit set, and is started by a user other than the owner of the executable, `System class >> performOnServer:` failed due to permission errors on temporary files. (#50191)

## Object >> subclassResponsibility error was not reported correctly

The messages composed by `subclassResponsibility` and `subclassResponsibility:` reported the receiver, rather than the class of the receiver; in addition `subclassResponsibility:` incorrectly parsed the selector for class methods. (#50072)

## Transaction log debug level change

The configuration parameter `STN_TRAN_LOG_DEBUG_LEVEL`, when set to values greater than 1, cause additional information to be written to the transaction logs; this should only be done as instructed by GemTalk Engineering. In v3.6.6, the output when `STN_TRAN_LOG_DEBUG_LEVEL` is set to level 1 includes page allocation information that previously required a higher level. The additional transaction log space requirement at level 1 is expected to be on the order of 1% of additional space over the requirement for level 0. (#50241)

## Private method could SEGV

The private method `Array>>_insertAt:value:value:value:value:value:numToMoveDown:` invoked the wrong primitive, and would SEGV. (#50346)