


GemStone/S 64 BitTM

Release Notes

Version 3.7.1

March 2024



INTELLECTUAL PROPERTY OWNERSHIP

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

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

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

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

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

COPYRIGHTS

This software product, its documentation, and its user interface © 1986-2024 GemTalk Systems LLC. All rights reserved by GemTalk Systems.

PATENTS

GemStone software has been covered by U.S. Patent Number 6,256,637 "Transactional virtual machine architecture", 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**, **macOS**, and **Macintosh** are trademarks of Apple Inc.

Instantiations is a registered trademarks of Instantiations, Inc.

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

Raspberry Pi is a trademark of the Raspberry Pi Foundation.

RabbitMQ is a trademark of VMware, Inc.

Prometheus is a registered trademark of The Linux Foundation.

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

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

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

No separate Installation Guide is provided with this release. For instructions on installing GemStone/S 64 Bit version 3.7.1, or upgrading or converting from previous products or versions, see the Installation Guide for version 3.7.

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 3.7.1

Overview	9
Supported Platforms	9
Platforms for Version 3.7.1	9
GemBuilder for Smalltalk (GBS) Versions	10
VSD Version.	10
1.1 Administrative Changes.	11
Updated Library Versions.	11
Distribution changes	11
gcisup_ts.hf	11
GemStone now includes Rowan v3.x as well as Rowan v2.x	11
Linux on ARM now supported for production use.	11
Native code on Linux/ARM.	11
Native code on Mac	12
Preliminary Support for Raspberry Pi for Development.	12
Tranlogging Changes	12
Extent pregrow on ZFS file systems	12
Configurable location of services.dat lookup file	12
Added startnetldi -Q argument	13
startnetldi now requires port to match GEMSTONE_NRS_ALL	13
Solo logins temporary files renamed	13
Upgrade for customer-defined special classes	13
Simplified finding reference path methods	14
Added Repository analysis methods.	14
Backup and Restore Changes	15
Backup, Restore and other scans report progress	15
Performance improvement in restoring a zipped tranlog	15
restoreFromBackups:threads: now accepts filename	15
Hot Standby-related changes and bug fixes	15

Tranlog position may have been inconsistent between master and slave	15
failOverStatus now includes if logreceiver is connected	16
Avoid running out of stack while handling an AlmostOutOfStack error.	16
GEM_MAX_SMALLTALK_STACK_DEPTH maximum changed.	16
Utility Changes	16
startstone, pageaudit -W to disable cache warming	16
Added Topaz SET commands	16
gemnetobject script cleanup	17
GCI Changes.	17
GciTsAddOopsToNsc.	17
GciTsFetchNamedOops.	17
GciTsFetchVaryingOops	18
GciTsStoreIdxOops	18
GciTsStoreNamedOops.	18
1.2 GemStone Smalltalk Changes	19
Instance-variable specific dbTransient feature.	19
Behavior class definition changed	20
SessionTemps no longer stubs contents.	20
TreeDictionary and TreeSet added	20
Now possible to step into accessor methods.	20
ProfMonitor object creation profiling enhancements and bug fixes	20
Object creation profiling now includes profiling of some kernel classes	20
ProfMonitor object creation profiling reported sample count incorrectly.	21
Avoiding performance impact of process host statistics	21
FileSystem changes and bug fixes	21
ZnStream, ZnCharacterEncoder hierarchy changes; ZnObject removed	21
Manual flush no longer required for FileReference write streams	22
Directory creation errors were FsEACCES, now FilePermissionDenied	22
Added FileSystem-related methods	22
FileSystem iteration over NFS-mounted directories may miss reporting some files	22
Symlinked directories and files omitted from results	22
GsTsExternalSession forkAndDetachBlock:	22
Added GCI flag	23
GsTestCase added methods	23
IndexManager new methods.	23
Reporting tranlog positions	23
Other added and moved methods.	23
allInstances	23
Array >> pairsDo:	23
Behavior >> extractSelector:	23
BlockClosure >> doWhileFalse:	24
CharacterCollection >> substringsSpace	24
Duration >> asMilliseconds	24
Integer >> >> shiftAmount.	24
SmallInteger >> permutedHashA:c:d:	24
String >> evaluateInContext:.	24

System class >> waitForDebugWithAbort	24
Removed methods	25
1.3 Changes in Encrypted extents, tranlogs, and backups	26
Support for up to 24 encryption keys	26
System class >> repositoryPublicKey renamed	26
Path requirements and -K	26
Utility Changes related to encrypted DBFs	27
copydbf argument changes	27
copydbf may have prompted for passphrase	27
copydbf -X, -Y output filenames did not include extension	27
updatesecuredbf argument changes	27
updatesecuredbf output modified	28
startstone argument changes	28

Chapter 2. Bug Fixes

Sending #value to special resulted in SEGV.	31
startstone recovery may fail with tranlogs on raw partitions	31
Stone may shutdown on checkpoint when tranlog disks are full	31
Garbage collection issues	31
Performance regression in markForCollection	31
markForCollectionWait: with argument of -1 gets prim failure	32
reclaimAll may fail to reclaim some dead	32
C mutex deadlock possible from ProfMonitor/SIGUSR1	32
NetLDI issues	32
NetLDI shutdown timing issue resulted in unable to restart	32
netldid executes without \$GEMSTONE but does not function	32
GsSecureSocket instance method did not properly used chained certificate file	32
GsSocket >> writeWillNotBlockWithin: does not handle nil returned by writeWillNotBlock	33
Unicode string copyReplaceAll:with:, insertAll:at:, broken with String arguments	33
Collection issues	33
Array >> replaceFrom:to:with:startingAt: may not have copied elements correctly	33
Dictionary become: does not update internal variables	33
Numeric and Time related issues.	33
Strings with leading '-0.' lose sign when converted using asNumber or Number fromString/Stream:	33
Duration new created an uninitialized instance.	33
Issues related to upgrade	34
upgradeImage reports success even if upgrade fails	34
Excessive attempts to start AdminGem during upgrade.	34
copydbf returns wrong version for secondary extents after upgrade.	34
Session methods must be disabled for upgrade.	34
Upgrade created instances of ObsoleteMetaClass	34

GsNMethod >> recompileFromSource loses method category 34
perform:withArguments: may crash on bad arguments 34
Spurious OutOfMemory when in-memory pom_gen scavenge fails 34
Fileout format reverted to older style in v3.7 35
GsExternalSession >> executeString: fails if attempting to return an empty String .
35
defaultGemNRSFromCurrent may be incorrect 35
GLASS/Seaside auto-migrate may error after GC operation 35
ensure blocks in superDoit scripts do not execute on error 35
Issues with GsProcess terminate. 35
 Overlapping terminate request semantics 35
 SEGV when a terminated GsProcess is sent terminate 36
Gem failed to respond to SIGTERM on 1-core VM 36
CodeModificaton required to list Pragmas with session methods enabled. . . . 36
waitstone may return before Stone is ready to accept logins. 36
Array with:* primitive failure error reports incorrect. 36

Release Notes for 3.7.1

Overview

GemStone/S 64 Bit™ 3.7.1 is a new version of the GemStone/S 64 Bit object server. Version 3.7.1 includes a number of new features, feature enhancements, and bug fixes.

These Release Notes include changes between the previous version of GemStone/S 64 Bit, v3.7, and v3.7.1. If you are upgrading from a version prior to 3.7, review the release notes for each intermediate release to see the full set of changes.

The Installation Guide has not been updated for this release. For installation, upgrade and conversion instructions, use the Installation Guide for version 3.7.

Supported Platforms

Platforms for Version 3.7.1

GemStone/S 64 Bit version 3.7.1 is supported on the following platforms:

- ▶ Red Hat-compatible Linux 7.9, 8.9, and 9.3, and Ubuntu 20.04 and 22.04, on x86; and Ubuntu 20.04 on ARM
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.
- ▶ AIX 7.1 and 7.2
- ▶ macOS 14.3 (Sonoma) and 11.7.10 (Big Sur) on x86 and Apple silicon (ARM)

Note that GemStone/S 64 Bit v3.7 and later on Linux are built using compilation features that are not available on older CPUs (more than about 10 years old); v3.7.1 will not run on these CPUs, regardless of the Linux OS version.

For more information and detailed requirements for each supported platforms, please refer to the *GemStone/S 64 Bit v3.7 Installation Guide* for that platform.

GemBuilder for Smalltalk (GBS) Versions

The following versions of GBS are supported with GemStone/S 64 Bit version 3.7.1:

GBS/VW version 8.7

VisualWorks 9.1.1 32-bit and 64-bit
<ul style="list-style-type: none"> ▶ Windows 10 ▶ RedHat ES 7.9, 8.9, and 9.3; Ubuntu 20.04 and 22.04

GBS/VA version 5.4.7

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

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

VSD Version

The GemStone/S 64 Bit v3.7.1 distribution includes VSD version 5.6.1. This is the same version of VSD that was included with the previous release, v3.7.

VSD 5.6.1 is included with the GemStone distribution, and can also be downloaded as a separate product from <https://gemtalksystems.com/vsd/>

1.1 Administrative Changes

Updated Library Versions

The version of openssl has been updated to 3.0.13

The version of libssh has been updated to 0.10.6

The version of openldap has been updated to 2.6.6

The version of zlib has been updated to 1.3

The version of MIT Kerberos has been updated to 1.21.2

The version of cJSON has been updated to 1.7.17

The version of curl has been updated to 8.5.0

The version of zoneinfo has been updated to 2023d

The version of prometheus-cpp has been updated to 1.2.0

The version of AWS has been updated; azure-sdk-cpp to version 1.10.3, and aws-sdk-cpp to version 1.11.243

Distribution changes

gcisup_ts.hf

The file `$(GEMSTONE)/include/gcisup_ts.hf` is now included; this provides headers for functions supporting Small* classes and the Special56BitN classes.

GemStone now includes Rowan v3.x as well as Rowan v2.x

GemStone v3.7.1 includes the infrastructure to support both Rowan 2.x and Rowan 3.x. The existing directories, files and scripts continue to apply to Rowan 2.x, as in v3.7.

A new main level directory, `$(GEMSTONE)/rowan3/`, holds the Rowan v3.x scripts.

The Rowan extent included in the distribution in previous releases, `$(GEMSTONE)/bin/extent0.rowan.dbf`, is a Rowan v2.x extent. A new extent, `$(GEMSTONE)/bin/extent0.rowan3.dbf`, is now also included.

`$(GEMSTONE)/examples/superDoit/` now includes examples and templates for superDoit scripts using Rowan 3.x; existing examples and templates continue to use Rowan 2.x

The Rowan3 is branch masterV3.1; the Rowan2 branch is masterV2.6.

Linux on ARM now supported for production use

Previously, Linux on ARM was supported only for development. Sufficient testing has been done that Linux on ARM is now certified for production use.

Native code on Linux/ARM

In earlier releases, Linux on ARM ran in interpreted mode only. Native code is now enabled for Linux on ARM.

Native code on Mac

As in v3.7, native code is not available for Mac on silicon (ARM).

Recent Mac versions may allocate virtual addresses for temporary object memory more than 32bits offset away from code for libgcilnk.so, resulting in native code errors. To avoid such issues, native code is disabled by default for Mac on x86 (GEM_NATIVE_CODE_ENABLED = 0).

Preliminary Support for Raspberry Pi for Development

Some testing has been done on Raspberry Pi, and this platform is now supported for development only, on Raspberry Pi 5 with 8GB memory, running Debian GNU/Linux 12 (bookworm).

Tranlogging Changes

Some additional incremental (rather than complete) tranlogging has been added for operations on small objects.

ScavengablePages are now logged with STN_TRAN_LOG_DEBUG_LEVEL=1.

Extent pregrow on ZFS file systems

In earlier releases, extent pregrow on ZFS file systems was very slow, since the OS call to allocate disk (fallocate()) does not work, and the backup of appending blocks of zeros is not performant, and also is inappropriate for a COW (copy on write) file system. In this release, the preallocation is done using ftruncate().

Note that with ZFS or any COW (Copy On Write) file system, preallocation does not guarantee that writes to the file will not run out of space. Extent pregrow may not provide improved performance nor reserve disk space exclusively.

Configurable location of services.dat lookup file

The location and handling of services.dat is now configurable. `$GEMSTONE/sys/services.dat` is a services lookup file that is used:

- ▶ by the Stone, to looking administrative services that are only invoked by the Stone, such as starting up the GcGems.
- ▶ by the NetLDI, to lookup **gemnetobject** and other login service scripts, which are not used by the Stone
- ▶ by both the Stone and the NetLDI, to lookup services that support remote Gems.

With 3.7.1, there are changes in the way this file is used, and the requirements for the Stone to lookup administrative scripts and for the NetLDI to lookup scripts for login.

As in previous releases, if `$GEMSTONE/sys/services.dat` is present, the services (**runreclaimgem**, etc.) may have different names and be located in different directories, and the lookups in `$GEMSTONE/sys/services.dat` will be used for all Stone service lookups; missing service names or incorrect paths will cause Stone startup to fail.

Now, however, the Stone no longer requires `$GEMSTONE/sys/services.dat` to be present, as long as all the services (scripts) that would have been looked up in that file, are present with their distribution names in `$GEMSTONE/sys/`.

Likewise, the NetLDI does not require `$GEMSTONE/sys/services.dat` to be present, provided the services files (i.e. **gemnetobject** and other scripts) are present with their distribution names in `$GEMSTONE/sys/`.

If `$GEMSTONE/sys/services.dat` is present, this will be used for lookups.

NetLDI can also be provided with the path to a custom services lookup file, using the new **startnetldi -Q** argument. When provided, the services lookup file specified with the **-Q** argument will be used for all NetLDI lookups, and the `$GEMSTONE/sys/services.dat` will not be read by the NetLDI, although it will still be used by the Stone. In this case, the NetLDI will not use services with their distribution names that are present in `$GEMSTONE/sys/`, but for which there are no mappings in the **-Q** argument services lookup file.

Added startnetldi -Q argument

startnetldi has a new argument, **-Q pathToServicesLookupFile**, which allows you to pass in the name and path of a services lookup file. This file will be used for all lookups done by the NetLDI. While this is most importantly for **gemnetobject** and related login scripts, this also includes lookups for support processes such as remote gem shared page caches.

The **-Q** argument is read when the NetLDI starts up. If you change an entry in the specified services lookup file, you must restart the NetLDI.

startnetldi now requires port to match GEMSTONE_NRS_ALL

`GEMSTONE_NRS_ALL` is commonly used to define the listening port for a NetLDI; in this case, **startnetldi** can be started without specifying a name or port, and it will be started on the port specified in `GEMSTONE_NRS_ALL`.

Previously, if you invoked **startnetldi** including the name or port, (e.g., `startnetldi 54321` or `startnetldi gs64ldi`), and the port number specified or the port mapped from the text name does not match the port specified in `GEMSTONE_NRS_ALL`, it would start up the NetLDI on the **startnetldi** argument port, and print a warning about the mismatch. Now, this configuration is a error, and the NetLDI will not start.

This avoids certain corner cases in which there may be confusion in accessing the correct NetLDI, such as external sessions.

To start up a NetLDI with a name or port other than the port specified in `GEMSTONE_NRS_ALL`, you should unset `GEMSTONE_NRS_ALL`, or redefine it to provide the intended port number.

Solo logins temporary files renamed

Solo logins create a temporary file in `/tmp`, previously named `gemRO_pid_extent1.dbf`; this may be left behind on certain kinds of crashes. To make this more identifiable, the file is now named `gemSolo_pid_extent1.dbf`.

Upgrade for customer-defined special classes

With customer-defined special classes, added in v3.7, you rename and customize one of a number of kernel classes, `Special56Bit0..Special56Bit15`. The upgrade process for these classes does not delete all methods on these classes, as normally happens during upgrade.

The Money class, an example of customer-defined special classes, has minor refinements in v3.7.1.

Simplified finding reference path methods

The following methods have been added, which invoke search methods on `GsSingleRefPathFinder`, and suppress printing of some intermediate search details.

```
Object >> findReferencePath
```

```
Object >> findReferencePathString
```

These methods allow you to find a reference path to an object, by executing a expression such as:

```
(Object objectForOop: 97418241) findReferencePath
```

Added Repository analysis methods

The following methods have been added:

```
Repository >> countInstances: anArray numThreads: maxThreads
```

```
objMaxSize: aSize
```

Scans the entire Repository and generates an array of instance counts of the classes contained in *anArray*, counting only objects with size \leq *aSize*. Does not include in-memory objects.

```
Repository >> listInstances: anArrayOfClasses toDirectory: aString
```

```
numThreads: maxThreads objMaxSize: aSize
```

List instances of specified classes, listing only objects with size \leq *aSize*. Writes files named *className*-instances.bm files in the directory specified by *aString*.

```
Repository >> listReferencesFromBmFiles: anArray numThreads:
```

```
numThreads toDirectory: aDirectoryString
```

Reads bitmap files of instances whose paths are listed in *anArray*. Writes files, one per thread, in the directory specified by *aDirectoryString*. Files are named references-*N*.txt, where *N* is thread number, containing one line per object containing a reference, with the format

```
<objId> <1-based iv offset>:<objId of value> <ivOffset>:<valueId>
```

Excludes references to `OOP_EMPTY_INVARIANT_ARRAY`, `OOP_EMPTY_INVARIANT_STRING`, and `OOP_EMPTY_SYMBOL`, which have oops 233217, 233473, and 233729. Also excludes references from instances of `VariableContext`, `GsNMethod`, and `GsProcess`.

Returns a report of classes and number of instances of each class containing references to objects specified by *anArray*.

Backup and Restore Changes

Backup, Restore and other scans report progress

Operations such as backup, restore, markForCollection, and similar take considerable time on larger repositories. Now, these will report progress.

By default, every three minutes (180 seconds), a message is reported with the timestamp, progress count, and percentage, e.g.

```
...
[01/12/2024 14:35:27.416 PST] ProgressCount = 116961902 (22%)
[01/12/2024 14:35:30.503 PST] ProgressCount = 193092488 (36%)
[01/12/2024 14:35:33.591 PST] ProgressCount = 265600707 (50%)
[01/12/2024 14:35:36.679 PST] ProgressCount = 336629872 (63%)
...
```

Note that there may be a delay before the first message is printed, due to time required for the initial scan.

Progress messages can be disabled, or the timing of progress count messages can be configured in session state, using the methods:

```
System class >> progressPrintInterval: seconds
Sets the progress print interval for long operations such as backup, restore,
markForCollection, etc. A value of 0 disables printing. Negative values and objects
other than SmallIntegers are silently ignored.
```

```
System class >> progressPrintInterval
Answers the progress print interval for long operations such as backup, restore,
markForCollection, etc. A value of 0 means progress will not be printed.
```

Performance improvement in restoring a zipped tranlog

Restoring gzip- or lz4-compressed tranlogs has always been much slower than restoring an uncompressed tranlog.

A new thread is used in the Stone for decompressing if the tranlog being replayed is compressed, making the restore significantly faster.

restoreFromBackups:threads: now accepts filename

The method `Repository >> restoreFromBackups: fileNameOrArrayOfNames threads: numThreads` now accepts a single filename string, as well as an Array of file names.

Hot Standby-related changes and bug fixes

Tranlog position may have been inconsistent between master and slave

In a hot standby system, the reporting for the master's tranlog position and the slave's tranlog position did not use the same definition of the tranlog position, which resulted in cases where the slave was up to date, but the numbers were reported as different.

Now, cache statistics for tranlog replay, and the information reported by `Repository >> restoreStatus` and `Repository >> failOverStatus` consistently use the next physical tranlog record id as the tranlog position. Checkpoint stats continue to report the physical id of the commit record. (#50738, #50766)

failOverStatus now includes if logreceiver is connected

Another element has been added to `Repository >> failOverStatusInfo`, and is reported by `failOverStatus`; element 11 is a boolean indicating if a logreceiver is connected to the slave stone (for the slave for which information is reported by `failOverStatus`, which may be one of several slaves).

Avoid running out of stack while handling an AlmostOutOfStack error

The `AlmostOutOfStack` error is sent when the process has a small amount of stack space remaining, allowing handling of that error rather than terminating the process. However, if the handler code uses further stack space, it may run out of stack space and shut down.

Now, the amount of stack headroom set aside between the limit at which `AlmostOutOfStack` is sent, and the hard limit on stack size, is now scalable and configurable. By default, this is 25% of the stack depth; the stack depth defaults to 1000.

GEM_SMALLTALK_STACK_ERROR_PERCENT

After an `AlmostOutOfStack` or `AlmostOutOfStackError` is signaled, and if an error handler (on:do:) is handling the notification or error, execution of the error handler runs in a yellow zone area of the stack. If the stack grows beyond the yellow stone a not-trappable `AlmostOutOfStackError` is signalled to the GCI. The value of this parameter is computed as a percentage of `GEM_MAX_SMALLTALK_STACK_DEPTH`.

Default: 25 min: 10 max: 100

GEM_MAX_SMALLTALK_STACK_DEPTH maximum changed

Previously, the upper limit for `GEM_MAX_SMALLTALK_STACK_DEPTH` could have different values; now, the upper limit is 80000.

Utility Changes

`copydbf`, `updatesecuredbf`, and `startstone` includes changes to support multiple encryption keys; see “Backup and Restore Changes” on page 15

startstone, pageaudit -W to disable cache warming

Cache warming can be configured to automatically to be run on Stone startup using the configuration parameter `STN_CACHE_WARMER_ARGS`. Now, it is possible to disable this automatic cache warming using the added `-W` argument to `startstone/pageaudit`. When `-W` is specified, `STN_CACHE_WARMER_ARGS` disabled.

Added Topaz SET commands

SET STONE

The option `stone` has been added to the `set` command. `set stone` has the same function as `set gemstone`, but is a more correct name for this function.

SET AUTOSTACKSAVE

`set autostacksave` requires one argument, OFF or ON.

If on, **stack save** is executed automatically for each exception returned to topaz from an execution (**doit**, **run**, **printit**, **send**).

The default is ON if the topaz stdin is a terminal, otherwise default is OFF.

gemnetobject script cleanup

The gemnetobject scripts (`gemnetobject`, `gemnetdebug`, `gemnetobject_keeplog`, `gemnetobject_slow`, and `gemnetobject_noop`) have been updated; primarily in comments, but there are minor changes in behavior.

When these scripts were updated for the gem location requirement changes introduced in v3.7, the `gemnetobject_slow` and `gemnetobject_noop` scripts were not correctly updated, and did not actually login using slow or noop libraries. (#50834)

GCI Changes

The following thread-safe functions have been added:

GciTsAddOopsToNsc

```
(BoolType) GciTsAddOopsToNsc(  
    GciSession sess,  
    OopType theObject,  
    const OopType *theOops,  
    int numOops,  
    GciErrSType *err );
```

Returns FALSE if an error returned in *err. (theObject isKindOf: IdentityBag) must be true. Adds objects to theObject, as per IdentityBag >> addAll:

GciTsFetchNamedOops

```
(int) GciTsFetchNamedOops(  
    GciSession sess,  
    OopType theObject,  
    int64 startIndex,  
    OopType *theOops,  
    int numOops,  
    GciErrSType *err );
```

Returns -1 if an error returned in *err, otherwise returns the number of oops returned in *theOops, which will be <= numOops. startIndex is one based (Smalltalk style). startIndex must be >= 1. startIndex - 1 + numOops must be <= (theObject class instSize). numOops must be >= 0. theOops must be non-NULL if numOops > 0.

GciTsFetchVaryingOops

```
(int) GciTsFetchVaryingOops(
    GciSession sess,
    OopsType theObject,
    int64 startIndex,
    OopsType *theOops,
    int numOops,
    GciErrSType *err );
```

Returns -1 if an error returned in *err, otherwise returns the number of oops returned in *theOops, which will be <= numOops. startIndex is one based (Smalltalk style). startIndex must be >= 1. numOops must be >= 0. theOops must be non-NULL if numOops > 0. Returns elements of an Nsc, as per IdentityBag >> _at: or varying elements of an Array, as per Array >> at:.

GciTsStoreIdxOops

```
(BoolType) GciTsStoreIdxOops(
    GciSession sess,
    OopsType theObject,
    int64 startIndex,
    const OopsType *theOops,
    int numOops,
    GciErrSType *err );
```

Returns FALSE if an error was returned in *err. startIndex must be >= 1. Stores into varying instVars of an oop format object, as per Array >> at:put:.

GciTsStoreNamedOops

```
(BoolType) GciTsStoreNamedOops(
    GciSession sess,
    OopsType theObject,
    int64 startIndex,
    const OopsType *theOops,
    int numOops,
    GciErrSType *err,
    BoolType overlay = FALSE );
```

Returns FALSE if an error was returned in *err. startIndex must be >= 1. startIndex - 1 + numOops must be <= (theObject class instSize). If overlay==TRUE, theOops may contain elements with value OOP_ILLEGAL corresponding to instVars whose state will not be changed.

1.2 GemStone Smalltalk Changes

Instance-variable specific dbTransient feature

The dbTransient feature can now be applied to instance variables, as well as entire classes.

The GemStone class creation includes the option dbTransient, which allows you to create classes in which the data (instance variable contents) is not written to the extents. Now, you can also specify only specific instance variables of a class to be dbTransient; this allows you to have an instance that includes both transient data and persistent data.

When faulted in from disk, an object's dbTransient instance variable values will be nil. To preserve the in-memory state of the instVar, ensure the object is kept reachable in memory, such as from `SessionTemps current`, or from the Smalltalk stack.

The value on disk of a dbTransient instVar is always nil. Flush for commit writes nil to disk, and does not change the in-memory value.

Only the first 60 of the instance variables of a class (including inherited instance variables) may be dbTransient. If *aSymbol* specifies an instVar beyond 60 (i.e. instVarAt: 61 or subsequent instVar), an Error will be signalled.

dbTransient only applies to named instance variables. For an indexable class with named instance variables, if an instance's total size is greater than 2034 (self size + self class instSize) then this feature is disabled; the per-instVar dbTransient attribute will always behave as false on that instance.

Instance variables that are dbTransient may not participate in any index on an UnorderedCollection; errors will be signaled if any element of a path for an Index evaluates to a dbTransient instance variable.

The changes specified by an execution of `Class >> instVarDbTransient:value:` will not take effect completely until after this session commits, and will only be completely in effect in sessions which login after such commit. For instances committed prior to setting dbTransient attribute to true, that instance variable, may be non-nil on disk, but that non-nil value will not be visible to Smalltalk execution in sessions that login after the attribute change. `listReferences` and `markForCollection` will see the non-nil value; the next commit that changes some other instance variable in the instance will set that instVar on disk to nil.

The following methods have been added:

```
Class >> dbTransientInstVarNames
```

Returns an Array (possibly empty) of instance variable names that are dbTransient on a per-instVar basis.

```
Class >> dbTransientMask
```

Returns a SmallInteger

```
Class >> instVarDbTransient:aSymbol value:aBoolean
```

Change the per-instVar dbTransient attribute of instVar specified by *aSymbol* to *aBoolean*. The change will only be completely in effect for sessions that login after this change is committed.

Behavior class definition changed

Behavior has an additional instance variable, dbTransientMask.

The instance variable primaryCopy and its accessor method have been removed; these were only used by Maglev.

SessionTemps no longer stubs contents

To support preservation of instance variable-specific dbTransient values, associations and values added to SessionTemps will no longer be stubbed.

TreeDictionary and TreeSet added

The TreeDictionary and TreeSet classes have been added, along with a number of supporting Ht* classes. TreeDictionary and TreeSet provide improvements in performance and disk space over the functionally equivalent classes KeyValueDictionary and Set.

Internally, TreeDictionary is structured as a tree which is sorted based on the hash of the dictionary's keys. Its leaf nodes are page-sized hash tables, and its internal nodes are a B+tree variant that allows duplicate keys. An improved hash algorithm allows much more even hash distribution. Several of the Ht class instances that the TreeDictionary or TreeSet uses are DbTransient, avoiding creating garbage during split operations.

TreeDictionary implements all of the required AbstractDictionary protocol, and can be used in place of Dictionary or KeyValueDictionary.

TreeSet likewise implements most Set behavior; however, it does not currently support the operators #+, #-, or #*, and indexing over a Tree is not currently allowed.

TreeDictionary performance is comparable to or better than KeyValueDictionary, and performs faster for additions. Since TreeDictionary does not have to perform "grow" operations that rebuild the entire collection, it avoids the periodic multi-second delays when a threshold is reached on a particular add operation.

TreeDictionary requires moderately less storage, and it grows and shrinks incrementally in increments of a few pages. Unlike KeyValueDictionary, which does not reduce its internal table size, TreeDictionary can shrink down to its original size if all the elements are removed.

Now possible to step into accessor methods

Previously, the debugger would not step into methods that consisted of only a simple instance variable access; this was intentional, for implementation reasons. To improve the flow of debugging, this limitation has been removed.

ProfMonitor object creation profiling enhancements and bug fixes

Object creation profiling now includes profiling of some kernel classes

Object Creation profiling now includes counts for ExecBlock, VariableContext, and GsNMethod, which were previously excluded. This is useful since creation of these objects can impact performance.

ProfMonitor object creation profiling reported sample count incorrectly

When using ProfMonitor to profile object creation, the sample count was computed incorrectly. (#50759)

Avoiding performance impact of process host statistics

In v3.7, the way GemStone collects host process stats on Linux was updated to get additional and more accurate statistics; this however has a significantly greater impact on performance than the equivalent query in earlier releases.

New methods have been added that collect only the less expensive statistics, reporting the smaps-based statistics as 0.

```
System class >> hostEasyStatisticsForMyProcess
  Same as hostStatisticsForMyProcess on all platforms except Linux. On Linux, same
  as hostStatisticsForMyProcess except certain statistics which are expensive to
  collect are not computed and have zeros stored in their place.
```

```
System class >> hostEasyStatisticsForProcess: anInteger
  Same as hostStatisticsForProcess: on all platforms except Linux. On Linux, same as
  hostStatisticsForProcess: except certain statistics which are expensive to collect
  are not computed and have zeros stored in their place.
```

With these methods, the following host process statistics are collected:

```
UserTime
SysTime
MinorFaults
MajFlt
PercentCpuUsed
PageTablesMemoryKB
ThreadCount
VolCSW
IVolCSW
```

The following host process statistics are not collected, and returned as 0:

```
HeapKBytes
PrivateDirty
PrivateClean
PSSSize
Swap
SwapPss
MemMapRegions
```

FileSystem changes and bug fixes

ZnStream, ZnCharacterEncoder hierarchy changes; ZnObject removed

ZnStream now subclasses from Stream, rather than ZnObject.

ZnCharacterEncoder subclasses from Object, rather than ZnObject.

The class ZnObject is no longer used and is no longer in the image; or, in repositories upgraded from 3.7, it has been moved to ObsoleteClasses.

Manual flush no longer required for FileReference write streams

Previously, it was required to invoke flush for some kinds of ZnStreams. Now, all Zn write streams will automatically flush on close.

Directory creation errors were FsEACCES, now FilePermissionDenied

For a permission denied directory creation error, previously FsEACCES was signaled, now it will signal FilePermissionDenied and include the filename.

Added FileSystem-related methods

ZnEncodedWriteStream >> nextPutAllBytes: *aCharacterCollection*
Invokes nextPutAll:.

ZnEncodedStream class >> on: *aWrappedStream* encoder: *anEncoder*
Create an instance of the receiver with the specified encoder.

FsDirentStruct >> isSymlink
Return true if the receiver is a symlink.

FsDirentStruct >> isUnknown
Return true if the receiver is an unknown type.

FileSystem iteration over NFS-mounted directories may miss reporting some files

Not all filesystems reliably report the file type in the results from a readdir() system call. Files/directories of type unknown were not being included in the results for iterations. (#50730)

Symlinked directories and files omitted from results

When sending #directories or #files, if the FileReference target includes symbolic links, these were not included in the results. (#50875)

GsTsExternalSession forkAndDetachBlock:

Normally, processes started by GsTsExternalSession are limited to executing while the GsTsExternalSession is logged in. This limits its value for starting background processes and running scripts.

The following methods have been added to support this use case:

GsTsExternalSession >> forkAndDetachBlock:

GsTsExternalSession >> forkAndDetachBlock:withArguments:

GsTsExternalSession >> forkAndDetachString:

These methods perform a non-blocking execute of the given string or block. If the operation returns, it must be followed by GsTsExternalSession >> nbResult. Execution will continue running after nbLogout of this session, but until logout, it will respond to softBreak or hardBreak from the external session.

Note that the argument should include error handling and printing, so that there is enough information in the Gem log to allow problems to be debugged.

Added GCI flag

The flags available for `GciPerformNoDebug` (and other `continue` and `perform` calls) now include `GCI_PERFORM_DETACH`.

GsTestCase added methods

The following methods have been added to `GsTestCase`:

`should: aBlock raise: anObject description: descString details: detailsString`
`should: aBlock raise: anObject details: aString`
`should: aBlock raiseClasses: expClass numbers: expectedErrorNumbers description: descString`

IndexManager new methods

The following indexing methods have been added:

`IndexManager >> allIndexesReport`
Provide a report on all indexes in the image.

`IndexManager >> auditIndexes`
Perform an auditIndexes on all indexes in the system.

Reporting tranlog positions

The following methods have been added. They return a position as a `ScaledDecimal`, in which the integer portion is the transaction log number and the fractional digits indicate the record id within the tranlog.

`lastCheckpointPosition`
Returns a `ScaledDecimal` representation of the last checkpoint position within the tranlogs.

`lastTranlogPosition`
Returns a `ScaledDecimal` representation of the tranlog position.

Other added and moved methods

allInstances

The methods `allInstances` and `fastAllInstances` have been moved from `Class` to `Behavior`.

Array >> pairsDo:

`Array >> pairsDo: aTwoArgBlock`
Perform the `aTwoArgBlock` block, on the elements in the receiver.

Behavior >> extractSelector:

`Behavior >> extractSelector: sourceString`
Extract selector from source of a method, and return the selector `Symbol`; otherwise signal a `CompileError`.

BlockClosure >> doWhileFalse:

BlockClosure >> doWhileFalse: *conditionBlock*

Evaluate the receiver once, then again as long the value of *conditionBlock* is false.

CharacterCollection >> substringsSpace

CharacterCollection >> substringsSpace

Return an array containing the substrings in the receiver that are separated by Character space. Similar to `asArrayOfSubstrings`, except only spaces are considered dividers; lf, tab, etc. are left embedded.

Duration >> asMilliseconds

Duration >> asMilliseconds

Return the number of milliseconds in the length of time represented by the receiver.

Integer >> >> shiftAmount

Integer >> >> shiftAmount

Perform a bit right shift. `shiftAmount` must be greater than zero.

SmallInteger >> permutedHashA:c:d:

SmallInteger >> permutedHashA: *a* *c*: *c* *d*: *d*

With the receiver, a non-negative SmallInteger *x*, and three constants *a*, *c*, and *d*, compute

$$f(x) = dx^2 + ax + c \pmod{2^{60}}$$

This function will compute a full permutation on the interval $[0 .. 2^{60})$, as long as *a*, *c*, and *d* are all less than the modulus, *c* is odd, and *d* is $a-1 \pmod{4}$ (i.e. the low two bits of *d* = low two bits of *a*-1). A full permutation means that every possible input in the interval will produce a different output in the interval.

For good spreading of small integers, choose *a* and *c* to be large primes, and *d* should also be large, but less than the modulus.

This function can also be used to compute permutations with modulo powers of two smaller than 2^{60} . Choose *a*, *c*, and *d* appropriately, and use `bitAnd` to discard the high-order bits of the result.

String >> evaluateInContext:

String >> evaluateInContext: *anObject*

MultiByteString >> evaluateInContext: *anObject*

Perform `evaluateInContext:symbolList:`, using the current session's symbol list.

System class >> waitForDebugWithAbort

System class >> waitForDebugWithAbort

Perform a `waitForDebug` and abort.

Removed methods

The following methods have been removed:

GsFileIn >> classMethod:

GsFileIn >> method:

Removed by redesign of method parsing to avoid some corner conditions

GsTestResult class >> excludeInSoloSession:

No longer required.

PositionableStreamPortable >> originalContents

this was a cover method for PositionableStreamPortable>>collection

Removed methods related to obsolete features:

Behavior >> primaryCopy

instance variable removed; see page 20.

Class class >> classVariablesAssociationClass:

GsNMethod class >> dynamicLiteralVariables:

Removed methods that are now inherited:

CanonStringDict >> become:

KeyValueDictionary >> become:

Private removed methods:

Behavior >> _checkCompileResult:source:

BinaryFloat class >> _finishFromStream:signFound:factor:
integerPart:

BinaryFloat class >> _finishFromStream:signFound:factor:
integerPart:fractionalPart:exponent:

Number class >> _finishFromStream:signFound:factor:integerPart:
fractionalPart:exponent:exponentPart:

1.3 Changes in Encrypted extents, tranlogs, and backups

Support for up to 24 encryption keys

In previous releases, encrypted backups supported encryption with up to 8 keys, while extents and tranlogs only supported encryption with a single key.

In v3.7.1, extents, tranlogs, and backups all support up to 24 encryption keys. These can be certs, public keys, or a mix of certs and keys. In practice, the limit for extents and tranlogs that are encrypted using certs (rather than public keys) is 21 keys, due to space constraints within the file; certs use more space than public keys.

The private key corresponding to any encryption key that is in the set of keys on an extent, tranlog, or backup can be used to decrypt or read the dbf.

There are a number of changes associated with this:

- ▶ **copydbf** has been updated to allow the initial encryption to include multiple keys; see “copydbf argument changes” on page 27.
- ▶ **upgradesecuredbf** can add and remove multiple keys; the arguments have changed; see “updatesecuredbf argument changes” on page 27.
- ▶ The Stone can read extents as long as any one of the extent’s encryption keys matches the Stone’s startup private key.

startstone has been updated to allow you to specify additional keys, which are only used for reading encrypted tranlogs. The Stone can read transaction logs as long as any of the tranlog’s encryption keys matches the Stone’s startup private key or any of the additional private keys; see “startstone argument changes” on page 28.

- ▶ Tranlogs are written by the Stone are encrypted using all of the encryption keys for the first extent. Any one of these keys can be used by the Stone to read the tranlogs.

System class >> repositoryPublicKey renamed

The method `System class >> repositoryPublicKeys` has been added, which returns an Array of keys used to encrypt the repository.

The existing method `System class >> repositoryPublicKey` is now deprecated; use `repositoryPublicKeys` instead.

Path requirements and -K

Previously, the filename for the public cert and private key must have been specified without the path, the private key’s passphrase required a full path, and the **-K** argument was always required. In v3.7.1, this has been made much more flexible. With some exceptions, you may specify the full path for cert, key, and passphrase files, and omit the **-K** argument, or specify only the filenames and include **-K** to specify the directories containing these files.

The exceptions are that **-K** is required for encrypted backup files:

- ▶ **updatesecuredbf**, for a secure backup file, always requires the **-K** argument.
- ▶ **copydbf -V** (verify digital signature), on a secure backup file, always requires the **-K** argument to the directory containing the signing key/passphrase.

Utility Changes related to encrypted DBFs

copydbf argument changes

- ▶ **-e** may be specified up to 24 times to encrypt an extent or tranlog, to result in a dbf with up to 24 public certs or keys. When using certs, copydbf is expected to fail with more than 21 arguments.
- ▶ **-C** has been added, allowing you to confirm encryption with a public key (similar to **-c**, which performs verification with a private key). With **-C**, use the **-e** argument to specify the public key you wish to verify against.

Note that verification with **-C** does not require the private key, and if you do not have the private key you would not be able to use the file, e.g. to start a stone or restore a backup.

- ▶ Arguments to **-D**, **-e**, **-J** may now include the path or not. If the path is not included with any of the filenames, the **-K** argument specifying the directory containing that file is required.
- ▶ To determine if an extent is encrypted with a specific private key, using the **-c** flag, it requires that the Stone be shutdown, or use the **-O** argument. Since this operation is read-only, the warnings printed with **-O** are not applicable; these are no longer printed when **-c** is used.

copydbf may have prompted for passphrase

The default behavior of **openssl** is to prompt for the passphrase for a private key; this behavior is suppressed by **copydbf**. Under some conditions in which the private key and passphrase were not a required argument, **copydbf** allowed the **openssl** prompt to occur, although it did not read keyboard input for that prompt. (#50737)

copydbf -X, -Y output filenames did not include extension

copydbf -X and **-Y** extract information from a secure backup and write it to a specified filename. When a filename was supplied without an extension, no extension was automatically added. Now, the correct extension is added; respectively, **.pem** and **.bin**. (#50807)

updatesecuredbf argument changes

- ▶ **-e** has been removed; it is replaced by **-A**.
- ▶ **-A** has been added: Add an additional encryption key. Reencrypts the encrypted extent, tranlog, or backup with the additional public key or cert. This argument may be specified up to 24 times. The key limitation of 21 to 24 keys applies, but if the maximum number of keys will be added or are already present, you can use **-A** to add and **-R** to remove existing key/s in a single operation. This operation requires a private key corresponding to one of the previously existing encryption keys.
- ▶ **-R** has been added: Remove an existing encryption key. Removing the last key is disallowed; when removing the only key you must also use **-A** to add a new key. This

argument may be specified up to 24 times. It is not required to have access to the private key corresponding to the **-R** argument.

CAUTION: Note that to use an encrypted dbf you must have the private key as well as the public key. It is possible to remove all the public encryption key/s for which you have the corresponding private keys, and leave only public encryption key/s for which you do not have the corresponding private key. In this case **the dbf is unusable** unless you can acquire the corresponding private key.

updatesecuredbf operates on the dbf file in place. **It is strongly recommended to make a copy of the dbf file before making updates using updatesecuredbf.**

- ▶ Arguments to **-A**, **-D**, **-J**, **-R**, **-S**, **-T** may now include the path or not. If any of these arguments do not include the path, the **-K** argument specifying the directory containing that file is required. **-K** is always required with operations on secure backups.

updatesecuredbf output modified

The output from **updatesecuredbf** was unclear in that it included "source" and "destination", which were identical since **updatesecuredbf** modifies in place. Since the details were for the source (before the update), the output was misleading with respect to the number of keys on the dbf.

The output now does not include the source dbf details, and the message on success has been changed to describe the specific changes made. For example, the following is an example of the complete output after changing the single encryption key on an extent:

```
[Info]: Updating secure extent: pathToExtent
      1 Public key successfully removed.
      1 Public key successfully added.
```

startstone argument changes

The Stone's extents may be encrypted with multiple keys. The tranlogs written by the stone are encrypted with all of the keys that were used to encrypt the extents. The Stone's startup key (supplied using **-D**) allows it to read all extents and tranlogs whose encryption keys include the startup key. If an extent's encryption keys are changed (which must be done when the Stone is not running), a new tranlog is written that is encrypted with the new set of keys. In order for the stone to restart and read the extents, it must be started with an encryption key that is among the new set of keys on the extents.

When the Stone starts up in restore mode, it needs to read an earlier tranlog, which will fail if the earlier tranlog does not include the Stone's new startup key. Programmatically in some cases, this key can be added using `Repository>> setTranlogPrivateKeysForRestore:`

To simplify this, **startstone** now allows you to supply multiple keys. The startup key is still supplied with the **-D** argument (plus **-J** or **-j** as needed), and is needed to access the encrypted extents.

Additional keys are supplied using the new **-X** (plus **-Y** or **-y** as needed), which supplies a private key and the passphrase as needed. The extents may or may not be encrypted with these keys; these keys are only used to read encrypted tranlogs.

When using **-X** to supply additional keys, if any of the additional private keys requires a passphrase, the **-J** or **-j** argument is required for all **-X** keys (the argument to **-J/-j** can be omitted), and the **-J/-j** arguments must in the same order as the **-X** arguments.

The following new arguments have been added to startstone:

- X** Specifies an additional private key file used to read tranlogs which require a private key other than the key specified with the **startstone -D** option. May be specified up to 8 times, and may be a full path or a filename within one of the **-K** directories. If any private key in the list requires a passphrase, exactly one **-y** or **-Y** option for each **-X** option specified must also be provided.
- y** Specifies a passphrase for the private key specified by an **-X** argument. May be specified up to 8 times. **-y** and **-Y** options may be mixed but must be in the same order as the **-X** options. For keys with no passphrase, omit the argument after the **-y** option. If one or more **-y** or **-Y** options are specified, the combined total number of **-y** and **-Y** options must match the number of **-X** options.
- Y** Specifies a file containing the passphrase for the private key specified by an **-X** argument. May be a full path or a filename within one of the **-K** directories and may be specified up to 8 times. **-y** and **-Y** options may be mixed but must be in the same order as the **-X** options. For keys with no passphrase, omit the argument after the **-y** option. If one or more **-y** or **-Y** options are specified, the combined total number of **-y** and **-Y** options must match the number of **-X** options.

The following bugs were present in v3.7 and are fixed in this version.

Sending #value to special resulted in SEGV

Sending #value to a special (such as a SmallInteger) resulted in a SEGV in the Gem process. (#50748)

startstone recovery may fail with tranlogs on raw partitions

After an unclean Stone shutdown, such as a **kill -9**, on restart the Stone performs recovery by reading the current tranlog that was being written before the shutdown. When tranlogs are configured as a list of raw partitions, there was a codepath in which the stone may fail to find the tranlogs needed for recovery; in this case, the stone will not start unless **startstone -N** is used. (#50843)

Stone may shutdown on checkpoint when tranlog disks are full

When the tranlog disks are full, and the stone has paused checkpoints waiting for tranlog disk space to become available, if the checkpoint timeout interval expires and a checkpoint initiated, the stone shut down. (#50744).

A similar issue, bug #49704, was fixed in v3.7; this is a slightly different code path. The Stone may only shutdown due to lack of tranlog disk space during **startstone** or when restoring from log/backup.

Garbage collection issues

Performance regression in markForCollection

In v3.6, code changes were introduced to avoid thread-safety issues in garbage collection leaf caches. This change resulted in significantly slower performance (for some configurations) in v3.6.x, vs. 3.5.x and earlier. The fix has been rewritten and performance is now significantly faster than pre-3.6 releases as well as faster than 3.6.x versions. (#50770).

markForCollectionWait: with argument of -1 gets prim failure

Executing `Repository >> markForCollectionWait:` with an argument of `-1` (meaning wait forever), resulted in a primitive failure. (#50769)

reclaimAll may fail to reclaim some dead

It was possible for the `reclaimGem` to retain an old copy of the `deadObjs` and not actually reclaim all dead. (#50798)

There is additional information in the progress printed information, and `reclaimAll` signals a Repository error if it is stuck; that is, if there is no progress after 10 minutes.

C mutex deadlock possible from ProfMonitor/SIGUSR1

There was a small timing window in which was it possible for a Gem to deadlock on a C mutex, when processing a VM interrupt. This occurred periodically with ProfMonitor running, and when the Gem is sent SIGUSR1. (#50729)

NetLDI issues**NetLDI shutdown timing issue resulted in unable to restart**

When a NetLDI services a request to fork a gem, it forks a child process and then does an exec. During the period after the fork and before the exec, a thread in the child process makes system calls to collect information for printing the log file banner. If the NetLDI is stopped using either `kill -TERM` or `stopnetldi` while a thread is making a system call, this could have resulted in a deadlock in the child.

The fix removes the system call in which the deadlock has been observed. The forked child also now immediately closes its inherited file descriptor for the NetLDI's listening socket, rather than waiting for the close-on-exec of that socket to take effect.

Since informational system calls are no longer made during the forked period, the Gem log headers now show a reduced section for "NetLDI Child Task", which includes only the Gem command line. The system information is already duplicated in the header information printed by the Gem. (#50697)

netldid executes without \$GEMSTONE but does not function

The `netldid` executable started up without the `$GEMSTONE` environment variable correctly set, however, it did not function correctly. Now, this case will error on startup attempt. (#50782)

GsSecureSocket instance method did not properly used chained certificate file

The method:

```
GsSecureSocket >> useCertificateFile:withPrivateKeyFile:private
  KeyPassphrase:
```

did not handle chained certificates; only the leaf certificate of a chained cert file would be used. (#50900)

The related class methods did work correctly with chained certs.

GsSocket >> writeWillNotBlockWithin: does not handle nil returned by writeWillNotBlock

It was possible for `GsSocket >> writeWillNotBlock` to return a nil, which resulted in an error in `GsSocket >> writeWillNotBlockWithin:.` (#50775)

Unicode string copyReplaceAll:with:, insertAll:at:, broken with String arguments

When the receiver is a unicode string and the argument is a legacy string, the methods `copyReplaceAll:with:`, `insertAll:at:`, and any methods that invoke these methods, would see a Message Not Understood error. This is related to changes in string conversion introduced in v3.7. (#50805)

Collection issues

Array >> replaceFrom:to:with:startingAt: may not have copied elements correctly

Using `replaceFrom:to:with:startingAt:` where the source for the replace is the same as the receiver and the starting position in the source for the replace is less than the target starting index, the replace must copy in reverse. This case did not correctly perform all the copy operations for large collections, and produced incorrect results. (#50721)

Dictionary become: does not update internal variables

After sending `become:` to a Dictionary containing `EqualityCollisionBuckets`, the back references from the `EqualityCollectionBucket` to the Dictionary incorrectly was the `become:` argument dictionary, not the receiver. (#50717)

Numeric and Time related issues

Strings with leading '-0.' lose sign when converted using asNumber or Number fromString/Stream:

Converting a string containing a decimal number that is less than zero and greater than -1, such as '-0.1', using unspecific conversion methods `asNumber`, `Number fromString:`, or `fromStream:`, resulted in a positive value, the absolute value of the correct result. (#50751)

Duration new created an uninitialized instance

`Duration new now` returns `Duration zero` rather than an uninitialized instance. (#50710)

Issues related to upgrade

upgradeImage reports success even if upgrade fails

The upgradeImage script returned true in 3.7, even when the upgrade actually did not succeed. (#50750)

Excessive attempts to start AdminGem during upgrade

During upgrade, after the Stone has been started but before **upgradeImage** has been run, the Stone continuously attempted to start the Admin GcGem, which failed due to the version mismatch until **upgradeImage** completes. (#50773)

copydbf returns wrong version for secondary extents after upgrade

After upgrade, the dbf version in the extent is updated with the new version; this is the version reported by **copydbf -i**. This update was incorrectly only applied to the primary extent. For repositories with multiple extents, **copydbf -i** on any of the secondary extents reported the old version. (#50749)

Session methods must be disabled for upgrade

Session methods are a feature used by GsDevKit and GLASS, but should not be enabled in the ordinary base GemStone without knowledgeable assistance. If GsPackagePolicy current is inadvertently enabled, it causes a number of problems. This is not a supported configuration for upgrade; now, upgrade will error if GsPackagePolicy current enabled returns true. (#50453)

Upgrade created instances of ObsoleteMetaclass

The upgrade process created instances of ObsoleteMetaclass. This did not affect operation, since they were automigrated when faulted into memory. (#50810)

GsNMethod >> recompileFromSource loses method category

The method GsNMethod>>recompileFromSource changed the method category to '(as yet unclassified)'. (#50731)

perform:withArguments: may crash on bad arguments

When the arguments to Object >> perform:withArguments: is bad (for example, nil), native code did not handle this, and the session was terminated. Linked topaz reported Segmentation fault (core dumped), but an RPC gem log did not report a message. (#50864)

Spurious OutOfMemory when in-memory pom_gen scavenge fails

When pom_gen scavenge failed, in-memory GC may have signaled an OutOfMemory error when not actually out of memory. (#50711)

Fileout format reverted to older style in v3.7

The class fileout format lines for removing existing methods was updated in v3.6 to use **topaz removeAllMethods** commands. In v3.7, this was inadvertently reverted to the older style using **doits**. This has been corrected, and fileouts from v3.7.1 images include **topaz removeAllMethods** commands. (#50732)

GsExternalSession >> executeString: fails if attempting to return an empty String

When the result of executing `GsExternalSession executeString: cmdString` is an empty String, it resulted in an `ArgumentError` with "source of copy goes past end of source memory". (#50882)

defaultGemNRSFromCurrent may be incorrect

The method `GsNetworkResourceString >> defaultGemNRSFromCurrent` composes a gem login parameter, such as `!@hostname#netldi:33333!gemnetobject`. This used a netldi port specified by `GEMSTONE_NRS_ALL` in the session's environment, which could have been different than the NetLDI actually used to spawn the session.

By restricting a NetLDI to have a matching port and `GEMSTONE_NRS_ALL` (see "startnetldi now requires port to match GEMSTONE_NRS_ALL" on page 13), this is now handled correctly. (#50046)

GLASS/Seaside auto-migrate may error after GC operation

In the GLASS/Seaside environment, Class versioning (including that done by **upgradeSeasideImage**, as well as code development), automatically migrated instances to the new Class version, which invokes a repository scan operation to detect the instances. This scan operation may conflict with voting or promote operations that occurs as part of an Epoch or after a `markForCollection`. (#50078)

ensure blocks in superDoit scripts do not execute on error

Ensure blocks in the `doit Block` of a `superDoit` script were not executed if the code encountered an error. (#50874)

Issues with GsProcess terminate

Overlapping terminate request semantics

If a `GsProcess` that is in the process of terminating another `GsProcess` was itself terminated, it could have resulted in an error such as "termination already started", or potentially no error.

Now, the in-process terminations of the other `GsProcesses` are tracked. If a `GsProcess` receives a request to terminate, it will first wait for its own earlier termination requests to complete or timeout. (#50754, #50756)

SEGV when a terminated GsProcess is sent terminate

From GBS, it was possible to interrupt a GsProcess that was in the process of being terminated, and send another #terminate message to it. This SEGVed if the timing was such that the GsProcess had already been terminated. (#50702)

Gem failed to respond to SIGTERM on 1-core VM

On a 1-core virtual machine, a gem may have failed to honor a SIGTERM. (#50844)

CodeModification required to list Pragmas with session methods enabled

When session methods are enabled (this is a feature that is used in Seaside and GsDevKit), finding or enumerating Pragmas required the user to have #CodeModification privilege; which should not be needed for a view-only operation. (#50620)

waitstone may return before Stone is ready to accept logins

Under some conditions, observed with **startstone -R**, **waitstone** can return with the message "stone is ready", while the Stone is still in startup and not ready to accept logins. (#50642)

Array with:* primitive failure error reports incorrect

The Array >> with:with:with:[with:]+ methods invoke a primitive. If this primitive fails, the error message did not correctly report the number of with: arguments in the selector and/or the argument list. (#50786)