GemStone/S 64 Bit™
Release Notes

Version 3.7.5

March 2026

Doc rev 75.1

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

PATENTS

GemStone software has been 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".

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**, **Windows Server**, and **Azure** 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.
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**, m**acOS**, 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
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 3.7.5 release. Read these release notes carefully before you begin installation, upgrade, 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 3.7.5 for your platform/s.

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

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

Email: 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 3.7.5

Chapter 2. Bug Fixes

Chapter

1 Release Notes for 3.7.5

Overview

GemStone/S 64 Bit™ 3.7.5 is a significant new release of the GemStone/S 64 Bit object server, including a number of new features and feature enhancements and fixing several critical bugs. This release also adds support for Debian and for installation using APT.

These Release Notes include changes between the previous version of GemStone/S 64 Bit, v3.7.4.3, and 3.7.5. If you are upgrading from a version prior to 3.7.4.3, review the release notes for each intermediate release to see the full set of changes.

For details about installing GemStone/S 64 Bit 3.7.5 or upgrading from earlier versions of GemStone/S 64 Bit, see the *GemStone/S 64 Bit Installation Guide* for v3.7.5 for your platform.

Supported Platforms

Platforms for Version 3.7.5

GemStone/S 64 Bit version 3.7.5 is supported on the following platforms:

- Red Hat-compatible Linux 7.9, 8.10, and 9.6, Ubuntu 22.04 and 24.04, and Debian 12 and 13, on x86_64.
  GemStone is tested on a mixture of Red Hat, CentOS, and Rocky; these are all considered fully certified platforms. Any reference to Red Hat applies to any Red Hat-compatible distribution.
- Ubuntu 24.04 on ARM. Linux ARM is for development only, not for production.
- macOS 15.5 (Sequoia) and 13.7.6 (Ventura), on x86 Apple silicon (ARM) and x86. macOS distributions are for development only, not for production.

For more information and detailed requirements for each supported platforms, please refer to the *GemStone/S 64 Bit v3.7.5 Installation Guide* for that platform.

GemBuilder for Smalltalk (GBS) Versions

The following versions of GBS are supported with GemStone/S 64 Bit version 3.7.5:

GBS/VW version 8.8.1

VisualWorks 9.5 64-bit	VisualWorks 9.4.1 64-bit	VisualWorks 9.3.1 64-bit	VisualWorks 8.3.2 64-bit	VisualWorks 8.3.2 32-bit
▸ Windows 11 ▸ RedHat ES 8.10 and 9.6; Ubuntu 22.04 and 24.04	▸ Windows 11 ▸ RedHat ES 7.9, 8.10, and 9.6; Ubuntu 22.04 and 24.04	▸ Windows 11 ▸ RedHat ES 7.9, 8.10, and 9.6; Ubuntu 22.04 and 24.04	▸ Windows 11 ▸ RedHat ES 7.9, 8.10, and 9.6; Ubuntu 22.04 and 24.04	▸ Windows 11

GBS/VA version 5.4.7

VAST Platform 13.0.1	VAST Platform 11.0.1	VA Smalltalk 8.6.3
▸ Windows Server 2016 and Windows 11	▸ Windows Server 2016 and Windows 11	▸ Windows Server 2016 and Windows 11

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

VSD 5.6.5 is included with the GemStone distribution, and can also be downloaded as a separate product from https://gemtalksystems.com/vsd.

GemBuilder for Java

You must use GemBuilder for Java (GBJ) v3.2.1 with GemStone/S 64 Bit v3.7.5. GBJ v3.2.1 adds support for Windows clients, but has no other functional changes from v3.2.

There there were a number of significant infrastructure changes between GBJ v3.1.3 and v3.2; see the Release Notes for GemBuilder for Java v3.2 for more information on upgrading.

Rowan and Jadeite

Rowan is an open-source source code management system for GemStone, with underlying git repositories managing tonel-format code source files. GemStone/S 64 Bit v3.7.5 includes Rowan v2.10 for legacy users and v3.5 for use with Jadeite for Pharo.

Jadeite is an open-source GUI application for developing and debugging Smalltalk code in a GemStone repository; currently alpha-level and suitable for testing and evaluation. Jadeite supports code management operations on rowanized source code in Rowan projects and packages. Jadeite for Pharo clients are installed into Pharo 13.0 images and can be used on Linux, Windows, and Mac. Issues with the Jadeite alpha should be reported as tickets on the Jadeite project, github.com/GemTalk/JadeiteForPharo/issues

For information and help on Jadeite and getting started, see the online help: docs.gemtalksystems.com/current/JadeiteHelp.

To find out more about the current status of these open-source projects, inquire on one of GemTalk's email forums: the customer forum at gemstone-smalltalk@lists.gemtalksystems.com, or the GLASS forum at glass@lists.gemtalksystems.com.

Jadeite for Pharo with Rowan

Jadeite for Pharo clients can connect to GemStone servers running Rowan 3, in a repository based on `$GEMSTONE/bin/extent0.rowan3.dbf`. Currently, Rowan 3 cannot be installed in existing extent0.dbf-based repositories. Jadeite for Pharo provides support for Rowan code management, as well as code development and debugging support.

For more information and setup instructions, see: github.com/GemTalk/JadeiteForPharo/blob/main375/SetupForWithRowan.md

Jadeite for Pharo without Rowan

GemStone/S 64 Bit v3.7.5 includes a Rowan stub, which allows Jadeite to interact with an `extent0.dbf`-based repository, without a full Rowan installation. This allows Jadeite for Pharo to be used with an existing GemStone repository that has been upgraded to v3.7.5.

As in base GemStone, there is no code management support when running with the Rowan stub. Source code can be managed using filein/fileout to disk files. Monticello packages will be supported; but this support is not yet available.

The 3.7.5 distribution includes the RowanStubForJadeite installation under `$GEMSTONE/examples/jadeite`.

For more information and setup instructions, see: github.com/GemTalk/JadeiteForPharo/blob/main375/SetupForWithoutRowan.md

1. Installation and Distribution Changes

Updated shared library versions

The version of zlib has been updated from 1.3 to 1.3.1

The version of openssl has been updated from 3.0.16 to openssl 3.0.19

Installation using APT packages

GemStone is now supported on Debian 12 and 13.

Added .deb distributions

GemStone distributions now include a .deb package format, which can be installed using APT (apt or dpkg). This is supported on Debian-compatible Linux distributions, including Ubuntu. The installation process and restrictions are similar to that for RPM for Red Hat.

An additional Installation Guide is provided, *GS64-InstallGuide-LinuxAPT-3.7.5.pdf*.

GemStone PPA

GemTalk now provides a PPA (Personal Package Archive), that allows you to install using apt directly, rather than using the downloaded .deb file. Instructions are at ppa.gemtalksystems.com

2. Support for log rotation for application log files

In v3.7.4, support was added for system log file rotation, including gem and server process logs, using the SIGHUP signal. The general rotation process is: to first move (or delete) the existing disk log file, then send SIGHUP to the process, which will start writing to a new log file with the previous name and mode.

This process can now be applied to files opened by the application using GsFile or GsLog (for FileSystem files).

Image Changes

Added method GsFile >> reopen

The following method has been added:

```
GsFile >> reopen
```

If the underlying disk file no longer present, GsFile creates a new disk file with the same name and path and returns true. Reopen is done using freopen (see man 3 fopen), using the existing path and mode. If the disk file still exists, nothing is done and it returns false. If an error occurs, this method returns nil. Receiver must be a server file, with mode containing 'a' or 'w', and cannot be stdin, stdout, or stderr.

Added class GsLog

GsLog is a new class in v3.7.5, which provides a mechanism to do SIGHUP-based file rotation with a FileSystem-based log file. GsLog implements #reopen, used as with GsFile.

An instance of GsLog encapsulates a FileReference and a ZnCharacterWriteStream, allowing the underlying file to be reopened by a LogRotateNotification handler. GsLog is thread-safe. Writes to the log file are protected by a mutex. Reopen tries to lock the mutex and signals a Warning it cannot lock. If a file exists during initial open or during a reopen, the existing file is appended to. If the specified file does not exist, a new file is created.

Note that there is a risk of deadlock when file writes can occur during Ephemeron mourning. See the class comments for GsLog for more information.

Added class LogRotateNotification

The class LogRotateNotification has been added. An instance of this class is signalled asynchronously when the Gem process or topaz -l receives a SIGHUP. By default, it is not enabled. If enabled, it does not need to be reenabled after a signal is received. When enabled, handler for this notification should be installed that calls #reopen on each GsFile or GsLog that is configured for log rotation.

Log rotation process

The application sets up a handler for the LogRotateNotification signal (using `addDefaultHandler:`). This handler sends `reopen` to the file being rotated. If more than one file is rotated, even if the rotation is on different schedules, the handler should reopen all of these files. A reopen is a no-op if the file was not moved or deleted.

At the OS level, log rotation of application GsFiles should move the disk file to an archive location, and send SIGHUP to the Gem.

Note that using **mv** within the local file system ensures no writes can be lost. If a file write occurs within the small gap between log archiving and sending SIGHUP, there is a risk a write may be lost if the archive process does a log copy and deletes the old log, including **mv** to a non-local file system destination.

Example handler

In this example, `myGsFile` is a GsFile and `myGsLog` is an GsLog, both of which are open for write or append. The following handler reopens both.

```
LogRotateNotification enableSignalling.
LogRotateNotification addDefaultHandler: [:ex |
   myGsFile reopen ifNil: [
      self error: 'reopen failed ', myGsFile pathName].
   myGsLog reopen.
   ex resume.
   ].
```

3. Changes in Tranlog Restore

GcReclaim threads for tranlog restore

The method `Repository >> restoreFromBackup:*` shuts down the reclaimGem, and when it completes, restarts the reclaimGem with an increased number of threads, to ensure reclaim does not hold up tranlog restore. Previously, the system computed the number of threads, which could be considerably larger and did not respect STN_MAX_GC_RECLAIM_SESSIONS. Now, the maximum number of threads is the configured STN_MAX_GC_RECLAIM_SESSIONS or 4, whichever is larger.

You should review your configuration of STN_MAX_GC_RECLAIM_SESSIONS, to ensure the setting is appropriate for your application. This is most important for repositories performing a large volume of tranlog restore.

Tranlog restore does not continue if session terminated

In previous releases, when the session that initiated restore from tranlogs was terminated, the restore continued until all specified tranlogs were restored. Now, terminating the session will stop tranlog restore at that point. Restore can be continued by logging in and re-executing the tranlog restore operation.

One-step restore and commitRestore

After a programmatic `restoreFromBackup*` (of a repository in full tranlogging mode), you must login again to execute `commitRestore`. In cases where you do not need to restore tranlogs, the restore process can be simplified with the added method `disableRestoreFromLogs`. After executing this method (in both encrypted as well as non encrypted extent systems), the `restoreFromBackup` skips restoring tranlogs and automatically performs a `commitRestore`.

Note that any commits that are not in the programmatic backup are lost, since transaction logs are not replayed, and transaction logs cannot be replayed after `commitRestore`.

This method should be executed after login and before the restore operation. The status is set in SessionTemps, and only applies for that restore operation (since restore terminates the session).

`Repository >> disableRestoreFromLogs`
: Causes the next restoreFromBackup operation (normal or secure) by the session to immediately commit the restore and return the repository to normal mode. Executing the method #commitRestore is unnecessary.

Bug Fixes related to Restore

Excess ReclaimGem threads incorrectly left running after restore

After the number of reclaim threads was increased following a programmatic `restoreFromBackup*`, the extra threads were left running until the next Stone restart. (#51269)

Computation of STN_MAX_GC_RECLAIM_SESSIONS could overwrite STN_NUM_GC_RECLAIM_SESSIONS

When STN_MAX_GC_RECLAIM_SESSIONS is set to the default, a value is computed. Previously, this computer value could overwrite a specific setting for STN_NUM_GC_RECLAIM_SESSIONS. (#51259)

4. ParallelDo: distributed asynchronous parallel object processing

Note:
ParallelDo is a preview feature in v3.7.5. Additional large-scale testing is needed, and the API is subject to change.

The ParallelDo framework allows you to concurrently process a large number of objects across multiple Gems running on multiple hosts. This is designed for operations such as migrate, which may require modifying many objects. The objects are provided in a collection, to a manager gem; the objects are divided into chunks and distributed over the worker gems to do the actual processing.

The worker gems run asynchronously, and automatically commit to avoid out of memory issues, automatically catch and retry failed commits, and capture other errors for later examination.

ParallelDo classes include:

- **ParallelDoManager** - the main interface, which controls the ParallelDoWorkers. It is configured with the collection, the processing code, and the set of one or more hosts on which the workers will run. It coordinates between the workers and tallies and reports the results. The manager ensures that:
  - Each object is processed no more than once.
  - All objects are processed, if all worker gems run to normal completion.
  - At the end of the run, no new committed references to successfully processed objects exist. Successful processing means no run-time error was raised while processing that object.
- **ParallelDoWorker** - performs a subset of the work on a particular host. Workers login as the same UserProfile as the manager, using a one-time password supplied by the manager.

  The workers:
  - May be on the Stone's host, distributed across multiple remote hosts, or both.
  - Run asynchronously (non-blocking) from each other and the manager.
  - Automatically commit their transaction when temporary object memory starts to become full.
  - Catch and automatically retry transaction failures (failed commits)
  - Catch and automatically store Smalltalk run-time errors. including the call stack.
  - Are restarted by the manager if interrupted before completion.
  - Always run to completion, unless a fatal error occurs.
- **ParallelDoHost** - represents the host on which a ParallelDoWorker is doing work. The Stone's node can be used as a host, as well as remote nodes. Each host can be configured to run one or more worker gems, and you may specify some host-specific configuration values.
- **ParallelDoExternalSession** - a subclass of GsTsExternalSession with extensions to support ParallelDo.
- **ParallelDoReportStream** - supports reports with columns for ParallelDo reports.

ParallelDo Requirements

- The ParallelDoManager must run on the Stone's host.
- ParallelDoManager must be run as a UserProfile that is not SystemUser. This UserProfile must have #CreateOnetimePassword and #SessionControl privileges, as well as privileges such as #CodeCompile that may be required to perform the processing block. The workers will run with this same UserProfile.
- GEMSTONE_NRS_ALL needs to be set to specify the Netldi name and port number (unless using default-named gs64ldi). This must be defined in the environment for the Stone and Netldi, and in the Netldi's environment on remote hosts.
- The Netldi conforming to GEMSTONE_NRS_ALL must be running the Stone's host and all remote hosts.

Using ParallelDo to process objects

1. Create the ParallelDoManager

To start with, create an instance of ParallelDoManager, and provide it with the collection of objects to be processed and the processing code in a block.

All objects to be processed must be provided in a single collection, typically an Array or IdentitySet. The manager divides this collection into subcollections (called chunks). The maximum size of each chunk is specified by the user and may be no larger than 2034 elements, to ensure all chunks remain small objects.

The processing block must be a one-argument block. For example:

```
[:obj | obj migrate ]
```

Use `makeCurrent` so this ParallelDoManager instance is saved in this session's UserGlobals, and can be accessed using `ParallelDoManager current`.

For example,

```
(ParallelDoManager
   newForCollection: myColl
   workerBlock: [:obj | obj migrate ]) makeCurrent.
```

2. Configure and add hosts

Each host must have the host name and the number of sessions specified. There are methods to automatically set the host name for workers running on the Stone's host.

To use host with a default configuration, use methods such as

```
ParallelDoManager >> addLocalHostWithNumSessions: numSessions
```

```
ParallelDoManager >> addHostWithName: hostname withNumSessions:
   numSessions
```

For example:

```
parallelDoMgr
   addLocalHostWithNumSessions: 2;
   addHostWithName: 'lark' withNumSessions: 4.
```

Hosts (instances of ParallelDoHost) can be configured, with both Gem configuration options and remote cache options. To do this, create an instance of ParallelDoHost and configure it before adding it to the ParallelDoManager.

You may specify configuration for the gem

```
printCommitConflictDetails:
gemStatmonitorArgs:
tempObjCacheSizeMb:
tempObjCacheSizeGb:
```

and for the remote cache:

```
sharedCacheSizeGb:
sharedCacheSizeMb:
sharedCacheLargeMemoryPagePolicy:
cacheWarmerArgs:
sharedCacheLargeMemoryPageSizeMb:
```

The following example configures a remote host with gem and cache configuration options:

```
aHost := ParallelDoHost
   newForHostname: 'lark' withNumberOfSessions: 4.
aHost
   sharedCacheSizeGb: 4 ;
   tempObjCacheSizeMb: 200 ;
   sharedCacheLargeMemoryPagePolicy: 2 ;
   sharedCacheLargeMemoryPageSizeMb: 1024 ;
   cacheWarmerArgs: '-d -n 2 -L /tmp' ;
   printCommitConflictDetails: 2;
   gemStatmonitorArgs: '-i1 -u 3 -z -f
      /tmp/statmon1s_%%S_%%P_%d-%m-%y-%H:%M:%S'.
parallelDoMgr addHost: aHost.
```

3. Prepare and run

Before performing the processing, the manager must create chunks from the collection, create the workers and log them in.

- To create chunks with the default size (2034), use `createChunks`. You may configure the chunk size using `createChunksWithSize:`.; smaller chunks may be useful if there is more risk of conflict.
- Commonly, you would use `prepareToRun` to create and login the sessions and wait for cache warming to complete. Using `prepareToRunWithDebug` enables debugging on the workers. These methods execute the methods `createWorkers`, `createSessions`, and `waitForCacheWarming`.
- Perform the operation using `run`, and wait for processing to complete using `pollUntilCompleteAtIntervalMs:`, which checks for completion at the given interval.

Each worker searches for a chunk to process which is not locked and has not been processed by another worker. The worker does this by requesting a write lock on a candidate chunk. If the lock is granted and the last element in the chunk is not nil, then the worker processes the chunk while holding the write lock on it. The write lock guarantees a given chunk will be processed by no more than one worker.

As the worker enumerates a locked chunk, objects which have been processed are removed from it by storing nil. When a chunk is completely processed, all elements will be nil. The write lock on the chunk will be released by the next successful commit.

The next chunk available for processing is stored in global cache statistic. This serves as a hint to workers to indicate where in the list of chunks to start searching for the next chunk to process.

Workers are automatically logged out when they have completed their assigned elements, or on fatal error.

Gem logs for all workers are automatically retained and are not subject to deletion upon clean logout.

4. Check results and errors

Collect errors and the worker report, using `collectErrors` and `createWorkerReport`.

createWorkerReport will provide the results including the number of errors that occurred. If there are errors, examine the objects in `ParallelDoManager current allErrors`., which include the call stacks for the error as well as the objects on which the error occurred.

Example

The following example uses two workers on localhost and two workers on a remote host, including configuration of the remote cache, to migrate objects in the collection *theColl*.

```
| parallelDoMgr aHost |
parallelDoMgr := ParallelDoManager
   newForCollection: theColl
   workerBlock: [:obj | obj migrate].
parallelDoMgr makeCurrent.
parallelDoMgr addLocalHostWithNumSessions: 2.
aHost := ParallelDoHost
   newForHostname: 'lark' withNumberOfSessions: 2.
aHost
   sharedCacheSizeGb: 4;
   tempObjCacheSizeMb: 200.
parallelDoMgr addHost: aHost.
ParallelDoManager current
   createChunks;
   prepareToRun;
   run;
   pollUntilCompleteAtInterval: 500;
   collectErrors;
   createWorkerReport.
```

The final call to `createWorkerReport` prints a report formatted in columns:

Worker	Host	SessId	Pid	Time(s)	TotalObjs	Passed	Errors	Commits	FailedCommits
1	santiam	8	2198628	1	360676	360676	0	2	0
2	santiam	9	2198635	1	380358	380358	0	2	0
3	lark	10	331346	1	40680	40680	0	2	0
4	lark	11	331353	1	48816	48816	0	2	0

5. External Session Changes

GsTsExternalSession improved handling of String/Unicode string results

When code executed by a GsTsExternalSession returns a kind of string, the external session automatically converts the result into a string in the calling session.

Previously, only remote objects of type String or Unicode7 were correctly converted, since these hold Characters requiring no more than 8 bits. The behavior for these objects is unchanged; the local object created for the remote object is of the same class as the remote object.

Remote objects of DoubleByteString, QuadByteString, Unicode16 and Unicode32 were previously returned as ByteArrays. Now, these values are automatically encoded into instance of Utf8 prior to returning. This is the same process as the recommended workaround in previous releases.

Instance of Utf8 were, and are, automatically converted to a local object of the appropriate class:

- When Repository is not in Unicode Comparison Mode, a traditional string: String, DoubleByteString, or QuadByteString.
- When Repository is in Unicode Comparison Mode, a Unicode string class: Unicode7, Unicode16, and Unicode32).

Note that when the remote execution's return values are traditional or unicode strings that do not match the default classes for the local repository's Unicode Comparison Mode (most commonly if the remote repository has a different Unicode Comparison Mode than the local repository), the class of the result in the local repository may vary if some require encoding and others do not.

If you previously were not using the workaround, but doing the multi-step process required to decode the ByteArray into a kind of string, this will no longer work; you will need to remove the manual decode.

Note that legacy GsExternalSession has no change in behavior.

GsTsExternalSession bugs fixed

Compile problems when GsTsExternalSession executed source in Utf8

When the input to a GsTsExternalSession executed is a Utf8, the compiler may have attempted to parse beyond the end of the decoded input. This resulted in a compiler error (#51543)

Error during GciError >> signalCompileError when compile error occurred in GsTsExternalSession

When a compile error occurred in the external session code, during `resolveResult:` it could have encountered an error while processing the error. (#51563)

Force flushing of OOPs in the remote session's export set

Objects returned from the remote session have their OOPs added to the remote session's export set. Non-string type objects must be manually removed from the remote export set. String type objects are removed periodically. To force removal, the following method has been added.

```
GsTsExternalSession >> flushReleaseOops
```

Added setGemTrace:

The method `setGemTrace:` has been added to GsTsExternalSession and GsExternalSession, to control tracing of GCI commands. See the image method comments for details.

Other added GsTsExternalSession method

`GsTsExternalSession >> newUtf8Object:` *aUtf8*
Returns an oop of a Utf8 created in the external session containing the UTF-8 encoded contents of the argument, which must either already be a Utf8, or a string containing legal UTF-8 data.

Detached Execution changes and bug fixes

Detached execution refers to RPC Gems that are started using `GsTsExternalSession forkAndDetach*` methods, or the newly added `GsHostProcess >> forkAndDetach` (see page 25).

Detached execution Gem now logs out when execution string completes

Previously, when the code provided to the `forkAndDetach*` method completed, control returned to the instance of GsTsExternalSession and further executions was possible in that session. This was not intended behavior, and had the risk of the session holding the commit record. Now, the remote detached execution Gem will logout when it completes execution of the argument code.

Detach execution Gem did not handle DEBUGGEM

When a Gem was running with detached execution, and topaz DEBUGGEM was used to attach to that Gem, the resume command could not be used to resume; the detached execution Gem had to be terminated. (#51457)

Insufficient information for errors in remote execution

In normal (non-detached) external session code execution, errors in the remote code are reported in the calling session, but this is not possible in detached execution. In detached execution, the stacks are printed in the Gem log for the detached execution Gem. Previously, these stacks did not include enough information for debugging; now, stacks with arguments are printed. (#51541)

Detached execution handled compiler errors still printed to log

When a Gem runs with detached execution, unhandled errors cause stacks to be written to the log of the detached execution Gem. Previously, CompileErrors with handlers were excepted, and were logged in spite of the handler. Now, a CompileError that is caught and handled will not have a stack printed in the Gem log. (#51512)

6. DateAndTime and DateTime changes and bug fixes

The relationship between, and the organization of behavior, between DateAndTime and its superclass, DateAndTimeANSI, have been modified and cleaned up in this release. Behavior has been moved from one class to the other. See also "DateTime removed/deprecated methods" on page 26.

DateAndTimeANSI now abstract

DateAndTimeANSI was conceptually intended to be an dialect-neutral superclass containing the ANSI-specified DateAndTime behavior, while the DateAndTime implementation itself was GemStone-specific extensions and implementation details. However, both were previously concrete, and instances of DateAndTimeANSI could be created. With the introduction of SmallDateAndTime, issues were introduced on some code paths. DateAndTimeANSI is now abstract.

Creating a DateAndTime

The following instance creation method has been added:

`DateAndTime class >> posixSeconds:` *secondsSince1970* `offsetSeconds:` *offsetSeconds*
Return a SmallDateAndTime if arguments are in range. *aNumber* is UTC time in seconds since the beginning of 2001, *anOffset* is offset from UTC.

Converting between DateAndTime and DateTime

The following methods have been added, allowing conversion between the ANSI DateAndTime class and the legacy DateTime class.

DateTime instances know their specific TimeZone, but DateAndTime instances only know their offset, which could correspond to multiple different TimeZones. Two conversion methods are provided so you can either use the current TimeZone or provide a specific TimeZone as an argument for the converted DateTime instance.

`DateAndTime >> asDateTime`
Answer a DateTime representing the same moment in time as the receiver, in the current TimeZone. The offset of the receiver is ignored. DateTimes know their TimeZone, and the TimeZone knows about any daylight saving time transitions, but DateAndTimes only know their offset. Most offsets can occur in several different TimeZones, so we cannot infer the TimeZone from the offset.

`DateAndTime >> asDateTimeInTimeZone:` *aTimeZone*
Answer a DateTime representing the same moment in time as the receiver, in the given TimeZone. The offset of the receiver is ignored.

`DateTime >> asDateAndTime`
Answer a DateAndTime representing the same moment in time as the receiver, and with the offset appropriate to the receiver's TimeZone as of that moment.

Extracting Date from DateAndTime

The method `DateAndTime >> asDate` has been added, to extract the Date from a DateAndTime without needing the expensive conversion to a DateTime.

Change in handling of DateAndTime creation on the cusp of daylight saving time

When daylight saving time causes the clock to be set forward or back. there is a period of time in which there is either no local time, or two ambiguous local times. This was previously handled similarly to the legacy non-ANSI DateTime; this has been changed so DateAndTime conforms to the ANSI spec.

When the time is set back at the end of daylight saving time, there two local times corresponding to a single UTC time. ANSI specifies that an ambiguous local time should resolve to the earlier of the two local times. Previously, ambiguous DateAndTimes were resolved for the later of the two local times; now, the former one is created, to conform to ANSI.

When creating a DateAndTime from a local time that does not exist, previously this was allowed and created DateAndTime one hour later; for example, creating a DateAndTime for the non-existent time of 2:30 while daylight saving time is starting, created the actual local time of 3:30. The ANSI standard specifies that this should error, and attempting to create this nonexistent local DateAndTime now signals an error.

Note that the older, non-ANSI DateTime class resolves ambiguous times to the earlier of the two, and handles nonexistant times by creating the equivalant later time; DateTime is unchanged, for historic reasons. If your application requires rigorous handling of date-time values during DST transitions, it is recommended that you use UTC times, which are never ambiguous.

Incorrect values in DateTime instance creation out-of-range error

When the second or hour argument to `DateAndTimeANSI class>>year:day:hour:minute:second:offset:` was out of range, the error message provided an incorrect upper limit (off by one) (#51523)

DateAndTime creation with negative partial offsets handled incorrectly

DateTimes in TimeZones with a partial negative offset, that is, in a TimeZone east of UTC and west of the International Date Line, and with a offset that is not an increment of a whole hour, produced DateTimes with an offset that was incorrect by one hour. (#51657)

Unreliable methods disabled

The following methods could produce incorrect results, and the functionality has been removed; they will now return an error. The methods remain to provide the replacement method, and send deprecated: for deprecation tracking although they are not technically deprecated.

`DateAndTimeANSI class >> secondsLocal:offset:`
Use `secondsUTC:offset:` with the local time converted to UTC.

`TimeZoneInfo >> offsetAtLocal:`
Use `offsetAtUTC:` with the local time converted to UTC.

`TimeZoneInfo >> transitionAtLocal:`
Use `transitionAtUTC:` with the local time converted to UTC.

`TimeZoneTransition >> transitionTimeLocal`
Convert the result of transtionTimeUTC to local time.

Added methods related to Time Zone transitions

`TimeZoneTransition >> asDateAndTime`
Answer the transition moment expressed as a DateAndTime with the offset being transitioned to.

`TimeZoneInfo >> transitionsAtOffsetPosixSeconds:` *seconds*
Given a local 'wall clock' time encoded as the seconds relative to Jan 1, 1970 in the time zone defined by the receiver, answer a collection of transitions which would result in that time. Normally the result is size 1, but may 0 or 2 as a result of DST.

7. Other image changes

Finding reference paths to classes

Scans to find a reference path to an object, using GsSingleRefPathFinder or the recently added methods `findReferencePath` and `findReferencePathString`, now can be used to find paths to instances of Class or Metaclass3.

The GsSingleRefPathFinder printToLog instance variable is now a SmallInteger rather than a Boolean, with the following values:

0 – no printing
1 – print start and end times
2 – also print start and end times of each scan
3 – also print limit set details for each scan

findReferencePath* added to PrivateObject

The following methods have been added:

```
PrivateObject >> findReferencePath
PrivateObject >> findReferencePathString
```

Added method to find all instances with a single reference

A new method has been added to find all objects in the repository that have only a single unique reference.

`Repository >> listObjectsWithOneReferenceWithMaxThreads:` *maxThreads* `waitForLock:` *lockWaitTime* `percentCpuActiveLimit:` *percentCpu*
Scans the entire repository for objects referenced once and only once by any other object, and returns these as a GsBitmap.

Multiple references from a parent object to a child object are considered a single references and the child object will be included in the result. Objects which reference themselves will be included in the result if that reference is the only one.

This method begins a transaction and runs in this transaction for its duration. It should therefore not be used in production systems due to commit record backlogs which may cause excessive repository growth.

Uncommitted objects and dead not reclaimed objects are excluded from the result. Note that objects in the result set may be disconnected from the repository (unreachable) and therefore could disappear after the next garbage collection cycle.

Raises an error if the session has modified persistent objects. Raises an error if a garbage collection operation is in progress with the repository vote state is not zero (see `System class >> voteState`). The *lockWaitTime* argument is used to specify how many seconds method should wait while attempting to acquire the gcLock. No other garbage collection operations may be started or in progress while this method is running. There also must be no outstanding possible dead objects in the system for the GC lock to be granted.

Starts *maxThreads* on the host system and allows the host to run up to *percentCpu* percent CPU usage. A page buffer of 16 pages (256 KB) is allocated per thread.

Added method to search GsBitmap by class

The following method has been added:

`GsBitmap >> allCommittedInstancesOf:` *aClass*
Searches the receiver for all committed objects which are instances of *aClass*, and returns a new GsBitmap containing those instances. Uncommitted objects present in the receiver are ignored, and not included in the result.

Configuring the AdminGem

Starting in v3.7.2, the AdminGem only runs when needed: after markForCollection to handle voting, and if epoch GC is enabled.

The configuration method `System class >> setAdminConfig:toValue:`, which transiently sets values in the currently running AdminGem, usually had no effect if Epoch was not enabled. Now, this method returns an error if invoked when the AdminGem is not running.

Use the method `System class >> setPersitentAdminConfig:toValue:`, to define the configuration for the current AdminGem and which will be used the next time it starts up.

Method comments relating to the AdminGem have been improved; and some minor issues in GcGem session management have been fixed.

The following methods have been added:

`System class >> getPersistentAdminConfig:` *configSymbol*
Return the AdminGem persistent configuration settings, which are values in GcUser's UserGlobals. May differ from current runtime settings. See `getAdminConfig:` for relevant symbols.

`System class >> getPersistentReclaimConfig:` *configSymbol*
Return the ReclaimGem's persistent configuration settings, which are values in GcUser's UserGlobals. May differ from current runtime settings. See `getReclaimConfig:` for relevant symbols.

Added System class informational methods

Added methods to get information on session locks

The following methods have been added:

`System class >> sessionLocks:` *aSessionId*
Returns an Array describing locks held by the specified session, including
1. an Array of read-locked objects
2. an Array of write-locked objects
3. an Array of objects with deferred unlocks

Deferred unlocks are objects for which the unlock request was received by stone while another session was holding the commit token; they will be unlocked as soon as the commit token is released.

`System class >> sessionLocksReport:` *aSessionId*
Returns a String describing all of the objects locked by the specified session.

More information in System currentSessionNames

The report generated by `System class >> currentSessionNames` now includes the transaction level and the voting status. For example:

```
2 GcUser reclaimgcgem 1642585 on localhost
3 SymbolUser symbolgem 1642586 on localhost
4 DataCurator topaz -l 1705674 on localhost transactionLevel 1,
   this session
5 SystemUser gem 1693847 on localhost transactionLevel 1 not
   voted
6 GcUser admingcgem 1705766 on localhost
```

Added methods for cache information

The following methods have been added:

`System class >> cachesList`
Return an Array describing all shared caches that the stone process is managing, including the cache on the stone machine. Result contains one element per cache. Each element of result is a 8 element Array containing:
- hostName, a String
- a Boolean - true if cache was created as a mid-level cache
- a SmallInteger - total number of sessions connected to the cache

- a SmallInteger - max number of sessions for the cache
- a SmallInteger - size of cache in KB
- a SmallInteger - number of sessions using cache as a mid-level cache
- a SmallInteger - zero or sessionId of hostagent on stone host servicing the cache
- a String - ipAddress of the host of the cache
- a SmallInteger - zero or sessionId hostagent running on a mid-level cache host.

`System class >> cachesReport`
Return the result of `cachesList` formatted as a string, one line per cache.

`System class >> reportForSession:` *sessionId*
Returns a report containing internal information about the specified session. For example:

```
userId DataCurator, transactionLevel 1, viewTime 28 seconds
   ago, onOldestCr true, commitsSinceView 0,
gemProcessId 3814914, gemHost localhost, gciPeer 127.0.0.1,
   login 28 seconds ago, tempOops 1999
```

ClassOrganizer support for class categories

Methods have been added to provide information about class categories. These methods accept and return kinds of String, since they are designed to support GUI tools. These category methods treat hyphen-delimited class category strings as representing a hierarchal structure and the added methods are provided to query both for specific full category strings and association with logically-inherited supercategories.

`ClassOrganizer >> classCategoryNames`

`ClassOrganizer >> allClassCategoryNames`

`ClassOrganizer >> classCategoryNamesInDictionaryName:` *dictionaryName*

`ClassOrganizer >> allClassCategoryNamesInDictionaryName:` *dictionaryName*

`ClassOrganizer >> classNamesInClassCategoryNamed:` *categoryString* `dictionaryName:` *dictionaryName*

`ClassOrganizer >> classNamesUnderClassCategoryNamed:` *categoryString* `dictionaryName:` *dictionaryName*

See the image method comments for more details.

Other added image methods

Class >> subclass:instVarNames:classVars:inDictionary:

Similar to other class creation methods, but allows omission of the `classInstVarNames:` and `poolDictionary:` keywords.

GsFile class >> contentsOfServerDirectory:

The method `GsFile class >> contentsOfServerDirectory:` has been added. This is a convenience methods invoking `contentsOfDirectory:onClient:`.

GsHostProcess >> forkAndDetach

This method forks a child process, similar to the existing `fork` method. The child process will not be killed when this session's gem or topaz -l process exits, nor will it be killed when the receiver is finalized by in-memory GC of temporary objects.

Locale >> beCurrent

Sets the receiver instance of Locale to be the instance returned by Locale current.

Random >> boolean

Return a random Boolean value.

Add fullBackup method variants

The following methods have been added:

`Repository >> fullBackupGzCompressedTo:` *fileNames* `MBytes:` *mByteLimit* `threads:` *numThreads*

`Repository >> fullBackupLz4CompressedTo:` *fileNames* `MBytes:` *mByteLimit* `threads:` *numThreads*

These are variants of the existing `fullbackup` methods that provide specification of both extent sizes and the number of threads.

System class >> stoneCacheStatisticWithOffset:

Return the value of the cache stat with the given offset, which must be >=1 and <= size of the Array returned by the `cacheStatisticsDescription*` method applicable to the stone process. Returns nil if *anInt* is out of range. This method may be used on hosts remote from the stone process.

Rowan-related image changes and bug fix

Changes in STON support

STON is an object interchange format used by Rowan. Support has been added for FileReference, Path, Association, Class, Metaclass3, Fraction, and ScaledDecimal; instance of these classes previously were encoded incorrectly.

The STONFileReference class as been added to the image.

Behavior>>removeSelector:ifAbsent: did not complete removal in a Rowan repository

Executing `Behavior >> removeSelector:ifAbsent:` removed the method, but did not remove the loaded method in a Rowan repository. (#49861)

8. Deprecated and removed methods

Un-deprecated method

The method `System class >> myUserGlobals` has been un-deprecated. UserGlobals has been a required SymbolDictionary for several major releases. It is used by the new ParallelDo feature.

DateTime removed/deprecated methods

Methods that are newly deprecated:

```
DateAndTime >> currentTimeZone
DateAndTime class >> secondsSince2001
DateAndTimeANSI >> initializeAsNow
```

Disabled methods that send deprecated:

These methods were unreliable and the functionality has been removed; they remain in the image, and send deprecated:, for ease in tracking and additional information.

```
DateAndTimeANSI class >> secondsLocal:offset:
TimeZoneInfo >> offsetAtLocal:
TimeZoneInfo >> transitionAtLocal:
TimeZoneTransition >> transitionTimeLocal
```

Removed DateAndTime methods

The following private or implementation methods have been removed as part of the cleanup/reorganization of DateAndTime and DateAndTimeANSI.

```
DateAndTime class >> _scaledDecimal6mantissa:
DateAndTime class >> _smallFromMicrosecs:offset:
DateAndTimeANSI >> asFloatParts
DateAndTimeANSI >> partsFrom:
DateAndTimeANSI >> printString (now inherited)
DateAndTimeANSI >> _posixSeconds:offset:
DateAndTimeANSI >> _seconds:
DateAndTimeANSI >> _secondsLocal:offset:
DateAndTimeANSI class >> _zoneOffsetAtLocal:
DateAndTimeANSI class >> _zoneOffsetAtUTC:
```

Moved DateAndTime methods

The following list includes private methods that have been removed from DateAndTimeANSI, and added to DateAndTime or were already present.

```
DateAndTimeANSI >> asDays
DateAndTimeANSI >> asPosixSeconds
DateAndTimeANSI >> asString
DateAndTimeANSI >> beRounded  (now shouldNotImplement)
DateAndTimeANSI >> currentTimeZone
DateAndTimeANSI >> posixSeconds:
DateAndTimeANSI >> printJsonOn:
```

```
DateAndTimeANSI >> printRoundedOn:
DateAndTimeANSI >> printStringWithRoundedSeconds
DateAndTimeANSI >> rounded
DateAndTimeANSI >> _secondsUTC:offsetSeconds:
DateAndTimeANSI class >> fromString:
DateAndTimeANSI class >> migrateNew
DateAndTimeANSI class >> posixSeconds:offset:XX
DateAndTimeANSI class >> secondsUTC:offset:
DateAndTimeANSI class >> _zoneOffsetAtPosix:
```

Other removed methods

The following methods have been removed:

```
GsSingleRefPathFinder >> printTimestampToLog

ProcessorScheduler >> dbgfatallog:

Repository >> _getShrinkRepository

Repository >> _primRestoreSecureBackups:scavPercentFree:bufSize
   :privateDecryptionKey:passphrase:numThreads:shrinkRepos:newS
   ystemUserPassword:

Repository >> _restoreBackups:scavPercentFree:bufSize:numThread
   s:shrinkRepos:newSystemUserPassword:

System class >> _cacheStatWithName:opCode:
```

9. Changes in utilities

Split tranlogs using startlogsender have changes in file permissions

When **startlogsender** is invoked to write split tranlogs (using the **-F** and **-W** flags), the split tranlogs are created using the default umask & 666, and each one is made read-only when closed. Previously, the mask was 400, removing group and world read; now, the previous file mode & 0444 is applied, allowing group (or world) read depending on the default umask.

New utility script to validate configuration files

Previously, the only way to detect errors in configuration files was to attempt to start the Stone, or login, using that configuration file.

Now, you can validate a configuration file for use for the Stone, Gem, or X-509 secured Netldi. Note that this checks for syntax errors and required settings, but is not designed to determine that a particular set of extents can start up using that configuration file, e.g. for missing extents or available RAM.

Added validate_config script

The utility script **validate_config** has been added. This script takes two arguments: the path to the script, and the type; an integer indicating Stone (1), Gem (2), or X509-secured Netldi (3).

For example,

```
os$ validate_config -t 1 -p $GEMSTONE/data/system.conf
```

validate_config accepts posix "long" form argument syntax:

```
os$ validate_config --type=1 --path=$GEMSTONE/data/system.conf
```

The script logs in a superDoit solo session to execute the validation. If errors are found, the details are printed, and a message that validation failed; the script returns 1. If validation succeeds, it prints 'Validation succeeded' and returns 0.

Added System method

A method has been added that supports the config file validation. This method can only be run in a solo session. This method is intended to be used by OS level utility scripts, but can be invoked directly.

`System class >> parseConfigFile:` *fileNameString* `processType:` *typeInt*

fileNameString must be a kind of String containing the path to a configuration file, which may be absolute or relative.. *typeInt* must be the SmallInteger 1, 2, or 3, specifying how file should be parsed.

1 - parse as a Stone config file

2 - parse as a Gem config file

3 - parse as an X509-secured Netldi config file

If the file fails to parse, it prints an error with the parse details and signals an Error with number 2710. If the file parses correctly, this method returns self. This method can only be run in a solo session. Note that settings in the configuration file *fileNameString* for a type 2, Gem configuration file are applied to the solo session executing this method.

largememorypages and SHR_PAGE_CACHE_NUM_PROCS

The **largememorypages** calculation relies on an accurate value for SHR_PAGE_CACHE_NUM_PROCS for the machine that will be configured with Linux Huge Pages.

SHR_PAGE_CACHE_NUM_PROCS is normally left at a default value in configuration files. When this is at the default, GemStone calculates a value based on a number of individual configuration parameters, including STN_MAX_SESSIONS, but also including, for example, the number of reclaim threads and the number of cachewarmer threads. These parameters also support defaults that allow GemStone to calculate a value based on the environment, including the number of cores.

If SHR_PAGE_CACHE_NUM_PROCS is not explicitly set, and the **-P** argument is not used, the results are particularly inaccurate for remote cache calculation using largememorypages **-r**.

For calculations for the Stone's host:

- For the most accurate results for a specific Stone and host, run **largememorypages** with the specific configuration file to be used and on the actual host that the Stone will be running on, setup exactly as it will be used.
- For reproducible results, such as while modifying the configuration or running on a different host, run **largememorypages** with the **-P** argument or an explicit setting for SHR_PAGE_CACHE_NUM_PROCS

SHR_PAGE_CACHE_NUM_PROCS now required for remote cache calculations

Executing **largememorypages** with the **-r** argument now requires either **-P** or an explicit setting for SHR_PAGE_CACHE_NUM_PROCS.

New shared library to support statprom utility

The **statprom** utility provides an interface to use Prometheus to monitor GemStone by recording statistics from the GemStone shared page cache. In this version, **statprom** code is packaged in a new shared library, `libstatprom-3.7.5-64.so`. This simplifies installation in some configurations.

There are no changes in behavior or configuration.

Topaz Changes

Added topaz -P command line argument

Topaz now supports the **-P** *script* argument. This is similar to **-S**, but while **-S** inputs the script file argument and exits if no error occurs, **-P** inputs the script file argument but does not exit, unless there is an explicit exit in the argument script

When using **-P**, **quit** and **exit** are considered errors if they occur in a nested input file, other than the **-P** argument script itself.

Topaz -I did not correctly handle echo when error occurred

Using the -I option to Topaz is designed to allow input of an initialization script. With this option, echo is suppressed unless an error occurs; when an error occurs, the entire output is printed, but passwords are redacted.

- When the script contained an explicit EXIT and an error occurred, the echo output was not printed, so specifics of the error were not available for error diagnosis. (#51544)
- With nested scripts when an error occurred, the output was printed, but the password was not redacted in all cases. (#51545, #51546)

ExitClientError handling

An instance of ExitClientError can be signalled when the application wants topaz or other GCI client to exit with a specific error status.

Now, if ExitClientError is signalled to the GCI of an interactive linked topaz process, and the linked topaz was configured with GEM_LISTEN_FOR_DEBUG=true in a configuration file or on the command line, or if System listenForDebugConnection was executed, then topaz will stop at the command line, to allow the ExitClientError to be debugged, and not exit.

Automatic resultcheck

If the environment variable GS_TOPAZ_AUTO_RESULTCHECK is defined in topaz's environment, then result checking is automatically enabled, equivalent to DISPLAY RESULTCHECK. See the help text for DISPLAY RESULTCHECK for more details.

New superdoit variant for topaz scripting

Superdoit script variants previously were designed to execute Smalltalk code. A new variant, topaz, has been added to allow executing topaz expressions as well as embedded Smalltalk code.

The following examples have been added in $GEMSTONE/examples/superDoit:

```
error.topaz
simple.topaz
template.topaz
```

These scripts accept and use topaz command line arguments, and the contents can include any topaz commands. You may used default or explicit .topazini, or embed credentials in the script itself.

With topaz scripts, to avoid echoing the entire output, it is recommended to use the **-q** command line argument, and use GsFile `gciLogServer:` for any output intended for the user.

New environment variable GS_TOPAZ_ERROR_LOG_WITH_PID

If the environment variable GS_TOPAZ_ERROR_LOG_WITH_PID is defined in the environment in which topaz is started, a `topazerrors.log` file is instead named `topazerrors`*PID*`.log`.

10. Control over log file names

Log file names for all GemStone processes can now be composed using patterns. Previously, NRS #log: directives provided some patterns for the Gem log. Now, patterns can be applied to Stone, Netldi, and other system log files and X509-Secured GemStone gem log files; and additional pattern elements have been added. The new patterns are compatible with, and usable in, NRS.

If log file patterns are not used, log file names are handled as in previous releases.

Table 1.1 Supported log file and NRS pattern directives

format	meaning	limitations/requirements
%D	In NRS only; placeholder for startnetldi -D argument, in logs for processes started by Netldi.	%D and %%D are omitted in Stone and Stone child process logs, Netldi and LogSender/Receiver log files.
%H	Home directory	Only allowed as be the first element.
%M	Host machine name	
%N	Executable or script base name; e.g. stoned, netldid, gemnetobject	Either %N or %T is required in GEMSTONE_CHILD_LOG_PATTERN
%P	PID of the process	Required in GEMSTONE_CHILD_LOG_PATTERN.
%p	PID of the process's parent.	
%Q	listening port number	Only applies the -l argument to startlogsender, startlogeceiver, and startnetldi; otherwise, will be empty in the log file name.
%S	Stone or Netldi name; e.g. gs64stone, gem	For the Stone's child logs, this resolves to the name of the Stone. Not resolved in X509-Secured gem logs.
%T	Process type; e.g. stone, netldi, pcmon, gem	Either %N or %T is required in GEMSTONE_CHILD_LOG_PATTERN. Not resolved in X509-Secured gem logs.
%U	UNIX user name	
%%q	Timestamp milliseconds (0-999). Gemstone-specific; not part of strftime specification.	
other formats starting with %%	strftime(3) formats, for example, %%H for timestamp hours. Note that strftime %%D is not supported, to avoid issues with NRS %D.	
%%%	Literal '%'	NRS uses %% as the escape for literal %, which remains supported.

The Stone's child processes and threads that produce logs, including shrpcmon, symbol gem, reclaim gem and admin gem, cache warmer, openId, and login log, inherit their pattern from the Stone's log pattern provided by the **-l** argument or the $GEMSTONE_LOG environment variable.

This can be overridden by the new environment variable GEMSTONE_CHILD_LOG_PATTERN. This controls the log name for the Stone's child process/thread logs, but does not affect the Stone's log file name.

%P and either %N or %T are required in GEMSTONE_CHILD_LOG_PATTERN, to avoid multiple processes writing to the same log file or unrecognizable log file names. If either is missing, the log pattern is ignored and the default log file name is used. A message is written to the log file in this case.

If %Q (listening port) is used in the Netldi log file name, the value of -P command line parameter will be used, or a port specified in $GEMSTONE_NRS_ALL. If these are not available, a port assigned in the services database will be u sed. If the Netldi uses a random listening port assigned by the OS, then it is not possible to include the port number in the log file name; it will be omitted.

A new man page, logfilenames(5), has been added, as well as a short summary of formats in the -h output from processes that accept the -l argument.

Examples

Example 1. Stone log filename only

Starting the stone with:

```
startstone -l %S-%T-%%Y-%%m-%%d-%%H:%%M:%%S.log
```

Produces system log files with the names such as the following. Note that log names, other than the stone's log, are the default, since the pattern does not include %P.

```
gs64stone-stone-2026-01-07-14:44.log
gs64stone_390702reclaimgcgem.log
gs64stone_390703symbolgem.log
gs64stone_390671pcmon.log
gs64stone_login_2026-01-07-14:44:08.974.log
```

Example 2. GEMSTONE_CHILD_LOG_PATTERN

Starting the stone with:

```
EXPORT GEMSTONE_CHILD_LOG_PATTERN=GS_%%Y-%%m-%%d-
   %%H:%%M:%%S_%T_%P.log
startstone -l GS_%S_%%Y-%%m-%%d.log
```

Produces the log files with names such as the following. The child logs use the specified format.

```
GS_gs64stone_2026-01-07.log
GS_2026-01-07-14:25:14_pcmon_389149.log
GS_2026-01-07-14:25:14_reclaimgcgem_389180.log
GS_2026-01-07-14:25:14_symbolgem_389181.log
GS_2026-01-07-14:25:14_login_389147.log
```

Example 3. Gem log filename

Starting the netldi with:

```
startnetldi -g -l $GEMSTONE/logs/%T%Q.log -D $GEMSTONE/logs/
   34567
```

and in topaz RPC, logging in with:

```
set gemnetid !#netldi:34567#log:%N_%%Y%%m%%d-
   %%H:%%M:%%S.%%q_%P.log!gemnetobject
```

Writes the Netldi and Gem logs to the directory $GEMSTONE/logs/ (which must exist, or startup and login will fail), with the following log file names.

```
$GEMSTONE/logs/netldi34567.log
$GEMSTONE/logs/gemnetobject_20260107-13:46:39.238_385695.log
```

11. Configuration parameter changes

Remote cache reconnect over unreliable network

This release includes improved handling of remote caches over unreliable networks. (#51593). Note that 3.7.5 also includes the fix for bug #51839 which was present in the 3.6.9 LD release.

A new configuration parameter has been added to automatically attempt a reconnect after a remote cache connection timeout, rather than terminating the remote cache and the sessions on that cache.

Added configuration parameter

STN_REMOTE_CACHE_RECONNECT_TIMEOUT

Maximum time in seconds that the page manager thread in stone will wait for a remote cache that has disconnected to reconnect. This setting does not apply to remote X509 caches, which are always terminated when STN_REMOTE_CACHE_PGSVR_TIMEOUT expires.

Note that the Stone makes automatic adjustments to synchronize parameters:

- The value of STN_REMOTE_CACHE_RECONNECT_TIMEOUT is automatically increased if it is configured to less than 2.5 * STN_REMOTE_CACHE_PGSVR_TIMEOUT.
- The value of STN_GEM_LOSTOT_TIMEOUT is automatically increased if it is configured to less than 1.3 * the adjusted value of STN_REMOTE_CACHE_RECONNECT_TIMEOUT.

A value of -1 disables reconnect behavior; with this setting, the expiration of STN_REMOTE_CACHE_PGSVR_TIMEOUT causes immediate shutdown of that cache and sessions on that cache.

The default is 2.5 * the STN_REMOTE_CACHE_PGSVR_TIMEOUT, which by default is 15 seconds. With both settings at default, the reconnect timeout is 37 seconds.

Runtime equivalent: #StnRemoteCacheReconnectTimeout (requires SystemControl privilege)
Default: 2.5 * STN_REMOTE_CACHE_PGSVR_TIMEOUT
Max: 3600,
Min: -1

Added cache statistics

The following statistics have been added:

RemoteCacheReconnectCount (Stone)
Number of successful reconnects by remote caches after a network timeout.

RemoteCacheTimeoutCount (Stone)
Number of network timeouts seen by stone page manager thread when communicating to remote cache pgsvrs.

Statmonitor data file location for GEM_STATMONITOR_ARGS

When a linked login on a remote node triggers remote cache monitoring due to the configuration parameter GEM_STATMONITOR_ARGS, and the **-f** argument was a relative file path, the statmonitor data file was put into the home directory of the Unix user of the linked process.

Now, the data file is written to the working directory of the linked process that triggered the remote cache creation.

Removed Configuration Parameter

The obsolete configuration parameter GEM_TEMPOBJ_START_ADDR has been removed.

12. Netldi Changes

NetLDI reports number and duration of PAM calls

New arguments have been added to **startnetldi** to enable reporting of the number and duration of calls to PAM for authentication, and to trigger more detailed reporting if the duration takes more than a configured duration threshold. Reports are not printed if there has been no activity during the preceding interval.

This feature is only valid when the NetLDI is authenticating the user via PAM, that is, not in guest mode (**-g**); and not for the X509-secured Netldi (**-S**).

The added **startnetldi** arguments are:

-y *printIntervalSecs*
: Interval in seconds to print statistics on calls to PAM to the log file. Not compatible with **-g** or **-S**. When the **-d** flag (to enable debugging) is used and **-y** is not specified, a default of **-y**300 is inferred.

-z *printThresholdMs*
: Print detailed statistics on calls to PAM when any PAM call takes longer than *printThresholdMa* milliseconds. Not compatible with **-g** or **-S**.

Examples

Example of an interval report:

```
--- 10/06/25 12:10:13 PDT
PAM Performance: Calls total: 55, ok: 52, fail: 3, Time avg:
      8 ms, high: 103 ms, low: 0 ms, Threads: 1
```

Example of a threshold report:

```
--- 10/06/25 12:18:27 PDT PAM call exceeded threshold of 5 ms
Duration: 17 ms, success: 1, user: 'gsadmin'
Function         Duration calls ok fail retry err
================================================
getpwnam         0        1     1  0    0     0
pam_start        1        1     1  0    0     0
pam_authenticate 14       1     1  0    0     0
pam_end          0        1     1  0    0     0
================================================
```

Example of a report when an error occurred:

```
--- 10/06/25 12:29:25 PDT PAM call exceeded threshold of 5 ms
Duration: 6 ms, success: 0, user: 'gsadmin'
err msg: 'Password validation failed for user gsadmin,
   pam_authenticate error:7, Authentication failure'
Function         Duration calls ok fail retry err
================================================
getpwnam         0        1     1  0    0     0
pam_start        2        1     1  0    0     0
pam_authenticate 4        1     0  1    0     7
pam_end          0        0     0  0    0     0
================================================
```

Netldi added ability to record statistics for VSD

New arguments have been added to **startnetldi** to allow it to record statistics to a vsd-compatible statistics data file. This process does not use **statmonitor**, and does not attach to the shared page cache; the statistics are written directly to disk by the Netldi.

The interface to manage writing statistics is basic; most of the many **statmonitor** features have not been implemented for the Netldi. In particular, note that statistics data will be written to a single file for the lifetime of the Netldi. To manage these statistics data files, you will need to regularly restart the Netldi, providing a new **-f** filename argument.

The Netldi-generated statistics data files includes Netldi and OS system statistics only. The Netldi internal stats (see “Added Netldi-specific statistics” on page 37) are not accessible using **statmonitor** or from Smalltalk methods.

-f *statFileNameOrDir*

A file name or directory for writing statistics data. Requires **-j**. Also enables writing a PAM summary (**-y**) on Netldi shutdown only.

If a file name is given, the file must not exist. Files with a .gz suffix will be gzip compressed; files with a .lz4 suffix will be lz4 compressed. Otherwise file is not compressed.

If a directory is given, it must exist; with a directory argument, a file with a default file name containing the Netldi name and a timestamp with a .gz suffix will be created.

-j *intervalSecs*
Sample interval in seconds to write statistics data to the statistics data file. Requires **-f**.

Added Netldi-specific statistics

The following statistics are collected only when the startnetldi cache statistics feature is enabled; these are not taken from the shared cache and are not accessible using statmonitor or from Smalltalk methods.

In addition to these statistics, the standard OS statistics for the netldid process and Linux system statistics are also recorded by the Netldi.

ClientRequestAvgTime
Average time in microseconds that a client request takes to process.

ClientRequestCount
Total number of client requests processed by netldi.

ClientRequestFailCount
Number of failed client requests processed by netldi.

ClientRequestOkCount
Number of successful client requests processed by netldi.

ClientRequestTotalTime
Total amount of time in milliseconds netldi has spent processing client requests.

ClientThreadsActiveCount
Number of client threads active in the netldi.

ClientThreadsConfigCount
Maximum number of client threads the netldi is configured for.

ClientThreadsWaitAvgTime
Average time in microseconds spent waiting for a free client thread.

ClientThreadsWaitCount
Number of times a request waited for a free client thread.

ClientThreadsWaitTotalTime
Total amount of time in milliseconds spent waiting for a free client thread.

ForkAvgTime
Average amount of time in microseconds spent in a call to fork.

ForkCount
Total number of forks performed by netldi.

ForkFailedCount
Number of failed forks performed by netldi.

ForkOkCount
Number of successful forks performed by netldi.

ForkTotalTime
Total amount of time in milliseconds spent in calls to fork.

PamCallsAvgTime
Average duration in microseconds all calls to PAM.

PamCallsCount
Number of calls to PAM

PamCallsFailedCount
Number of failed calls to PAM

PamCallsFastestTime
Duration in milliseconds of the fastest call to PAM.

PamCallsOkCount
Number of successful calls to PAM

PamCallsSlowestTime
Duration in milliseconds of the slowest call to PAM.

13. Cache Statistics Changes

Netldi statistics

The Netldi now supports collecting cache statistics, and a number of Netldi-specific statistics have been added. See “Netldi added ability to record statistics for VSD” on page 36.

Aliases for GcAdmin sessions stats offsets were incorrect

Aliases allow meaningful names to be attached to SessionStat00-SessionStat48. The aliases had become incorrect for the GcAdmin gem, so the displayed meaningful names were incorrect. Note this fix is in VSD v5.6.5 that is bundled with the GemStone distribution, not in the GemStone/S server itself.

New stat for total Gem memory use

GemMemoryFootPrintKb (Gem)
Approximate total memory footprint of allocated temp obj memory.

Added and removed statistics for huge memory pages

Previously, Linux system statistics for HugePages did not differentiate between different memory page sizes. These old statistics have been replaced by new ones that include the page size in the name.

The following statistics have been removed:

HugePagesTotalKB
HugePagesFreeKB
HugePagesRsvdKB
HugePagesSurpKB
HugePageSizeKB

The following statistics have been added. Note that the unit of these statistics is pages, not KB/MB.

HugePages2MbTotal
The total number of 2 Mb huge memory pages configured.

HugePages2MbFree
The number of free 2 Mb huge memory pages.

HugePages2MbRsvd
The number of reserved 2 Mb huge memory pages.

HugePages2MbSurp
The number of surplus 2 Mb huge memory pages.

HugePages1GbTotal
The total number of 1 Gb huge memory pages configured.

HugePages1GbFree
The number of free 1 Gb huge memory pages.

HugePages1GbRsvd
The number of reserved 1 Gb huge memory pages.

HugePages1GbSurp
The number of surplus 1 Gb huge memory pages.

Linux ARM supports, in addition to 2MB and 1 GB huge pages, the additional huge memory page sizes 64 KB and 32 MB. The following are reported on Linux/ARM only:

HugePages64KbTotal
The total number of 64 Kb huge memory pages configured.

HugePages64KbFree
The number of free 64 Kb huge memory pages.

HugePages64KbRsvd
The number of reserved 64 Kb huge memory pages.

HugePages64KbSurp
The number of surplus 64 Kb huge memory pages.

HugePages32MbTotal
The total number of 32 Mb huge memory pages configured.

HugePages32MbFree
The number of free 32 Mb huge memory pages.

HugePages32MbRsvd
The number of reserved 32 Mb huge memory pages.

HugePages32MbSurp
The number of surplus 32 Mb huge memory pages.

14. Changes in GCI errors

Added Errors

The error corresponding to LogRotationNotification has been added:

```
RT_ERR_SIGHUP/6026
```

An error type has been added to allow better error reporting for loss of connection during a multithreaded operation.

```
FATAL_ERR_DURING_MT_OP 4011
```

The following errors, related to internal errors in X509-secured GemStone, have been added:

```
ERR_REBUILD_SCAVENGABLE/4022
ERR_DEPMAP_FAILURE/4023
ERR_OT_AUGMENT_FAILURE/4024
ERR_COMPOSE_CR_FAILURE/4025
```

Removed Errors

The long-obsolete error `GS_ERR_SHRPC_INVALID_CONFIG/4011` has been removed.

15. X509-Secured GemStone Changes and Bug Fixes

startHostAgent log file specification

Previously, the HostAgent log file was always written in the directory specified by the **startnetldi -D** argument, with the name hostagent-*stoneName-remoteHost-PIDStoneHost*.log.

The **startHostAgent** script now includes the **-l** argument. allowing the name and location of the log file to be specified. The patterns described “New utility script to validate configuration files” on page 28 can be used.

File name precedence:

1. The **-l** argument to **starthostagent**
2. GEMSTONE_NRS_ALL in the environment where **starthostagent** is executed.
3. GEMSTONE_NRS_ALL in the environment where **startnetldi** was executed, or the **startnetldi -X** argument NRS
4. The default, hostagent-*stoneName-remoteHost-PIDStoneHost*.log

Reconnect script for mid-cache HostAgent

On a X509-secured mid-level cache, if the HostAgent that is running on that mid-level cache goes down, but the cache itself is still running, it is possible for the HostAgent to be restarted and reconnect.

To simplify this, a reconnect script is now automatically generated by the Netldi when a cache becomes a mid-level cache, with the name `restartMidHostAgent_`*stoneName*`.sh`. This is a no-argument script that must be manually executed.

This script is written to the directory specified by the **startnetldi -D** argument.

X509 mid-level cache startup now fails without required configuration

Now, if the NETLDI_PORT_RANGE and other required configuration parameters are not supplied, the mid-level cache in an X509 configuration will not start up. (#51552)

GsX509Parameters now support quietLogin flag

The following methods have been added:

```
GemStoneX509Parameters >> clearQuietLogin
GemStoneX509Parameters >> quietLoginFlag
```

X509-Secured GemStone bug fixes

Mid-level caches failed to improve performance in X509 configuration

In X509-secured GemStone, if the page lookup on a leaf host missed, the leaf host did not attempt to read from the mid cache. (#51534).

Mid-level cache warming failed to completely warm the cache

The pusher threads in the HostAgent that warm the mid-level cache did not scan the Stone's entire cache, resulting in an incompletely warmed cache. (#51558)

X509-secured GemStone did not calculate default large page size

If SHR_PAGE_CACHE_LARGE_MEMORY_PAGE_SIZE_MB is not set, and large pages are enabled using SHR_PAGE_CACHE_LARGE_MEMORY_PAGE_POLICY, the default large memory page size on the given host should be used. This was not being done in an x509-Secured GemStone system. (#51423)

HostAgent >> startLeafCache did not respect timeout

It was possible for the HostAgent to create a commit record backlog when the NetLDI on a leaf host failed to respond. The underlying code was not respecting the timeout provided. (#51549)

SEGV in HostAgent after session logout during heavy use of InterSessionSignal

There is a risk of a SEGV in the HostAgent, when the HostAgent closes the SSL connection to the Gem during periods when the Gem can be receiving InterSessionSignals, or when the HostAgent is forwarding a sigAbort to an x509 Gem. (#51564)

Fatal error in HostAgent can leave stuck commit token

When a thread in the HostAgent had a fatal error, it did not correctly handle the exit to ensure the Stone knew the session had disconnected and release the commit token and other resources. (#51566)

Error in mid-cache HostAgent if connection to a leaf host was lost

The mid-cache HostAgent did not correctly handle the case where a leaf cache connection was lost. (#51538)

Using -E could have resulted in starthostagent failing to connect

When using -E, the netldi name generated could be malformed, resulting the starthostagent failing to connect to the remote netldi for a leaf host. (#51643)

After mid cache HostAgent died, Gem errored

Gems do not tolerate the lost of a connection to a mid-level cache; if the mid-level cache Host Agent dies, the Gem may error with "lost connection to pgsvr". (#51622)

Socket leak in HostAgent on mid-level cache node

In an X509-secured GemStone configuration, the Host Agent on a mid-level cache did not properly close its end of the socket to the HostAgent on the Stone's node for that mid-level cache node. This resulted in running out of file descriptors on the mid-level cache node. (#51619)

SEGV in HostAgent on mid cache upon no more slots free condition

When the mid-level cache Host Agent had no available slots and another Gem attempted to connect, it crashed. (#51644).

Now, the connection to the mid-level cache Host Agent will fail, but not affect the HostAgent. The Gem will run without using the mid-level cache and may see performance issues.

stophostagent during remote cache warming may leave warmer gem running

When stophostagent is executed shortly after starthostagent, while an X509 remote cache is configured with cache warming and the cache warming is not complete, the warmer gem fails to exit when the remote cache is shut down. (#41829)

Chapter

2 Bug Fixes

The following bugs were present in v3.7.4.3 and are fixed in this version.

Risk of corruption after restart during large commit record backlog

If the stone is stopped or shuts down while there is a commit record backlog, on restart it disposes of the commit record backlog. If there is a low free space condition in the extent space while it is disposing of the commit record backlog that was found on restart, such that the Stone attempts to grow the extents, this can cause repository corruption. This is a risk mainly when there is a very large commit record backlog and when there are commits before the disposal process has completed. (#51768)

After tranlog full condition, risk of missing logout record

A case was encountered where after a series of tranlog full conditions during which a session terminated in the middle of a commit, the Stone did not recognize that the session was gone, and the oldest tranlog required for restore remained at the one current at the time this session started the commit. No information was lost; the ability to restore from backup, however, became dependent on old tranlog retention. Deficiencies in the Stone's buffering of these records have been fixed. (#51708)

If Stone killed while opening new tranlog, restart may fail

There is a timing window when the Stone is opening a new tranlog but before it writes the tranlog's root record, during which if the Stone is killed, restart will fail with "readLastRecord fileId ... cannot find Root record". (#51700)

Risk of SEGV if SIGTERM during commit

When a process is in the middle of a commit and receives a SIGTERM, it may SEGV. (#51604)

Printing the Smalltalk stacks did not handle fatal errors properly

When printing the smalltalk stack using pstack or kill -USR1, if the Gem had a fatal error, such as termination due to a lostOT, this was not handled correctly; gdb may fail to detach, hanging the gem. (#51478).

SIGUSR1 is also now ignored if sent less than 2 seconds since the previous such signal.

Upgrade cleared Deprecated action

The Deprecated class supports automatically writing to a log or erroring if a deprecated method is executed, as well as doing nothing (the default). Upgrade cleared this status to remove a non default deprecated action. Now, upgrading a repository with a non-default deprecated action preserves the deprecated action (#51520)

GsObjectSecurityPolicy dynamicInstanceVariables failed on an upgraded repository

Sending `#dynamicInstanceVariables` to an instance of GsObjectSecurityPolicy in a repository upgraded from v3.5.6 or earlier reported a corrupt object error. (#51411).

SymbolGem may get commit conflicts in administrative update during user login

When a User with password features enabled logs in, triggers the SymbolGem to update the user's security data, for example, with the last login time. If an administrative change was made at the same time in another session that also updated security data, this may cause a commit conflict in the SymbolGem. Previously, the SymbolGem restarted. Now, the SymbolGem aborts the transaction. The login succeeds, but this user's security data is not updated by the login, which could affect account aging or other limits. (#51611).

When stopstone was prompting for input, it did not respond to Control-C

When stopstone is executed without arguments, it prompts for the Stone name, user name, and password. While in the prompt, the process did not respond to control-C to terminate. (#51428)

ProfMonitor profiling object creation may crash gem

When ProfMonitor monitors object creation (using the #objCreation option or report), and there is a scopes overflow, the Gem becomes unusable and will eventually either hard hang the Gem, or get a HostCallDebugger. (#51822)

Small memory leak on host password validation failure

There is a small memory leak in validating a host password when the result is not PAM_SUCCESS. (#51597)

Risk of lostOT during reclaimAll

`Repository >> reclaimAll` waited for an increasing time when no progress in reclaim was occurred. This created a risk of a lostOT when the ReclaimGem was committing at a high rate. (#51506)

In-memory garbage collection not aggressive enough

Cases have been fixed in which the in-memory GC of perm_gen and code_gen were not aggressive enough, resulting in out of memory issues earlier than necessary. (#51577)

Warming leaf caches could have caused commit record backlog

Cache warming runs in transaction, and when performing a large amount of warming over a slower connection, it could result in a commit record backlog. Now, while still running in transaction, leaf cache warming monitors the commit record backlog and aborts if necessary. (#50103)

With GEM_REMOTE_COMMIT and mid-cache, commits broken

When a remote Gem is using a mid-level cache, and is configured with:

GEM_PGSVR_USE_SSL = true;
GEM_REMOTE_COMMIT=true;

then commits are broken, due to in-memory corruption in the pgsvr for that session. GEM_REMOTE_COMMIT was added in v3.7.4.

Performance of GsSecureSocket secureAccept and close

The performance of `GsSecureSocket >> secureAccept` and of `GsSecureSocket >> close` were unreasonably slow. (#51782)

Copydbf on backup could miss reporting oldest tranlog required

copydbf on a programmatic .dbf backup reports the oldest tranlog required to restore this backup. If the oldest tranlog itself required an earlier tranlog, that was not correctly reported by **copydbf** of the backup file. This may be the case when a single commit spans two tranlogs. The oldest tranlog required for a tranlog is correctly reported by **copydbf** on that tranlog. (#51703)

Issues related to Hot Standby

Risk of Gem protocol error if session is signaled after logsenderSessionId

After executing `logsenderSessionId`, interrupt handling may sometimes result in a protocol error, 'Unexpected packet received from Stone'. (#41448)

stoplogreceiver may have silently failed to stop the logreceiver

It is possible for **stoplogreceiver** to return 0 and does not print an error, but to leave the logreceiver process running. (#51472)

Restarting tranlog restore after a stopContinuousRestore has risk of error requiring Stone restart

If `stopContinuousRestore` is executed while the Stone thread running recovery is processing a large work queue, there is a risk that a subsequent `continuousRestoreFromArchiveLogs:`, or `restoreFromArchiveLogs:`, may fail with a detected Fork in Time error. To continue, the Stone must be restarted. (#51548

GsFile issues

GsFile opening with mode 'ab+' did not allow read positioning.

When a GsFile is opened for read and append, the mode may be specified either 'a+', 'ab+', or 'a+b'. The specification 'ab+' was not handled correctly; the resulting file was open for append but could not be positioned for read. (#51525)

GsFile open with nil path or mode failed to set error message

GsFile operations that error return nil, rather than signalling an error, and put the error message in the server or client error buffer. This was not done for nil argument cases.

GciTsLogin() not thread safe from a GCI main program

There may be initialization errors if multiple C threads in a GCI program call GciTsLogin at the same time. (#51600)

Parser did not always correctly handle true, false, nil, and self as selectors

The ANSI standard states that #true, #false, #nil, #self and #super are disallowed as selectors. These can be used in GemStone, although they are not tested. The parser was unreliable in recognizing #true, #false, #nil, #self as selector tokens. (#51669)

Configuration file update could be incorrect when adding an extent

When an extent is added programmatically, the new DBF_EXTENT_NAMES and DBF_EXTENT_SIZES are automatically written to the config file. When the existing extent size was defined in units of GB, the programmatic argument was incorrectly interpreted as being in GB rather than MB. (#51605)

NRS precedence incorrect for remote Gem

A #dir: or #log: directive in the Stone's NRS in the login parameters for a remote Gem, did not correctly take precedence over the GEMSTONE_NRS_ALL provided to the remote Gem via the remote NetLDI. (#51507)

If listen on :: fails, Netldi will only listen on localhost

If the listening port provided to **startnetldi** already was in use by an existing socket, **startnetldi** succeeded, but was only listening on that port on localhost. (#51651)

Passivate of subnormal LargeIntegers was incorrect

Normally integers that are within the range of SmallIntegers are instances of SmallInteger; however, in some cases in upgraded applications, LargeIntegers may exist that are in the SmallInteger range. The passivated form was incorrect, and errored on activation. (#51625)

roundedHalfToEven sent to NaN or Infinity gets stack overflow

If `Number >> roundedHalfToEven` is sent to a kind of NaN or Infinity, it failed with a stack overflow. (#51658)

Symbol GC may leave internal buckets without back reference

After very large number of symbols are garbage collected, such that the internal buckets shrink from a large to a small object, the reference back to the KeyValueDictionary may be reset to nil. This is not expected to cause issues; it is caught by AllSymbols audit. It is recommended to audit AllSymbols after Symbol GC. (#51678)

Issues related to continueTransaction

Incorrect conflict results after a failed continueTransaction

When a continueTransaction fails, the methods that report on commit conflicts are incorrect. (#51720)

continueTransaction does not clear a dirty write lock

If an object has a write lock that is dirty, a successful continueTransaction was failing to refresh the view to allow a clean view of the lock. (#51647)

Restore from backup did not clear NotTranloggedGlobals

The NotTranloggedGlobals holds objects that are committed, but for which changes are not recorded in tranlogs. This was not getting cleared during restoreFromBackup. (#51672)

After mid-level cache restarted, gems may fail to connect

If a mid-level cache goes down and restarts via a Gem login, sessions may intermittently fail to reconnect to the mid-level cache. (#51664)

Deleting a UserProfileGroup may fail on upgraded repository with few security policies

A repository upgraded from v2.1 or earlier with fewer than 20 object security policies has nils in the SystemRepository (the collection that holds object security policies) for slots under 20. This causes `UserProfileGroup >> deleteGroup:` to fail. (#51518)

systemLocksDetailedReport did not handle in-logout sessions

`System class >> systemLocksDetailedReport` could have reported "session does not exist", if a session logs out during execution of the method. (#51474)

Seaside scripts invoked deleted method

The method `Breakpoint>> trappable:` was removed in recent releases. This method was in use by several Seaside scripts. The seaside scripts in `$GEMSTONE/seaside/bin`: **startMaintenance**, **startMaintenance30**, **startSeaside_FastCGI**, and **startSeaside30_Adaptor**, have been updated. (#51711)

Incorrect FileReference printOn: output

The printed form of a FileReference previously omitted the @ and quotes, and thus could not be used as-is to recreate the FileReference.

PrimitiveNumber cache statistic incorrect for MFC

The statistic for PrimitiveNumber was recorded incorrectly for a process running MFC. (#51636)

Wrong error message for invalid TimeZone lookup

In some configuration, particularly on Windows, using TimeZone named: with an invalid path resulted in an incorrect error such as a GciTransportError or an MNU. (#51679)

Some Page Cache stats not correct on remote caches

Several cache statistics for the Gem and Shrpcmon for pages in or added to the cache are not correct for remote caches. (#51535)