# GemStone/S 64 Bit™ Release Notes

## Version 3.5

June 2019

GEMTALK™ SYSTEMS

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

## TRADEMARKS

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

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

**Solaris**, **Java**, and **Oracle** are trademarks or registered trademarks of Oracle and/or its affiliates. **SPARC** is a registered trademark of SPARC International, Inc.

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

**Microsoft**, **Windows**, and **Windows Server** are registered trademarks of Microsoft Corporation in the United States and other countries.

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

**Red Hat** and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.

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

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

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

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

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

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

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

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

# Preface

## About This Documentation

These release notes describe changes in the GemStone/S 64 Bit™ version 3.5 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 3.5.

## 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 3.5*

## Chapter 2. Changes in Administration

## *Chapter 3. Changes in GemStone Smalltalk*

## *Chapter 4. Changes in the GCI Interfaces*

## Chapter 5. Bug Fixes

Chapter
1
# Release Notes for 3.5

## Overview

GemStone/S 64 Bit™ 3.5 is a new version of the GemStone/S 64 Bit object server. Version 3.5 includes support for X509-Secure GemStone, new infrastructure that supports remote OpenSSL-based TLS X.509 secure RPC logins, with many security features. These features operate independently of the classic GemStone architecture, which is not affected by X509-Secured GemStone. This release also includes many other new features, enhancements and bug fixes.

These Release Notes include changes between the previous version of GemStone/S 64 Bit, v3.4.3, and v3.5.

Upgrade is supported from GemStone/S 64 Bit v3.4.x, 3.3.x, and 3.2.x.

For details about installing GemStone/S 64 Bit 3.5 or upgrading from earlier versions of GemStone/S 64 Bit, see the *GemStone/S 64 Bit Installation Guide* for v3.5 for your platform.

## New keyfiles required

The keyfiles for v3.4.x and earlier cannot be used with v3.5. If you are upgrading from v3.4.3 or earlier, new keyfiles are required. To obtain a new keyfile for GemStone/S v3.5, write to keyfiles@gemtalksystems.com. In your request, include your license information, platform and any updates to contact information.

Please contact GemTalk Technical Support if you have issues or questions.

The X509-Secured GemStone feature requires specific keyfile permissions.

### Keyfile expiration dates

Keyfile expiration dates previously were printed with the potentially ambiguous date format MM/DD/YYYY. Now, the display is YYYY-MM-DD.

# Supported Platforms

## Platforms for Version 3.5

GemStone/S 64 Bit version 3.5 is supported on the following platforms:

- Red Hat Enterprise Linux Server 6.9 and 7.4, and
  Ubuntu 16.04 and 18.04, and SUSE Linux Enterprise 12, on x86
- Solaris 10 and 11.3 on x86
- AIX 6.1, 7.1, and 7.2
- OS X 10.13.2 (High Sierra), with Darwin 17.3.0 kernel, and
  OS X 10.14.3 (Mojave) with Darwin 18.2.0 kernel, on x86
  (Mac is supported for development only)

For more information and detailed requirements for each supported platforms, please refer to the *GemStone/S 64 Bit v3.5 Installation Guide* for that platform.

The X509-Secured GemStone feature is fully tested and supported on Linux platforms only.

## GemBuilder for Smalltalk (GBS) Versions

GemStone/S 64 Bit version 3.5 requires GBS version 8.4 or later for VisualWorks Smalltalk, or version 5.4.5 or later for VA Smalltalk. GBS versions 8.3/5.4.4 can login, but some signal-related operations may fail.

The following versions of GBS are supported with GemStone/S 64 Bit version 3.5:

### GBS version 8.4

| **VisualWorks<br>8.3.2<br>32-bit and 64-bit** | **VisualWorks<br>7.10.1<br>32-bit** | **VisualWorks<br>7.10.1<br>64-bit** |
|---|---|---|
| ‣ Windows 10 and Windows 7<br>‣ RedHat ES 6.9 and 7.4; Ubuntu 16.04 and 18.04 | ‣ Windows 10 and Windows 7<br>‣ RedHat ES 6.9 and 7.4; Ubuntu 16.04 | ‣ Windows 10<br>‣ RedHat ES 6.9 and 7.4 |

### GBS version 5.4.5

| **VA Smalltalk<br>9.1** | **VA Smalltalk<br>8.6.3** |
|---|---|
| ‣ Windows 10<br>‣ Windows 7 | ‣ Windows 10<br>‣ Windows 8.1<br>‣ Windows 7 |

GBS/VA does not support logins using X509-Secured GemStone.

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

## VSD Version

The GemStone/S 64 Bit v3.5 distribution includes VSD version 5.4. This is the same version of VSD that was included in the previous version of GemStone/S, v3.4.3.

The build number has changed: this reflects new definitions added to describe new statistics collected in v3.5.

## Upgrade

Upgrade is supported from all 3.3.x and 3.4.x versions without recompile of methods and blocks. Upgrade from 3.2.x does require method and block recompile. To upgrade from earlier versions, upgrade to a 3.3.x or 3.4.x version and then upgrade to 3.5.

### Upgrading from GLASS/GsDevKit/Seaside

The **upgradeSeasideImage** script that can be used during upgrade has been rewritten and expanded in this release. Applications using GLASS, GLASS1, the original GsDevKit environment, and tODE can upgrade using the processes described in the *Installation Guide*.

For more information on these different starting environments and the upgrade process, see the discussion here:

https://github.com/GsDevKit/GsDevKit_upgrade/blob/master/README.md#upgrading-glassgsdevkit-applications-to-gemstone-350

## Documentation Changes

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

In addition to the maintenance changes and addition of new information and features, the following improvements have been made:

- A new manual has been added, the *GemStone/S 64 Bit X509-Secured GemStone System Administration Guide*. This contains all configuration, setup, and architectural information related to X509-Secured GemStone.
- gemnetobject now is documented as a utility in the *System Administration Guide*, and using arguments to gemnetobject and to a Gem are described in detail.
- *System Administration Guide* Appendix F on tranlog analysis has been moved out of that manual and is now a separate document, http://downloads.gemtalksystems.com/docs/Other/SDoc-TranlogAnalysis-3.5/SDoc-TranlogAnalysis-3.5.htm.
- A new chapter has been added to the *Programmer's Guide*, providing some encryption and validation information using OpenSSL.
- The *Topaz Users Guide* has reorganized and expanded material on scripting, including the new topaz features.
- Indexes are no longer included in the .pdf documents; since these documents are no longer printed; search tools provide a more reliable and maintainable way of locating information.

## 1.1 Library and Distribution changes

### Updated library versions

The version of lz4 has been updated to v1.9.1.

The version of OpenSSL has been updated to 1.1.1b.

The version of OpenLDAP has been updated to 2.4.47.

The version of MIT Kerberos has been updated to 1.17.

### Updated ZoneInfo (TimeZone database)

The zoneinfo TimeZone database has been updated from version 2017b to 2019a. As part of this update, the files within the timezone directory tree have been reorganized to adhere more closely to standard directory naming conventions.

### Raspberry Pi

An initial version of support for GemStone clients running on Raspberry Pi is now supported. The Raspberry Pi runs Linux distribution "Raspian", which is a Debian Linux variant.

Raspberry Pi is a client-only distribution with the name `GemStone64BitClient3.5.0-arm.Linux`. No specific *Installation Guide* has been created; the process is similar to a client-only installation on Linux.

### 32-bit openSSL client now included

A 32-bit openSSL client application is now included, `$GEMSTONE/bin32/openssl`.

### Change in how shared libraries are packed

In v3.4, the three shared libraries previously included in the distribution were replaced by a single library, libfloss.

However, this did not allow sufficient flexibility in specifying the openSSL library, so this change has been backed out. In addition, extra libraries are now used by the NetLDI.

In v3.5, `$GEMSTONE/lib` now includes:

```
libkrb5-NN-64.so
libldap-NN-64.so
libldipcmon-3.5.0-64.so
libldipgsvr-3.5.0-64.so
libnetldi-3.5.0-64.so
libssl-NN-64.so
```

`$GEMSTONE/lib32` now includes:

```
libkrb5-NN-32.so
libssl-NN-32.so
```

(as well as other files); and the `libfloss*` libraries are not shipped.

## File documenting special selectors

A file, `$GEMSTONE/doc/specialSends.txt`, has been added. This lists reserved selectors and other special cases in which there is no message send for code execution.

## shrpcmonitor shell removed

The shrpcmonitor shell was previously usable for certain commands, such as computing large memory page requirements. This shell has been removed, and `startshrpcmon` is no longer included in the distribution. Large memory page requirements are now calculated using the added **largememorypages** script, described on page 34.

## Open Source Library versions

The distribution now includes a file containing the versions of open source software included in the product, under `$GEMSTONE/licences/ossversions.txt`.

In v3.5, the contents of this file are:

```
Package         Version(s)
=========================================
icu             58.2, 54.1, 51.2
MIT Kerberos    1.17
libicuid        1.4.1
linenoise       none
lz4             1.8.3(server), 1.3.1 (VSD)
oniguruma       5.9.1
openldap        2.4.47
openssl         1.1.1b
rbc             0.1
tcl             8.6.8
tk              8.6.8
tix             8.4.3
zlib            1.2.11
zoneinfo        2019a
```

## Temporary files on Linux now created in /tmp, not /usr/tmp

Recent linux distributions do not define `/usr/tmp/`, as the standard location for temporary files is now `/tmp/`. GemStone now creates temporary files in `/tmp/`.

## Banner printing changes

The log files and topaz banners have changes on Linux, Mac, and Windows; memory information is now on a separate line, and processor information is included. Note that both the number of physical cores and the number of CPUs are provided, the CPU count is the number of hyperthreaded cores, usually 2X physical cores.

## DbfHistory now includes build as well as version

The Global DbfHistory tracks the upgrade history of the GemStone image. The version that is listed now includes the build number as well. E.g.

upgrade to GemStone 3.5.0 64bit-45899 at 20/02/2019 16:17:25

# 1.2 Optimizations

## Reduced round-trips from Gem to Stone

During login, most Gems made seven round trips to the Stone, which is expensive over slower networks. This has been reduced to four round-trips for most logins.

A remote Gem was making a separate call to the Stone to fetch the lock sets, if any object was locked. Now, the lock sets are returned by the Stone, as part of the serialization request, saving a round trip during a critical region.

## SSL-encrypted Network Connections

Now, all network communications may be encrypted using anonymous SSL.

**STN_ANONYMOUS_SSL**
If true, the network connections between the stone and other processes and between shrpcmonitor and other processes are encrypted with Anonymous SSL. In this case, the settings for GEM_PGSVR_USE_SSL and GEM_RPC_USE_SSL are ignored and the GCI to gem and gem to pgsvr connections will always use SSL. If false, which is the default, the behavior is as in previous releases.

This parameter does not affect any connections established using X509-secured GemStone; those connections always use SSL.

## Loading a specific OpenSSL library version

An external OpenSSL library (that is, an OpenSSL library that is not part of a GemStone distribution) can now be loaded instead of the OpenSSL library provided by GemStone. To do this, set the environment variable GS_OPENSSL_LIB to reference the external library.

The external OpenSSL directory (not the file) must also be in the LD_LIBRARY_PATH, to ensure that the dynamic loader can find the `libcrypto.so` library.

To ensure that failing to configure LD_LIBRARY_PATH is detected, GemStone prints out the value of LD_LIBRARY_PATH when attempting to load a non-GemTalk SSL library.

Only OpenSSL 1.1 and later are supported.

## Improved support for additional execution environments

In addition to the default Smalltalk execution environment with envId 0, GemStone allows additional execution environments. The handling of code in envIds greater than 0 is improved in v3.5:

- topaz commands are all environment aware
- A number of issues and bugs, such as in Behavior's handing of class and method operations, and ClassOrganizer functions, have been fixed.
- Behavior fileout methods have been added, with an additional environment variable keyword.

## Expansion in file and path names

Smalltalk methods, topaz commands, and other operations use of strings to describe file paths and names, which may include references to shell environment variables. Previously, these were limited to a single leading environment variable, and other instances of embedded dollar signs $ were interpreted literally.

Now, any dollar sign found in a string defining a file name/path is interpreted as a environment variable name to be expanded. Valid characters in environment variable names are a-z, A-Z, 0-9 and underscore. Any other character terminates the environment variable.

If no matching environment variable is defined, an empty string is used.

This includes a change in behavior for ambiguous usages. If `$FOO` is defined as `foo`, using `$FOObar` will now return an empty string, and may produce an error on subsequent use; attempting, for example, to write to a directory. In previous versions, it would have produce a file path or name with a literal `$`, and a subsequent use for writing would have written, for example, to a file with the name `$FOObar`.

It is also now supported to include curly braces `${`*varname*`}` to define environment variables that are substrings. For example, with `$FOO` is defined as `foo`, `${FOO}bar` would be expanded to `foobar`. `{$FOO}bar` expands to `{foo}bar`.

## Automatic Asynchronous Aborts

GemStone/S 64 Bit v3.5 and GBS v8.4 now include support for an automatic asynchronous abort feature. This allows a Gem that is outside of a transaction to automatically process sigAbort from the Stone, using handler blocks to provide detailed instructions for processing. This feature has specific integration requirements, and should be used under the guidance of GemTalk Engineering.

The server function is enabled with the configuration parameter **GemAutoServiceSigAbort**, which is described on page 41.

The GBS API is documented in the GBS method `GbsSession>>asyncAbortStartAction:finishAction:`.

## Spin locks reduced

The Free frame spinlock and the Free PCE spinlock have been eliminated, and instead managed using bitlists, which are atomically updated. This improves performance.

## Array with: now implemented as primitive

A primitive has been added for `Array class >> with:`.

## Cache line size

The cache line size was changed from 128 to 64 bytes on Intel/AMD CPU platforms (Solaris x86, Linux x86, Darwin)

## OOB signal

The out-of-band signal from the Stone to Gems is a 6 byte packet in v3.5.

# 1.3 Support for X509-Secured GemStone

X509-Secured GemStone is a new architecture that supports mutually-authenticated secure SSL connections between all processes within a distributed GemStone environment. These are designed to allow Gems to run on a remote node that is less secure than the Stone’s node.

The Stone’s node is expected to remain in a secure environment; there are no differences in the way the Stone is configured, started up, or secured.

You must have a keyfile with appropriate permissions in order to use X509-secured logins.

X509-Secured GemStone Configuration, Login, Administration, and other details are described in the new manual, the *GemStone/S 64 Bit X509-Secured GemStone System Administration Guide*. Changes that only support X509-Secure GemStone, such as additional utility command arguments and utilities that generate certificates, are **not** included in these Release Notes.

## New HostAgent process

The HostAgent process has been added to support X509-Secured logins. The HostAgent provides necessary services to remote X509-secured processes.

The HostAgent runs as a new system user, HostAgentUser, with its own GsObjectSecurityPolicy. The HostAgent is started with the new script **starthostagent** that is executed on the Stone’s node.

Each remote node running X509-secured Gems has a HostAgent process running on the Stone’s node

## New login parameter set

X509-secured logins are performed using new C calls that require a different set of login parameters than ordinary logins. Arguments have been added to topaz and GBS 8.4 to support X509-secured logins.

## All connections require certificates

All interprocess connections in X509-secured GemStone require X.509 certificates.

GemStone has added utilities to support generation of Host, User, certification authority, and Certification-Revocation-List PEM files. Certain utilities, such as **startnetldi** and **gslist**, have added arguments to allow specification of certificate files.

## Remote and mid-level shared page caches

To mediate remote X509-secured logins, NetLDIs must be started in X509-secured mode on both the remote Gem host and the Stone’s node, configured with the appropriate certificates, before an X509-secured remote cache can be started.

The remote shared page cache is started by the remote NetLDI, when requested to do so by the HostAgent on the Stone’s node.

X509-Secured GemStone also supports X509-secured Mid Level Caches.

## Object Filtering

Object Filters are applied to all objects on any pages that are requested by a remote X509 cache, determined by GsObjectSecurityPolicy and IP address. Objects that are protected are not transmitted to the remote cache. This avoids the risk that sensitive data may be present at all in a remote cache.

The classes ObjectFilteringPolicy and ObjectFilteringPolicyMap have been added to support this function. The classes IPv4Subnet, Cidr, CidrGrammar and CidrParser have been added to represent IP addresses and allow subnet filtering.

Chapter

2

# Changes in Administration

This chapter describes changes and new features that apply to the administration of a GemStone Repository, and changes in the GemStone environment, configuration, and tools, including:

# 2.1 Changes in Users and ObjectSecurityPolicies

## Added system users

Two new users have been added, HostAgentUser and CodeLibrarianUser, each with new GsObjectSecurityPolicies. These are system users that, like other system userProfiles, are immune from password expiration.

### HostAgentUser

The HostAgent processes run as the HostAgentUser. HostAgentUser's UserGlobals contains a number of classes that implement the HostAgent's responsibilities. These are intended for internal use by the HostAgent.

The HostAgentUser maintains security information for remote X509 caches, specifically object filtering; you should update and manage the password for this user as you do for DataCurator.

### CodeLibrarianUser

The CodeLibrarianUser's UserGlobals contains classes that support the new package management in v3.5.

All Cypress-prefixed class names have been moved into this space.

## Changes in GsObjectSecurityPolicy

### Names

Instances of GsObjectSecurityPolicy now know their name, and methods have been implemented, `name` and `name:`. The actual name is kept in a dynamic instance variable `#_name`.

### Display Changes

The `asString/printString` display of a GsObjectSecurityPolicy has been changed to include the name. For example,

```
DataCuratorObjectSecurityPolicy(#2 in Repository
  SystemRepository, Owner DataCurator write, Group
  DataCuratorGroup write, World read)
```

### Added GsObjectSecurityPolicy for object filtering

There is an additional GsObjectSecurityPolicy in this version, the ObjectFiltersSecurityPolicy. The objects in this policy are not transmitted to remote X509 caches.

These are the GsObjectSecuritiesPolicies in a new 3.5 repository, and the permissions for the recently added policies.

1 SystemObjectSecurityPolicy

2 DataCuratorObjectSecurityPolicy

3 GsTimeZoneObjectSecurityPolicy

4 SecurityDataObjectSecurityPolicy

5 GsIndexingObjectSecurityPolicy

6 PublishedObjectSecurityPolicy

7 GcUserObjectSecurityPolicy

8 NamelessObjectSecurityPolicy

9 CodeLibrarianUserObjectSecurityPolicy
Owner CodeLibrarianUser write, World read

10 HostAgentUserObjectSecurityPolicy
Owner HostAgentUser write, World read

11 ObjectFiltersSecurityPolicy
Owner DataCurator write, Group DataCuratorGroup write, World read

Note that if your repository already contains security policies with the given indices, these remain unchanged; the new GsObjectSecuritiesPolicies are added starting at the available index.

## UserProfile instance variable renamed

UserProfile's instance variable `spare3`, which has been in use for loginHook functionality, has been renamed to `loginHook`.

## New authentication scheme for X509-Secured GemStone

#X509 is now one of the authentication schemes (in addition to #GemStone, #UNIX, #LDAP, and #SingleSignOn). Users with #X509 authentication may only login via X509-Secured GemStone.

However, unlike other authentication schemes, users with any authentication can also login using X509-Secure GemStone, provided they have the appropriate user certificates.

The following methods have been added to support limiting authentication for specific users to X509-secured logins.

```
UserProfile >> authenticationSchemeIsX509

UserProfile >> enableX509Authentication

UserProfile >> x509loginStatus

UserProfile class >> newX509WithUserId:
```

## 2.2 Changes in Garbage Collection

### Repository reclaimAll with wait added

`Repository >> reclaimAll` is usually invoked after a mark/sweep garbage collection, to ensure all garbage is reclaimed. However, `reclaimAll` cannot do the reclaim until several other steps are completed, including each session's voting on possible dead. To allow reclaimAll to be invoked immediately after a mark/sweep, rather than checking for progress in the overall garbage collection sequence, a method that includes a wait has been added.

`Repository >> reclaimAllWait:` *timeLimitSeconds*

Explicitly triggers the reclamation of all shadowed objects. If the Reclaim GcGem configuration parameter #reclaimDeadEnabled is true, then it waits until the dead objects are ready to reclaim (voteStateIdle) before it starts the reclaim.

If the *timeLimitSeconds* value is -1, then it waits indefinitely for the reclaim to complete. This should be done with caution, as under certain conditions the session could appear to wait forever. To avoid this, make sure that other sessions are committing/aborting to allow voting on possible dead to complete.

This method signals an error if:

- The voteState is not idle and AdminGem is not running.
- ReclaimGem is not running.
- The timeout expires before the reclaim completes. Explicit errors are returned to indicate whether it was waiting for voteStateIdle or the reclaim.
- No progress is detected for 5 minutes. A warning is logged if a session takes more than 20 seconds to vote.

### Setting persistent GcGem parameters

GcGem parameters for the Reclaim and Admin GcGems are set in two places:

- The persistent location is in the GcUser's UserGlobals, which contains parameter names and values. Logging in as GcUser and updating the values does not affect a currently running GcGem; the values take effect when the GcGem is restarted. Also, these values are not range checked until the GcGem is restarted, at which time range errors appear in the appropriate log file.
- Transiently, using previously existing `setAdminConfig... and setReclaimConfig...` methods. These methods performed range checking and take effect immediately, but did not persist through GcGem restart. When the GcGem restarted, the value reverted to the persistent value in the GcUser's UserGlobals.

To allow you to set both the persistent and transient values in a single operation, and to provide range checking for persistent settings, the following methods have been added:

`System class >> setPersistentAdminConfig:` *configSymbol*
`toValue:` *aVal*

`System class >> setPersistentReclaimConfig:` *configSymbol*
`toValue:` *aVal*

These methods update the value in GcUser's UserGlobals, and update the currently used value by the appropriate GcGem, and if these operations are successful (if the values are in range and the user has appropriate privileges), commits the changes.

These methods perform an abort and will error if there are uncommitted changes in the image.

The GcUser's UserGlobal associations remain the location of record, so there is no change required to existing code that reads or updates these settings.

### Parameter descriptions

The configuration parameters and the maximum/minimum and default values for the Reclaim and Admin GcGems were previously documented by a single string at the value `#defaultValues` in the GcUser's UserGlobals.

Now, the Reclaim and Admin GcGem parameters, which are entirely disjoint as of recent releases, are documented by separate strings, at the key `#defaultAdminConfigValues` and `#defaultReclaimConfigValues`. These are now available when logged in as other userIds via the new methods:

```
System class >> adminGemConfigs
```

```
System class >> reclaimGemConfigs
```

This returns a string containing the details for each parameter that applies to the given GcGem. For example

```
'#adminVerboseLogging
   Controls the amount of logging information written to the
   log file.
   If set to false, normal logging information will be written.
   If set to true, extra logging information is written to the
   log file.
     (default: false)

 #epochGcMaxThreads
   MaxThreads used for next epochGc
     (default:1, min:1, max: 32)
...
```

GcUser's UserGlobals no longer contains the key `#defaultValues`.

### Reset to default

To reset the configuration values to the default:

```
System class >> setDefaultAdminConfigs
```

```
System class >> setDefaultReclaimConfigs
```

### objectTableCleanupIdleTime removed

The GcUser's UserGlobals' entry `#objectTableCleanupIdleTime` is obsolete and has been removed.

### maxTransactionDuration minimum reduced

The minimum value for `#maxTransactionDuration` was previously 10 and is now 1.

## 2.3 Changes in Backup

### Compressed backups using lz4 have improved compression

Lz4 backups could be significant larger than the gzip version. The default lz4 file compression level has been changed from 0 to 3, reducing the resulting compressed backup size differential.

### Repository >> fullBackupTo: can accept an array

When fullBackup was made multithreaded, the longest complete form (`fullBackupTo:MBytes:compressKind:bufSize:`) was designed to accept either a single filename or an array of filenames. The restriction of fullBackupTo: to a single file was unnecessary, and has been lifted; this method can now be used with an Array of 1 or more filenames.

### restoreFromBackup: accepts filename without compression suffix

Backup files may be uncompressed, or compressed using .gz or .lz4. Previously, the compression extension was required. Now, this can be omitted. If no exact match is found for the specified backup filename, the system will look for that filename with .gz suffix, then for that filename with .lz4 suffix.

## 2.4 GemStone access without a Stone / Solo sessions

Using a "solo" session, you can login and perform many GemStone operations when a Stone is not running. This is designed to allow Smalltalk-based scripting, for single-user operations that do not require updating persistent state in the session's own environment.

By default, a solo login uses the clean, empty distribution extent, and no additional setup is required to perform a topaz solo login and execute GemStone Smalltalk code.

You may also use another extent, either empty or containing application code, data or other modifications, provided the following are true for the repository extent:

- The extent must not be part of a multi-extent repository.
- The extent file must either have read-only file permissions, or it will be exclusive-locked by the solo session.
- If the extent file is not read-only, it must have been previously cleanly shutdown by the Stone.

The configuration parameter GEM_SOLO_EXTENT specifies the extent file to be used by a Solo session. This defaults to `$GEMSTONE/bin/extent0.dbf`.

From the environment of a solo session, there is no Stone. Many System class and Repository methods are disallowed in a Solo session, since they would require a connection to a Stone. This includes `markForCollection`, `reclaimAll`, and methods that make and restore backups. Attempting to execute these methods would result in an ImproperOperation Error (#2050).

## Solo login from topaz

To login Solo from topaz linked or RPC, execute `set solologin on`, then login.

For example:

```
topaz> set solologin on
topaz> login
[Info]: LNK client/gem GCI levels = 35001/35001
[Info]: Read-Only Repository:
/benton2/users/lalmarod/GS6435/bin/extent0.dbf
[Info]: using libicu version 58.2
[Info]: Gave this process preference for OOM killer: wrote
to /proc/20027/oom_score_adj value 250
[03/26/2018 16:40:33.628 PDT]
  gci login: currSession 1  linked session
successful Solo login
topaz 1>
```

Values set for the Stone using **set gemstone** are ignored if **set solologin on** is set.

topaz RPC, which may have multiple sessions logged in, is not limited to all logins being either ordinary or solo logins. You have both kinds of logins within the same topaz RPC.

## Object creation and memory use

Each Solo RPC or linked Gem also opens a 10MB read-write temporary file, `/tmp/gemRO_<pid>_extent1.dbf`, which is deleted on logout or process exit.

Object creation in a Solo session is limited to temporary object memory, but you may create objects as needed up to the limit of memory. To ensure there is sufficient memory, you may:

- Set a larger value for GEM_TEMPOBJ_CACHE_SIZE in the configuration file used by the topaz or Gem session.
- For linked sessions, use -T *cachesize* on the topaz command line.
- For RPC sessions, include -T *cachesize* in the NRS gemnetid login parameter.

## Solo sessions other than from topaz

Solo sessions are not limited to topaz. When the GCI flag #GCI_LOGIN_SOLO is used in the login parameters, any GCI application may create a solo login.

The following image methods are available:

`GemStoneParameters >> soloLoginFlag`

Used by `GsExternalSession >> loginSolo`; returns the bits equivalent to GCI_LOGIN_SOLO.

`GsExternalSession >> loginSolo`

This specifies that the external session should be logged in as a solo session, using the GEM_SOLO_EXTENT value in the config file used by the gem process for the new session, or the value passed in with the Gem NRS 'gemnetobject -C GEM_SOLO_EXTENT=*extentname*').

`GsSession >> isSolo`

Returns true if the receiver is logged in as a solo session, false otherwise.

`GsSession class >> isSolo`

Returns true if the current session is logged in as a solo session, false otherwise.

## Scripting with topaz solo using she-bang

You may now create executable text files containing GemStone code, and execute these in a solo session at the command line. The following restrictions apply:

- The environment variable $GEMSTONE must be set
- she-bang scripts execute in a solo session, that is, do not log into a running stone.
- she-bang scripts run in a linked topaz session
- no topaz command line options can be passed in
- No .topazini files are read

The first line of the she-bang can be defined two ways:

`#!/usr/bin/env topaz`

This form requires that `$GEMSTONE/bin` be on the machine executable search path.

`#!`*fullPathToExecutable*`/topaz`

This does not require topaz on the machine search path ($GEMSTONE must still be defined). *fullPathToExecutable* must be the full path; she-bang will not expand an environment variable.

For example, an executable text file with the following contents:

```
#!/usr/bin/env topaz
set u DataCurator p swordfish
login
run
  | files sz |
  files := GsFile
      contentsOfDirectory: '$GEMSTONE/data/tranlog*.dbf'
      onClient: false.
  sz := 0.
  files do: [:ea | sz := sz + (GsFile sizeOfOnServer: ea)].
  'tranlogs consume ', (sz / 1024) asInteger asString, ' KB'.
%
```

Can be executed, without a running stone, to report the sum total size of tranlog files in the data directory. Topaz exits when the script completes.

# 2.5 Other Changes related to administrative operations

## Local host reported as localhost

Some output that previously provided the complete IPv4 address and the host name that this resolved to, now return localhost when the host is the same as the Stone's host.

For example, cases that would previously report `::ffff:204.45.141.15` and `kata.gemtalksystems.com`, if on the local node, now report `127.0.0.1` and `localhost`.

This change affects the Stone login log produced when STN_LOGIN_LOG_ENABLED is set, and element 11 of the Array returned from `System class >> descriptionOfSession:`, among others.

Related changes also resolve issues with using 0.0.0.0 as a logsender listening address (bug 47787)

## Printing of socket addresses changed

The printing of IPv6 mapped IPv4 socket addresses to log files and the results for methods such as `GsSocket >> peerAddress` have been changed to remove the`::ffff:` prefix.

## GEMSTONE_NRS_ALL and handling of NetLDI child process logs

Now, the forking of a gem or pgsvr by the NetLDI does not alter the GEMSTONE_NRS_ALL that was inherited from the NetLDI. This ensures that the directory locations specified by the NetLDI are not modified such that a remote process acquires invalid information.

The startnetldi -D option now controls the default location for child log processes, in the absence of other specifications, and is the recommended way to configure child log file directory locations. You do not need to specify the %D in the NRS for the child process for this to take effect. There are other changes in how -D is handled; see "NetLDI log directory handling" on page 33.

The precedence for log file locations is:

1. If there is setting for **#dir** in the Gem login parameters, this directory is used as the working directory and as the log files location.
2. Otherwise, if there is a setting for **#dir** in the GEMSTONE_NRS_ALL in the client environment, this directory is used for log files.
3. Otherwise, if there is a setting for **#dir** in the GEMSTONE_NRS_ALL in the environment in which the NetLDI was started, or if the **startnetldi -D** argument was specified, this is used for log files. Note that the NetLDI will not start with both of these specified as different locations. Like **#dir**, **-D** defines the working directory for the process as well as the log file directory.
4. Otherwise, the home directory of the unix user.

# Changes in System class >> descriptionOfSession:

## Change in return values for invalid or partially logged in sessions

If the session argument is not a valid session, or is partially logged in or out, `descriptionOfSession:` now returns nils, zeros, and empty strings rather than entirely nil elements.

## Added elements

The Array returned from `System class >> descriptionOfSession:` has added elements:

23. The sessionId of the HostAgent session through which this session is communicating to stone, or -1 if the session is not using a HostAgent.

24. If the specified session is a HostAgent, the HostAgent's listening port. Otherwise -1.

# performOnServer: using alternate shells

The method `performOnServer:` by default executes using /bin/sh. It is now possible to specify that a different shell be used, using the new method:

`System class >> performOnServer:` *aString* `withShell:` *aShellOrNil*

If *aShellOrNil* is nil, '/bin/sh' shell is used. Otherwise, the argument should be the path to a shell that will be used to process the command via the -c argument.

Korn Shell (/bin/ksh), Z Shell (/bin/zsh) and bash (/bin/bash) are known to work.

C Shell (/bin/csh) and its variants (/bin/tcsh) are known to not work.

# Disconnecting from one mid-level cache and connecting to another

When there are multiple mid-level caches available to a Gem, it is now possible to cleanly disconnect from using one of them as a mid-level cache, and reconnect to a different one. This can be used with both regular and X509-secured mid level caches.

This is done using the added method:

`System class >> midLevelCacheReconnect:` *hostName*

If already connected to a mid level cache, disconnect from that cache. Then, attempt to connect to an existing mid-level cache on the specified host, using the same NetLDI name that was used to login to the stone.The session's gem process must be on a machine different from the machine running the stone process in order for this to succeed.

# Determining what mid-level cache a session is connected to

The following new method allows you to confirm what node the current session is using as a mid-level cache. It returns the IP address, or nil if no mid level cache is in use.

```
System class >> midLevelCacheAddress
```

## C stack traces on Windows for fatal errors

On Windows, access violation errors (e.g. SEGV) from RPC GCI will now provide a C stack for subsequent debugging. All GCI calls will handle access violations this way. The stack is written to the GCI trace file, if that is active, or to a file with the name GCI*.tmp in the windows temporary directory,

```
c:\users\userid\AppData\Local\Temp\
```

Note that the .pdb files from the GemStone Windows Client distribution must be present in order to capture a stack trace.

## Handling cygwin file names on windows

When running GemStone under cygwin, path names are now automatically corrected to the windows-compatible format.

# 2.6 Changes in Errors

## New Error Classes

`CryptoError`

This is a subclass of ExternalError. It is an exception signaled when certain encrypt/decrypt methods fail.

Error number: 2756. C mnemonic: ERR_CryptoError

`MigrationError`

This is a subclass of Error. It is an exception that can be signaled when a migration operation cannot be completed, with one argument: the object for which migration failed.

Error number: 2757. C mnemonic: ERR_MigrationError

## Added Error Numbers associated with existing error classes

ABORT_ERR_GemAutoAbort 3007
image symbol: #errAutoAbort class: TransactionBacklog
Gem process serviced a sigAbort while waiting for a GCI command while in #manualBegin transactionMode, and while #GemAutoServiceSigAbort is true.

ABORT_ERR_GemAutoLostOt 3008
image symbol: #errAutoLostOt class: RepositoryViewLost
Gem process serviced a LostOt while waiting for a GCI command in #manualBegin transactionMode, and while #GemAutoServiceSigAbort is true.

ERR_CACHE_FAILURE 4097
PageCacheConnectionFailure or PageCacheSType::fatalError

ERR_SOLO_SESSION 2050 class: ImproperOperation
Operation not allowed or supported in a Solo session. Args: (1) details (a String)

GS_ERR_HA_TRANSPORT 4018
Transport layer error between a gem and a HostAgent thread

GS_ERR_SEM_FAILURE 4017
Semaphore failure during shared memory communication to stone

RT_ERR_BREAK_FROM_STONE 6020
Break, used from stopHostAgent OOB

## Changes in Error message arguments

Error 2025 #repErrCantCreateFile, has an additional argument; certain extent add operations, such as adding an extent on an NFS-mounted drive when NFS is disallowed, will now return more error details.

Error 2160 #rtErrSelectiveAbort, now is used for multiple error conditions and an additional argument is used to hold the error condition.

The printing for Transaction error 2407 / rtErrObjIsNp (object cannot be committed) has been updated to include the OOP of the object.

## 2.7 Changes in Utilities

### startnetldi

#### -t option no longer implemented

The **startnetldi -t** *timeout* argument is no longer implemented.

#### NetLDI log directory handling

If the **-D** argument to **startnetldi**, and the **#dir** specified in a the NetLDI’s GEMSTONE_NRS_ALL environment variable, are both set to specify explicit paths and are different, then startup is disallowed.

If neither the NRS of an incoming request, nor the NetLDI’s GEMSTONE_NRS_ALL environment variable, contain a **#dir**, and if **startnetldi -D** is specified, the NetLDI will use the value of the **-D** argument, rather than the child's home directory, for the working directory of the child process and the location in which to place child process log file.

### copydbf changes

The copydbf utility does not make certificate-based connections.

The **copydbf -p** option has been removed. This was used to specify pageservers for remote copydbf; with modern network connections, it was no longer useful to support copydbf operations using page servers.

### pageaudit additional statistics

pageaudit, in addition to reporting if there are any errors found in the page allocation with a repository, also reports statistics about the repository pages. This information is written to the pageaudit log file, *tmpstnname*`-pageAudit.log`.

The additional lines of information include:

```
OT Internal Pages                0.05 Mbytes          3 pages
OT Leaf Pages                    0.81 Mbytes         52 pages
Empty OT leaf pages              0.00 Mbytes          0 pages
Empty data pages                 0.00 Mbytes          0 pages
Data pages with 25%+ free        5.94 Mbytes        380 pages
Data pages with 50%+ free        4.05 Mbytes        259 pages
Free space in data pages         4.25 Mbytes     272.25 pages
```

### gemnetobject arguments and documentation

The **gemnetobject** script, and related scripts **gemnetdebug**, **gemnetobject_slow**, **gemnetobject_noop**, and **gemnetobject_keeplog**, now have better documented arguments. The -h output has been updated and expanded, a man page added for **gemnetobject**, and **gemnetobject** is described in the *System Administration Guide*, Appendix B, with the other utilities.

In addition to the existing arguments, you may include additional arguments, with or without a leading dash and letter, at the end or within the known arguments. Unrecognized arguments and tokens are passed along to the gem and ignored.

The **-U** argument has been specifically added to allow passing in arbitrary strings; this ensures that future changes to the **gemnetobject** arguments will continue to support **-U**.

Note that minimal error checking is done; you should verify that your arguments were entered correctly by examining the gem log file.

All **gemnetobject** arguments are passed to the gem and available by `System class >> commandLineArguments`.

## startlogsender/startlogreceiver

### Hot standby change in arguments for startlogsender

The way master tranlog directories are specified for **startlogsender** has changed in this release. The **-s** *stoneName* and **-T** *tranLogDir* arguments are now mutually exclusive, so most hotstandby configurations will require update.

This change streamlines the process of specifying the tranlog directories, and allows configurations to be setup such that new tranlog directories can be added at runtime.

With this change, there are two disjoint ways to specify the set of tranlog directories that will be used by the logsender:

- Explicitly provide a list of up to 20 directories using the **startlogsender -T** argument**.**
- Using the **startlogsender -s** argument to specify the Stone. The Stone's transaction log directories (specified by STN_TRAN_LOG_DIRECTORIES) will be used as the set of tranlogs used by the logsender. New transaction logs can be added at runtime (using `Repository >> addTransactionLog:size:`), and will be automatically picked up by the logsender.

There are no changes in the logreceiver, which accepts tranlog records from the logsender as in previous releases.

The maximum number of STN_TRAN_LOG_DIRECTORIES is now 100; this includes both tranlog directories defined in configuration files and those added at runtime.

### logsender and logreceiver stats for block ID

Currently, the logsender and logreceiver can record, in statmonitor data, only the tranlog file being processed. Now the block ID is recorded in SessionStat00 in logsender, and SessionStat23 in logreceiver. (#47685)

## New utility largememorypages

In previous releases, specialized commands to a pgsvr shell were required to compute the number of large/huge memory pages to configure on Linux or AIX. The pgsvr shell is no longer usable for this purpose in v3.5. Now, this information can be computed using the new utility **largememorypages**.

**largememorypages** computes the number of large or huge memory pages that will be required for a given shared page cache size and other system requirements, using a UNIX-standard command line interface.

Large or huge memory pages are available for use with GemStone on Linux and AIX. On Linux, GemStone supports 2MB and 1024MB (1GB) large pages. On AIX, 16MB large memory pages are supported. Large pages are not supported on other platforms.

Three arguments are required to do the computation: the SPC size or number of pages, the number of processes, and the number of shared counters. The memory large page size is optional and only applies on Linux. Arguments can be provided by settings in a configuration file or using command line arguments. Command line arguments have precedence over settings in the configuration file.

```
Usage: largememorypages [-e path ] [-z path] [-F cacheFrames | -M cacheKB]
      [-P maxProcesses] [-C maxSharedCounters] [-p largeMemoryPageSize]
where:
```

-e *path*
specifies an executable config file (same as startstone -e).

-z *path*
specifies a system config file (same as startstone -z).

If a config file is not specified, then -M or -F and -P and -C are required.

-F *cacheFrames*
shared cache size expressed in 16 KB frames

-M *cacheKB*
shared cache size with a optional units suffix.

-P *maxProcesses*
setting for SHR_PAGE_CACHE_NUM_PROCS

-C *maxSharedCounters*
setting for SHR_PAGE_CACHE_NUM_SHARED_COUNTERS.

-p *largeMemoryPageSize*
large memory page size. Not allowed on AIX, which always uses 16MB. Optional on Linux, which defaults to 2MB. May be used to specify 1GB page size on Linux. Overrides setting for SHR_PAGE_CACHE_LARGE_MEMORY_PAGE_SIZE_MB.

For example, on Linux, using the default GemStone configuration file:

```
unix> largememorypages -e $GEMSTONE/bin/initial.config
Cache config is 4736 pages = 74 MB,  total is 122 MB,  overhead
   64% of configured size
Large page size requested is: 2 MB.
Large page overhead: 1.16 MB
For 4687 pages, 52 processes and 1900 shared counters, 0
   pusherThreads
  required cache size is 127926272 bytes.
Number of 2 MB large pages required: 61
... <further instructions for update>
```

For more information, refer to the *Installation Guide* for Linux and for AIX.

## gslist no longer distributed on Windows

The **gslist** utility was previously included in the distribution on Windows, but was of minor value since no GemStone server processes can run on Windows. It is no longer included in the Windows distribution.

## startstone and startnetldi return codes

Previously, if **startnetldi** or **startstone** detected a netldid/stoned that was already running, it returned status 0; while a startup failure returns status 3, HOST_EXIT_ERR.

Now, if these processes detect an already running process, they will return status 2, HOST_EXIT_WARNING.

# 2.8 Topaz changes

## New command line argument -S for scripting

The topaz **-S** *filename* command line argument allows you to pass in a topaz script, similar to the **-I** option. The difference is that **-I** is intended for initialization, and suppresses all output (until the first error), while **-S** only suppresses topaz echo output. In combination with **-q**, which suppressed topaz environment output, a topaz script now can have complete control over the output display.

Output is stops being suppressed after the first error, or failed **expecterror** or **expectresult**, for error diagnosis. If the topaz has an interactive shell, processing also stops, to allow debugging.

## Improvements for support for additional compilation environments

Topaz commands are all now environment aware. A number of commands have been updated to correctly handle environments over envId 0 (the default GemStone environment). The changes include both bug fixes, new behavior, and specific definitions for behavior that was previously unclear.

If you are using environments > 0, refer to the topaz help text for the individual commands for specific defined behavior.

In addition, executing the **env** command with no argument now prints the current environmentId.

## X509-Secure GemStone login arguments

The login parameters for X509-Secured GemStone are a disjoint set from the standard login parameters. **set** has the following added options to specify the X509-secured login parameters. If any of these are set, the standard login parameter fields are cleared, and vice versa.

**netldi: logfile: key: extragemargs: directory: cacert: cert:**

For details, see the *GemStone/S 64 Bit X509-Secured GemStone System Administration Guide*.

## BREAK accepts new method specification syntax

The **break** command now accepts class >> method syntax,

*className* `[class] [(`*implementingSuperclassName*`)] >>` *methodName*

If a class *implementingSuperclassName* is specified in parenthesis after the *className*, *implementingSuperclassName* is used for lookup, not *className*.

For example:

```
topaz 1> break Delay >> wait
topaz 1> break SmallInteger class (Integer) >> fromStream: @4
```

## topaz objectSpec parsing numbers

Previously, object specs in the form of integers did not accept some reasonable inputs, for example base 16 notation. Now, these cases may use values that can be resolved as Integers using `Number>>fromString:`. (#47537)

## List classes allows subset filtering

The topaz command **list classes** lists all classes in all SymbolDictionaries. You may now specify by substring and only display classes with names that include that substring (case insensitive). For example,

```
topaz 1> list classes cert
GsX509Certificate
GsX509CertificateChain
```

## Omitting deprecated methods from method lists

The topaz **display** and **omit** commands have added the **deprecated** subcommand. This controls whether topaz commands that return lists of methods include or omit deprecated methods. By default, deprecated methods are included.

Deprecated methods are those that send one of the following messages:

```
deprecated
deprecated:
deprecatedNotification:
```

For example, the method `concurrencyMode` is deprecated:

```
topaz 1> implementors concurrencyMode
System class >> concurrencyMode
topaz 1> omit deprecated
topaz 1> implementors concurrencyMode
(Omitted 1 deprecated methods)
```

## Added topaz commands

### POLLFORSIGNAL

The command **pollforsignal** has been added, to support interactive testing of the GciPollForSignal code. This command causes topaz to wait for out-of-band activity from the Gem processing, using GciPollForSignal. It can be interrupted by control-C.

### TMETHOD to support code management

The code management subsystem under development uses a distinct Smalltalk method format. The new topaz command **tmethod** compiles a method that is formatted in this way. Code packaging is in active development and all functionality is subject to change.

## Subcommands added to topaz SET

### ENABLEREMOVEALL

**set enableremoveall on | off** controls whether topaz **removeallmethods** and **removeallclassmethod** commands are enabled or ignored. The default is **on**, allowing all methods to be removed.

### INPUTPAUSEONERROR

**set inputpauseonerror on | off** allows topaz processing to pause if an error occurs while executing an **input** command, making it easier to locate source file errors.

### SOLOLOGIN

**set sologin on | off** controls topaz solo login. When set sologin is on, subsequent logins will be solo logins, that is, logins without a Stone. For details, see “GemStone access without a Stone / Solo sessions” on page 26.

### ENVVAR

Used to set an environment variable within the environment of the topaz process. Requires two arguments, which will be interpreted as Strings. The word **envvar** cannot be abbreviated.

For example:

```
topaz 1> set envvar foo abc
```

## Subcommands added to OMIT and DISPLAY

### VERSIONEDCLASSNAMES

When **display versionedclassnames** is set, which is the default, topaz includes [*verionNumber*] following the class name, for instances of classes that are not the last one in the classHistory of that class. This is limited to when **display oops** is also set. **omit versionedclassnames** causes this version detail to be not displayed.

## Subcommand added to topaz STACK

### TERMINATE

STACK TERMINATE sends #terminate to the GsProcess of the current stack, if the stack is not owned by the scheduler.

# 2.9 Configuration Parameter Changes

## Parsing and Error handling for configuration files

The way configuration files are parsed has been reworked in this release.

- Syntax errors in configuration settings may now be treated as fatal errors, where in previous releases they were warnings, with the value ignored and defaults used.
- For some parameters, such as SHR_PAGE_CACHE_PERMISSIONS, an out of range value is now a fatal error, where previously it was a warning.
- Some configuration parameters that are not relevant for the specific process are not printed in the process log header.
- Some obsolete configuration values that were previously printed in log headers are now completely removed.

## Determining which Configuration files are in use

GemStone supports both system and executable Configuration files for both Gem and Stone. To determine the existence and names of these files, the following methods have been added:

```
System class >> stoneConfigurationFileNames
System class >> gemConfigurationFileNames
```

Each will return a dictionary with two keys, #GEMSTONE_SYS_CONF and #GEMSTONE_EXE_CONF, pointing to the full, expanded path and name of the system and executable configuration files, respectively, or nil if no file was specified or found.

### Added runtime-only configuration parameters

The configuration parameters GemConfigFileNames and StnConfigFileNames returns the names of the gem and system configuration files for the Gem and Stone, respectively. These are accessed using expression such as `System gemConfigurationAt: #GemConfigFileNames` and `System stoneConfigurationAt: #StnConfigFileNames`.

### Added keys in results of gem/stoneConfigurationReport

The result of `System class >> gemConfigurationReport` now includes the key GemConfigFileNames, and the result of `System class >> stoneConfigurationReport` includes the key StnConfigFileNames.

Each points to 2-element array containing system configuration file name or an empty string, and the executable configuration file name or an empty string.

## Removed Configuration Parameters

The following configuration parameters are not used in v3.5, and have removed. A warning is issued if they are included in a configuration file.

**GEM_PRIVATE_PAGE_CACHE_KB**

**STN_PRIVATE_PAGE_CACHE_KB**

## Changes in existing configuration parameters

### STN_COMMITS_ASYNC disabled

Testing encountered a case in which with this configuration parameter set to true, a record was missing from the transaction log during restore after the Stone ran out of tranlog disk space. (#48098) To avoid risk, this parameter has been disabled in v3.5.

### STN_TRAN_LOG_DIRECTORIES limited to 100

The maximum number of entries in STN_TRAN_LOG_DIRECTORIES is now 100.

### SHR_PAGE_CACHE_SIZE_KB rounded to up next MB increment

The calculation of the size of the shared page cache is based on SHR_PAGE_CACHE_SIZE_KB and on other parameters that compute overhead. The SHR_PAGE_CACHE_SIZE_KB value is converted to pages; the number of pages is now rounded up to the next 1 MB increment in size.

### SHR_PAGE_CACHE_NUM_PROCS provides additional count

The computation of SHR_PAGE_CACHE_NUM_PROCS based on STN_MAX_SESSIONS now provides one additional session to support threads in the ReclaimGem.

### DBF_ALLOCATION_MODE

If a list of values is used, the values should be in the range 5..200; values under 5 are disallowed for efficiency, and values up to 200 are now allowed.

### GEM_PGSVR_COMPRESS_PAGE_TRANSFERS, GEM_PGSVR_UPDATE_CACHE_ON_READ, STN_PAGE_MGR_COMPRESSION_ENABLED

The default for these parameters was previously false, and is now true.

### GEM_TEMPOBJ_MESPACE_SIZE and GEM_TEMPOBJ_POMGEN_SIZE

Previously, the maximum value was 1GB, now the maximum is 2GB.

### STN_LOG_IO_FLAGS

This previously documented as Solaris only, but can be used in Linux environments.

### GEM_TEMPOBJ_POMGEN_SIZE scaling

The size of the POM Generation memory space can be set explicitly using the configuration parameter GEM_TEMPOBJ_POMGEN_SIZE, but is normally derived from the GEM_TEMPOBJ_CACHE_SIZE setting.

Previously, the calculation of the default was linear, setting the POM gen space to about 0.8 times (80%) of the TOC size. For very large temporary object caches, this is larger than is useful. Now, the multiplier used is based on the size of the TOC, using interpolation on the line defined by these values:

| TOC size | Multiplier |
|---|---|
| <= 200MB | 0.8 |
| 600MB | 0.6 |
| 1GB | 0.27 |
| 5GB | 0.15 |
| >5GB | 0.1 |

## Added Configuration Parameters

### GEM_SOLO_EXTENT

Specifies an extent to be used for Gem session without a Stone. For details, see "GemStone access without a Stone / Solo sessions" on page 26.

### STN_ANONYMOUS_SSL

If true, all connections between the Stone and other processes are encrypted using anonymous SSL. Does not apply to X509-secured processes. For details, see "Loading a specific OpenSSL library version" on page 16.

## Added runtime-only configuration parameters

### DelayAutoServiceSigAbort

Runtime-only configuration option, applicable only in a session in which `System class >> clientIsRemote` returns true. Affects operation of GemAutoServiceSigAbort configuration option.

DelayAutoServiceSigAbort can be used to set an internal delay counter to a specified value, the delivery of the error 3007 will be delayed for the specified number of GCI calls received by the gem; this is to facilitate testing.

Runtime name: #DelayAutoServiceSigAbort
Default value zero, min 0, max 16r7FFFFFFF.

### GemAutoServiceSigAbort

Runtime-only config option, applicable only in a session in which System class >> clientIsRemote returns true. When the session's transaction mode is autoBegin or transactionless the gem process will automatically service any sigAbort received when there is no GCI traversal in progress and the gem is waiting for the next GCI command. After auto servicing of one or more sigAbort, the next GCI command will signal a TransactionBacklog with error number 3007.

After auto servicing of a sigAbort, the "GciAlteredObjs" portion of the traversal result of the next GciStoreTravDoTravRefs will include the result of writeSetUnion of commits by other sessions since last GciStoreTravDoTravRefs intersected with the PureExportSet.

Runtime name: #GemAutoServiceSigAbort
Default value: false

### GemCommitConflictDetails

Runtime-only config option for the gem.When true, `System class >> conflictsReport` and `System class >> transactionConflicts` return details about which session(s) caused commit conflicts. Must be enabled by `System gemConfigurationAt: #GemCommitConflictDetails put: true`, before the conflicting commit.

Runtime name: #GemCommitConflictDetails
Default value: false

### GemConfigFileNames

Runtime read-only configuration parameter, which returns the names of the system and executable configuration files for the Gem.

```
System gemConfigurationAt: #GemConfigFileNames
```

Returns an Array:

```
{ systemConfigFileName . exeConfigFileName }
```

### StnConfigFileNames

Runtime read-only configuration parameter, which returns the names of the system and executable configuration files for the Stone.

```
System stoneConfigurationAt: #StnConfigFileNames
```

Returns an Array:

```
{ systemConfigFileName . exeConfigFileName }
```

## Added X509-only Configuration Parameters

The following configuration parameters have been added to support X509-Secured GemStone. These configuration parameters are all intended to be used in the configuration file passed with the -E argument to startnetldi that is started on a remote or mid-level node, and have no effect otherwise. For details, see the *GemStone/S 64 Bit X509-Secured GemStone System Administration Guide*.

### GemRemoteCommit

If TRUE, an X509-secured remote gem will execute the critical region of commit in the session's thread in the pgsvr or HostAgent on stone host. Only applies, and is the default, if the gem host and the stone host have same byte order.

### NETLDI_HostAgentUser_cert

Specifies the HostAgentUser’s public key cert file for a mid-level cache node or for cache warming.

### NETLDI_HostAgentUser_key

Specifies the HostAgentUser’s private key file for a mid-level cache node or for cache warming.

## NETLDI_PORT_RANGE

Optional, specifies a port range for listening sockets in forked X509-secured Gems processes.

## NETLDI_START_MIDCACHE

If true, indicates that the node will be acting as a mid-level cache.

## NETLDI_WARMER_ARGS

Sets arguments to cache warming for an X509-secured mid or leaf cache. Arguments are **-n** *int* to specify the number of threads, **-d** to warm data as well as object table pages, and **-M** *midCacheHostName* to warm from a running mid-level cache on another node.

## SHR_PUSH_TO_MIDCACHES_THREADS

Specifies the number of threads to use to push changed pages committed by X509-secured Gems, from the Stone's cache to the mid caches.

# 2.10 Changes in Statmonitor and Cache Statistics

## Statmonitor automatic delete of older data files

You can now configure statmonitor with automatic new file creation to automatically delete older files. Using the **-K** *numFiles* flag to **statmonitor**, you can specify the number of older files to save (not including the file currently being updated). You may only use **-K** with either **-r** or **-R**.

## Removed statistics

The following statistics have been removed:

**LocalCacheAllocatedPceCount**
**LocalCacheFreeFrameCount**
**LocalCacheFreePceCount**
**LocalCacheOverflowCount**
**LocalCachePceCountLimit**
**LocalCachePceReclaimCount**
**LocalCacheStalePcesRemovedCount**
**LocalCacheValidPcesRemovedCount**
**NumCacheWarmers**
**ServiceMs**
**SpinLockFreeFrameSleepCount**
**SpinLockFreePceSleepCount**
**StnMessagesNeedWakeUpCount**
**StnMessagesNoWakeUpCount**

## Changed statistics

**WaitMs** now reports the average number of milliseconds to service an I/O, including both time spent in the I/O queue and time performing the I/O. Previously this only included the queue time.

## Added statistics

The following are now reported for Pgsvrs as well as for Gems:

**DepMapKeysChanged**
**GemHasCommitToken**
**TimeProcessingCommit**
**TimeStoneCommit**
**TimeWaitingForCommit**

The following cache statistics have been added:

**ClientPageReadsCommit** (Pgsvr)
Pages read by the HostAgent thread during commit.

**MessagesToStoneCommit** (Pgsvr)
The number of messages sent from a session to the stone during critical region of commit.

**PagePushListFullCount** (Stone)
Number of times a commit on stone host could not find a free entry in the page push list.

**PoorlyFilledPagesCount** (Gem)
Number of poorly filled data pages committed by this gem. This statistic is only updated during a commit.

**PriorityPagesNeedReclaimSize** (Stone)
The number of poorly filled pages that are pending, that is, the backlog waiting for the ReclaimGem reclaim task.

**PusherDroppedNewLife** (Pgsvr)
Pages in the page push list that had a new lifetime in stone's cache by the time the pusher thread processed that page in the list.

**PusherDroppedNotInCache** (Pgsvr)
Pages in the page push list that were no longer in stone's cache by the time the pusher thread processed that page in the list.

**PusherDroppedPoorFill** (Pgsvr)
Poorly filled data pages in the page push list that were skipped by a page pusher thread.

**PusherSentPages** (Pgsvr)
Pages sent to a mid cache by a page pusher thread in a HostAgent on the stone host.

**ReadWaitMs** (HardDisk)
The average number of milliseconds to service a read I/O, including both time spent in the I/O queue and time performing the read.

**ReceivedPages** (Pgsvr)
Pages received by a page receiver thread in a mid cache HostAgent.

**ReceivedPagesAlreadyInCache** (Pgsvr)
Pages received by a page receiver thread and ignored because the were already in the mid cache, or because a read of the page was in progress.

**ReceivedPagesCacheFull** (Pgsvr)
Pages received by a page receiver thread and ignored because the mid cache was full.

**ReceivedPagesReadInProgress** (Pgsvr)
Pages received by a page receiver thread and ignored because a read of that page was in progress.

**TempOopsAllocated** (All)
TempOopsAllocated is the number of oops allocated to a session, but not yet committed.

**TempPagesAllocated** (All)
TempPagesAllocated is the number of pages allocated to a session, but not yet committed.

**WriteWaitMs** (HardDisk)
The average number of milliseconds to service a write I/O, including both time spent in the I/O queue and time performing the write.

# Mac System and Host Stats added

## Statmonitor now collecting host stats on Mac

Previously, as Mac OS system statistics were not available, statmonitor on the Mac did not include arguments controlling the collection of process and system statistics.

Statmonitor on the Mac now supports the flags **-Q**, **-U**, **-X**, and **-Y**.

## Added Per-Process System Statistics

**ContextSwitches**
The number of times this thread has lost the CPU to another thread. Thread switches can occur either inside of a single process or across processes. A thread switch may be caused either by one thread asking another for information, or by a thread being preempted by another, higher priority thread becoming ready to run.

**CopyOnWriteFaults**
The total number of times a private copy of a shared page needed to be made due to a write to the shared page.

**MsgRecv**
The number of messages received by the process.

**MsgSent**
The number of messages sent by the process.

**PageFaults**
The total number of Page Faults by the threads executing in this process. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This will not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.

**PageIns**
The total number of times pages have been brought into memory from disk by the operating system's memory manager.

**ReadKBytes**
The total number of kilobytes read from the operating system I/O device.

**SystemCalls**
The total number of system calls.

**SysTime**
The number of milliseconds the process has been using the CPU to execute system calls.

**ThreadCount**
Number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.

**ThreadsRunningCount**
The number of threads owned by the process that are executing on the CPU.

**UserTime**
The number of milliseconds the process has been using the CPU to execute user code.

**WriteKBytes**
The total number of kilobytes written to the operating system I/O device.

## Added Mac OS Host Statistics

**ActiveFileMemoryKB**
The amount of memory used for buffering files that has been used recently.

**CopyOnWritePageFaults**
Number of copy on write page faults.

**CPUs**
The number of online CPUs on the local machine.

**ExternalPageCount**
Number of memory pages that are file-backed (non-swap).

**FreeMemoryKB**
The number of kilobytes of memory in the free list.

**InactiveFileMemoryKB**
The amount of memory used for buffering files that has not been used recently.

**InternalPageCount**
Number of memory pages that are anonymous.

**LoadAverage1**
The average number of threads ready to run over the last one minute.

**LoadAverage5**
The average number of threads ready to run over the last five minutes.

**LoadAverage15**
The average number of threads ready to run over the last fifteen minutes.

**PageCacheHits**
Number of Cocoa object cache hits.

**PageCacheLookups**
Number of Cocoa object cache lookups.

**PageFaults**
The total number of Page Faults by the threads executing in this process. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This will not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.

**PagesCompressed**
Number of pages which have been compressed.

**PagesDecompressed**
Number of pages which have been decompressed.

**PagesPagedIn**
The total number of pages that have been brought into memory from disk by the operating system's memory manager.

**PagesPagedOut**
The total number of pages that have been flushed from memory to disk by the operating system's memory manager.

**PagesPurged**
Number of pages which have been purged.

**PagesSwappedIn**
The total number of swapped out pages that have been brought back into memory from disk.

**PagesSwappedOut**
The total number of pages that have been moved from memory to disk due to a swap out operation.

**PercentCpuActive**
The percentage of the total available time that has been used to execute user or system code.

**PercentCpuIdle**
The percentage of the total available time that has been spent sleeping.

**PercentCpuSystem**
The percentage of the total available time that has been used to execute system (i.e. kernel) code.

**PercentCpuUser**
The percentage of the total available time that has been used to execute user code.

**PhysicalMemoryKB**
The amount of physical memory on the machine in kilobytes.

**Processes**
The number of processes in the computer at the time of data collection. Notice that this is an instantaneous count, not an average over the time interval. Each process represents the running of a program.

**PurgablePagesCount**
Number of purgable memory pages in use.

**ReactivatedPages**
Number of memory pages that have been reactivated.

**SpeculativePagesCount**
Number of memory pages holding data that was read speculatively from disk but has not yet been accessed.

**ThrottledPageCount**
Number of throttled pages.

**UnevictableMemoryKB**
The amount of memory that cannot be swapped.

**ZeroFilledPages**
The total number of pages have been block-cleared to contain all zeros.

Chapter

3

# Changes in GemStone Smalltalk

This chapter describes changes and new features important for programmers using GemStone Smalltalk, including:

# 3.1 Changes in Classes, compilation, and code management

## Code Management

This version includes preliminary work on the packaging and management of GemStone source code, including methods that support the tonel code format and the STON serialization format. This code is primarily within the CodeLibrarian's UserGlobals, but some support methods have been added to the image.

These new methods appear in the image with categories and/or method selectors that reference "tonel", "ston", or "metacello". All such methods are subject to change in future versions and should be used with caution.

### Added private methods for Tonel support

Methods are in the category ***tonel-gemstonecommon-core-private** include some methods integrated from a Pharo-compatible code base, which have functionality that duplicates existing GemStone methods. These private methods may be removed in a future release without prior deprecation.

### Cypress* classes no longer in Globals

All classes that were previously visible in the image with names beginning with 'Cypress' have been moved to the CodeLibrarianUser's UserGlobals.

## Compiling methods with reserved selectors

It is now allowed for sessions with CompilePrimitives privilege (as well as CodeModification) to compile methods with reserved selectors (see the *Programming Guide* for the list of reserved selectors). These selectors are still optimized and new definitions do not affect compiled code, but can be accessed, for example, via `perform:`.

## Added Behavior compile method

The following method has been added:

`Behavior >> compileMethod:` *sourceString*

Similar to `compileMethod:dictionaries:category:`, it compiles the given *sourceString* into the receiver class, using the current session's symbol list and into the category '(as yet unclassified)'.

## Added class creation options

### #selfCanBeSpecial

The subclass creation methods `options:` keyword may now include the option #selfCanBeSpecial. This is used in environments other than 0 with a modified class hierarchy, for classes in which self can be a special object, to avoid incorrect code being generated.

This option is not inherited from a superclass, but must be present in a superclass to be able to specify it for a subclass.

### Class byte size options

The following options: arguments are now accepted for Byte-format classes:

#'2byteWords' #'4byteWords' #'8byteWords' #'signed2byteWords' #'signed4byteWords' #'signed8byteWords'

These options allows specification of the word size of a byte format class, and allows underlying swizzling support for multi-byte word subclasses (such as DoubleByteString and QuadByteString).

## Multi-byte byte-format classes

Creating custom multi-byte classes that support automatic byte swizzling, and therefor are portable between big and little endian machines, can now be done by subclassing server classes.

The following classes have been added:

Uint16Array, Uint32Array, and Uint64Array

Int64Array, Int32Array, and Int16Array

These classes are defined using the new byte subclass creation options such as #'2byteWords', etc., and provide the additional methods that are needed to support byte swizzling. You may subclass these classes to create custom portable byte-format classes.

## Changes in handling some class options

The class options #dbTransient #instancesNonPersistent and #instancesInvariant are mutually exclusive, and redefining a class when there are only changes in adding these options updates the existing class and does not create a new class version.

Previously, redefining a class with a different one of these options would error as the new option was added while the old one remained. Now, redefining with a different one of these three options will remove any of the other options so the operation succeeds.

## Changes in handling Dynamic Instance Variables

Dynamic instance variables are per-instance instance variables, added using statements such as *anObject* `dynamicInstVarAt:` *key* `put:` *value*.

### GBS 8.4 Inspector display

In GemBuilder for Smalltalk v8.4, dynamic instance variables are now visible in inspectors.

### Handling during migration

Previously, dynamic instance variables were ignored during migration. Now, dynamic instance variables are added to the new instance during migration. As with any migration customization, override the method `migrateFrom:instVarMap:` in the new class version, to control instance variable mapping.

### Added methods

`Object >> copyDynamicInstVarsFrom:` *anObject*
Copy all dynamic instVar name/value pairs from *anObject* to the receiver.

`Object >> copyDynamicInstVars:` *aCollectionOfSymbols* `from:` *anObject*
Copy the dynamic instance variables with names in *aCollectionOfSymbols* from *anObject* to the receiver.

`Object >> dynamicInstVarPairs`
Returns the dynamic instance variables of the receiver in the form of an Array of name/value pairs.

## Class information when there are multiple versions of a class

The following methods have been added:

`Class >> versionNumber`
Returns the number of the receiver within its classHistory.

`Class >> versionedName`
Prints the name followed by [*versionNumber*], for classes that are not the last version in the classHistory.

`Class >> classHistoryAt:`
Equivalent to `classHistory at:`, returning the instance of class at the given position within the classHistory.

## Source code with control characters

GemStone source code may include literals for characters such as tab and lf, which may cause difficulties with source code management tools. To avoid such problems, GemStone code now accesses a set of literals using Class variables. The source code in $GEMSTONE/upgrade no longer includes these control characters in method sources.

- Character added class variables Backspace Cr Esc Lf NewPage Tab.
- Stream added class variables Lf Tab.
- AppendStream added class variables CrLf CrTab, as well as inherited Lf and Tab.
- WriteStream, including both WriteStreamLegacy and WriteStreamPortable, added class variables Cr CrLf CrTab, as well as inherited Lf and Tab.

## Pragmas have been made environment aware

Pragmas were previously stored in a class's extraDict attribute; they have been moved to be attributes of the GsNMethod itself. This allows Pragmas to be used per environment. Existing uses of Pragmas are unaffected.

The method `Behavior >> setPragmas:forMethod:`, which installed pragmas, is deprecated.

The method `Behavior >> pragmasForMethod:env:`, which fetches pragmas, has been added.

## Behavior methods for other environments

Code in environments other than the environment with envId 0 can be filed out now using methods in Behavior. The following methods have been added, which add an environmentId argument to similar existing methods:

```
Behavior >> fileOutCategory:on:environmentId:
Behavior >> fileOutClassDefinitionOn:environmentId:
Behavior >> fileOutClassOn:environmentId:
Behavior >> fileOutMethodRemovalOn:name:environmentId:
Behavior >> fileOutMethodsOn:environmentId:
Behavior >> fileOutPostMethodsOn:environmentId:
Behavior >> fileOutPreMethodsOn:environmentId:
Behavior >> removeAllMethods:
```

# 3.2 Transaction conflicts and object locks

## Detailed information on transaction conflicts

When two sessions commit changes that conflict with each other, the second session to commit gets a commit failure. Executing the method `System class >> transactionConflicts` reports the nature of the commit (Write-Write, for example) and the objects that were in conflict. However, it was previously impossible to determine any information about the commit that succeeded in making changes and was the root cause of the commit conflict.

It is now possible to configure your system to allow detailed reporting on the commit with which your commit conflicted.

First, you must enable tracking, by setting the runtime-only configuration parameter GemCommitConflictDetails to true. For example,

```
System gemConfigurationAt: #GemCommitConflictDetails put: true
```

This must be done before the commit transaction, since the work to collect the conflict information must be done at commit failure.

Then, once the commit fails, execute the new method `System class >> detailedConflictReportString`, which returns a string containing information about the conflicting other commit. For example,

```
System detailedConflictReportString
%
Commit failed , failure
Attempt to commit at: 2019-05-14 15:39:55.959
1 Write-Write Conflicts
  ( 12200193(a SymbolDictionary))
  1 commits by other sessions
  session 7 at 2019-05-14 15:39:45.607 userId DataCurator
    ( 12200193(a SymbolDictionary))
```

Commit conflict tracking requires extra work to be done in the gem for every failed commit, and an iteration over commit records; so it is only recommended that this be turned on for debugging commit conflicts.

## Other added methods on transaction conflicts

The following methods have also been added:

`System class >> conflictReportString`
Return a String describing the result of `System class >> transactionConflicts`.

**`System class >> conflictReportString:`** *conflicts*
The conflicts argument is Array returned by `System class >> transactionConflicts`. The report is limited to the first 100 oops in each category of conflict.

## Object locking changes

### Added lock report key #deferredUnlock

Reports on the kinds of object locks in the system, such as `sessionLocks` and `systemLocks`, now return #deferredUnlock, in addition to #read and #write.

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.

### New lock status method

The following method has been added:

`System class >> systemLocksDetailedReport`
Returns a String describing all of the locked objects. This is an expensive method, it makes a call to stone for each element of the result of System currentSessions. For each object, the oop and the class are printed.

For example:

```
session 5(
  (1 writeLocks: 11074305(a SymbolDictionary)))
session 6(  (1 readLocks: 240641(a Repository))
  (2 writeLocks: 979969(a UserProfile) 11074561(a
   SymbolDictionary)))
```

## 3.3 ClassOrganizer

### Filtering deprecated methods

The following instance methods have been added to ClassOrganizer:

```
includeDeprecatedMethodsInReports
includeDeprecatedMethodsInReports: aBoolean
```

By default, deprecated methods are included. By setting the value to false, deprecated methods are omitted from results that are lists of methods.

For example:

```
topaz 1> run
ClassOrganizer new implementorsOfReport: #concurrencyMode
%
System class >> concurrencyMode
topaz 1> run
ClassOrganizer new includeDeprecatedMethodsInReports: false;
      implementorsOfReport: #concurrencyMode
%
(Omitted 1 deprecated methods)
topaz 1>
```

### Include method oops in reports

The following instance methods have been added to ClassOrganizer:

```
includeMethodOops
includeMethodOops: aBoolean
```

By default, the OOPs of methods are not included. By setting the value to true, the oops are printed for methods, for results that are lists of methods.

For example,

```
topaz 1>
ClassOrganizer new includeMethodOops: true;
   implementorsOfReport: #today
%
Date class >> today   23101185
DateTime class >> today   5877505
topaz 1>
```

### referencesToLiteral: shortcomings

`ClassOrganizer >> referencesToLiteral:` returns methods that contain references to literals such as literal strings, numbers, and nil, and the offset of the reference within the source. This was failing to return the offsets (#47829), and, for symbols that were lookup references, returning references to symbol association value rather than the symbol itself. (#47102)

## 3.4 Parsing JSON

GemStone supports object serialization using JSON (JavaScript Object Notation), via methods `asJson` and `printJsonOn:`.

In v3.5, the class JsonParser has been added, which is subclassed from PetiteParser classes (PP*Parser). JsonParser supports parsing a JSON string to create an object structure.

Note that `asJson` writes the object structure and instance variables as dictionaries and Arrays; unlike passivation, the class itself is not preserved. When the object is parsed from the JSON into an object, more processing may be required by the sender to restore the classes of the original objects.

For example:

```
Date today asJson
'{"year":2018,"dayOfYear":304}'

JsonParser new parse: '{"year":2018,"dayOfYear":304}'
aDictionary( 'dayOfYear'->304, 'year'->2018)
```

## 3.5 Image level filein/GsFileIn added

The GsFileIn class has been added, to support image-based filein of code in GemStone fileout format. This implementation is not complete and is subject to change in future releases.

To use, send one of the `from*` methods. This also performs the file-in.

For example,

```
GsFileIn fromServerPath: 'mySourceCode.gs'
```

Only a subset of topaz commands is supported:

`doit, printit , run, nbrun`
`category:, category`
`method, method:, classmethod, classmethod:`
`removeallmethods, removeallclassmethods`
`commit, abort`
`input`
`send`
`env N`
`set compile_env N, set class, set class:` (other set commands are ignored)

The following are read but ignored:

```
expectvalue, expecterror, errorcount
iferr, iferr_list, iferr_clear
fileout, output,
fileformat
display, omit
level, limit, remark, status, time
list
```

Other topaz commands report an error.

# 3.6 Changes in Numerics

## Linux change affecting floating point #inexactResult exceptions

In Linux's libc v2.24 and later, the inexact exception has been removed from the arithmetic functions floor, ceiling, round and truncate. This impacts GemStone executing on the newly supported Ubuntu 18.04 distribution, which includes libc 2.27. Ubuntu 16.04, Red Hat and SUSE include older versions of libc.

This means that, if floating point exceptions are enabled, GemStone operations such as `9.75 ceiling` would signal an #inexactResult exception if running on Ubuntu 16.04 and do not signal any exception on Ubuntu 18.04.

For consistency of results across Linux and other platforms, operations such as floor, ceiling, round and truncate that invoke the OS calls will now ignore any #inexact exception, whether or not the underlying OS call signals inexact.

# 3.7 Other Changes and Enhancements

## Session Signals have additional argument

InterSessionSignal and the deprecated `System >> signalFromGemStoneSession` now include a fourth element, the number of pending signals.

The following methods have been added:

```
InterSessionSignal >> fromSession:signal:message:numPending:
InterSessionSignal >> numPending
```

## PetiteParser updated

The PetiteParser classes in GemStone have been updated with the code from PetitParser-JanKurs.290.mcz.

## GsFile nextPutAsUtf8: replaced by nextPutAllUtf8:

The method `GsFile >> nextPutAllUtf8:` has been added, and the equivalent method `GsFile >> nextPutAsUtf8:` will be deprecated.

This is compatible with other methods on AppendableStream and AppendableString.

## GsBitmap hiddenSetSpecifiers no longer include named Customer sets

`GsBitmap >> hiddenSetSpecifiers` previously included #CustomerSet1 through 5. Since individual GsBitmaps do not need identifiers, unlike the System hidden sets, static identifiers are unnecessary.

Also, #SaveDepMapChangedObjs and #SaveWrittenObjs have been removed, and #PreviousWsUnion has been added.

## Added Socket methods

The following methods have been added:

`GsSocket >> setCloseOnGc:` *aBoolean*
If *aBoolean* is true the receiver's underlying socket will be closed when the in-memory state of the receiver is garbage collected.

`GsSocket class hostIsLocalhost:` *hostNameOrIpAddrString*
Returns true if the host that *hostNameOrIpAddrString* resolves to is equivalent to localhost.

`GsSignallingSocket >>> acceptTimeoutMs:` *timeoutMs*
`errorOnTimeout:` *errOnTimeoutBool*
Returns a socket created for a new connection, or signal an Error if there was an error. If *errOnTimeoutBool* == true, timeout signal an Error, otherwise returns false on timeout. The result is an instance of `speciesForAccept`, and has non-blocking state equal to non-blocking state of the receiver.

`GsSocket >> read:` *maxBytes* `into:` *byteObj* `startingAt:` *index* `maxWait:` *timeoutMs*
Reads up to the given number of bytes into the given byte object (for example, a String). The first byte read will go into the position indicated by index. Returns the number of bytes read, 0 for EOF, nil if an error occurs, or false if the receiver is not ready to read within *timeoutMs*. If no data is available for reading the current GsProcess is suspended for up to *timeoutMs* until data arrives.

`GsSignalingSocket >> read:` *maxBytes* `into:` *byteObj* `startingAt:` *index*
`maxWait:` *timeoutMs*
Reads up to the given number of bytes into the given byte object. The first byte read will go into the position indicated by index. Returns the number of bytes read, 0 for EOF, or signals an error. If no data is available for reading the current GsProcess is suspended for up to *timeoutMs* until data arrives.

`GsSecureSocket >> secureAcceptTimeoutMs:` *timeoutMs*
`errorOnTimeout:` *aBoolean*
This is similar to the existing method `GsSecureSocket >> secureAcceptTimeoutMs:`, but provides the ability to return false rather than raise an exception.

## Added System Class methods

`System class >> currentSessionsReport`
Return a string describing the current sessions. For example,

```
2 GcUser reclaimgcgem 17447
3 GcUser admingcgem 17449
4 SymbolUser symbolgem 17451
5 DataCurator gem 662 on benton
6 DataCurator gem 26928 on benton
```

`System class >> stoneStartupId`
Returns an Integer, a 64bit random number created in stone startup.

`System class >> gemVersionReportString`
Returns a String with keys and values of `gemVersionReport`, one pair per line.

`System class >> stoneVersionReportString`
Returns a String with keys and values of `stoneVersionReport`, one pair per line.

`System class >> cacheStatusCount`
Returns a positive SmallInteger value. The value maintained in the stone process increments whenever a remote cache starts up, shuts down, or becomes a midCache.

`System class >> oldestCommitRecordAgeMilliseconds`
Return a SmallInteger representing the age of the oldest commit record in milliseconds. This value represents the elapsed time since the commit that created the oldest commit record.

## Added methods for GsExternalSession

### Login/logout messages

By default when using GsExternalSessions, both login and logout write a log message using `GsFile >> gciLogServer:`, and additional user logging can be invoked using `log:`; all these messages can be suppressed using `suppressLogging`.

Now, GsExternaleSessions can be configured to suppress these messages without affecting user logging using `log:`. The following methods have been added:

`GsExternalSession >> quiet`
Disables messages on login and logout

As of the previous version v3.4, GCI login/logout messages are no longer included in GsExternalSession output.

### Testing for login status

The following method has been added:

`GsExternalSession >> isLoggedIn`
Returns true if the external session is logged in.

## Added methods for Squeak support

The following methods are implemented on Object and inherited/overridden by other classes:

`Object >> squeakBasicAt:` *anIndex*
For an OOP format receiver, return the value at *anIndex*. For a byte format receiver, return the unsigned Integer value of the word specified by *anIndex*, using *Class* `_bytesPerWord` as the size of a word.

`Object >> squeakBasicAt:` *anIndex* `put:` *aValue*
For an OOP format receiver, set the value at *anIndex*. For a byte format receiver, set

the value at the word offset *anIndex* (calculating the offset using *Class* `_bytesPerWord`) to *aValue*.

`Object >> squeakBasicSize`
For an OOP format receiver, return the total number of instance variables. For a byte format receiver, return the size in words using *Class* `_bytesPerWord` as the size of a word.

`Behavior >> squeakBasicNew:` *anInteger*
Return an instance of the receiver, which must be indexable. If the receiver is OOP format, *anInteger* specifies the size; if byte format, the size in words, calculating the size using *Class* `_bytesPerWord`.

# Changes to Strings and ByteArrays

## Converting from ByteArray containing UTF-8 to decoded string

The following methods have been added:

`ByteArray >> decodeFromUTF8ToString`
The contents are bytes in UTF-8 encoding; decode and return as a String, DoubleByteString or QuadByteString.

`ByteArray >> decodeFromUTF8ToUnicode`
The contents are bytes in UTF-8 encoding; decode and return as Unicode7, Unicode16 or Unicode32.

As in recent 3.4.x versions, `Utf8 class >> withAll:` no longer accepts a ByteArray argument.

## Support for Base64 encoding

Methods have been added to convert Strings and ByteArrays to Base64 strings.

Base64 is a binary-to-text encoding schemes that translates data into radix 64 representation. Base64 originates from a MIME content transfer encoding.

GemStone's implementation allows conversion to include the output as 64-character lines or as a single line.

`CharacterCollection >> asBase64StringOnOneLine:` *aBoolean*
`ByteArray >> asBase64StringOnOneLine:` *aBoolean*
Return a String which represents the receiver represented in base64 format. If aBoolean is true, the resulting is one long line which does not contain newline characters. If *aBoolean* is false, newline characters are inserted such that each line does not exceed 64 characters.

`CharacterCollection >> asBase64String`
`ByteArray >> asBase64String`
Return a one-line String which represents the receiver represented in base64 format.

`ByteArray class >> fromBase64String:` *aString*
Creates an instance by decoding *aString*, which must be an instance of String or another single-byte character collection class. The argument must be in base64 format. The argument may contain newline and other whitespace characters, which are ignored.

## Support for Hex String decoding

A hex string is a character collection that contains characters in the ranges 0 - 9, a-f, and A-F. Hex strings can be created using `asHexString` methods.

The following methods have been added to convert such hex strings into ByteArrays, in which two-characters substrings of the hex string are converted to an integer element. The argument hex string may additionally include either the C prefix of 0x or 0X in the first two characters, or the Smalltalk hex prefix of 16r as the first three characters.

If the hex string has an odd number of characters, then padding is done, either to the left or right; the default is to left pad.

`ByteArray class >> fromHexString:` *aHexString* `leftPadded:` *padLeft*
Create a new instance containing the byte values contained in aHexString. If aHexString contains an odd number of hex digits, then the result object is padded according to the *padLeft* argument; if true, the result is left-padded, meaning the first four bits of the first byte in the result object will be zero. If *padLeft* is false, the result is right-padded, meaning the last four bits of the last byte in the result object will be zero.

`ByteArray class >> fromHexString:` *aHexString*
Create a new instance containing the byte values contained in aHexString. If aHexString contains an odd number of hex digits, then the result object is left-padded.

For example,

```
ByteArray fromHexString: '201' leftPadded: true
aByteArray( 2, 1)

ByteArray fromHexString: '201' leftPadded: false
aByteArray( 32, 16)
```

## Bitwise access to ByteArrays

The following methods have been added:

`ByteArray >> bitAtZ:` *zeroBasedOffset*
Returns a bit of the receiver, as if the receiver is an Array of bytes. The expression `((zeroBasedOffset bitShift: -3)+1)` defines the byte addressed in the receiver. Within that byte,
`bitAt:((zeroBasedOffset bitAnd: 7) + 1)` is returned.
Returns 0 or 1.

`ByteArray >> bitAtZ:` *zeroBasedOffset* `put:` *aBit*
Stores a bit into the receiver, as if the receiver is an Array of bytes. The expression `((zeroBasedOffset bitShift: -3)+1)` defines the byte addressed in the receiver. Within that byte,
`bitAt:((zeroBasedOffset bitAnd: 7) + 1)` is modified.
*aBit* must be either 0, 1, true, or false.

## ByteArray >> byteAt: and byteAt:put:

The methods `ByteArray >> byteAt:` and `byteAt:put:` have been added, which are the same as existing methods `at:` and `at:put:`.

## Other Added methods

`Time >> asStringMs`
Returns a String that expresses the receiver in local time in the format HH:MM:SS.sss, where sss are milliseconds.

`DateAndTime >> asStringMs`
Returns a String that expresses the receiver in local time in the format YYYY-MM-DD HH:MM:SS.sss, where sss are milliseconds.

`GsHostProcess >> killChild`
Terminate a GsHostProcess process that is running.

`GsHostProcess >> killChild:` *timeout*
Wait for the specified timeout and if the GsHostProcess child process is still running, terminate it.

# 3.8 Deprecations and removed methods

## Obsolete classes moved to ObsoleteClasses

The following classes are no longer in Globals:

```
Activation
CompiledMethod
EUCString
EUCSymbol
GsCloneList
InvariantEUCString
JapaneseString
JISCharacter
JISString
```

## Cypress Classes moved to CodeLibrarian's UserGlobals

Cypress classes with names **Cypress*** were moved to CodeLibrarian's UserGlobals; see the information under "Code Management" on page 50.

## Deprecated method

The method `AutoComplete >> strings:cluster:` has been deprecated; replace with `AutoComplete>>strings:`.

The method `Behavior >> setPragmas:forMethod:` has been deprecated. Pragmas are now generated into GsNMethod's debugInfo during compile.

## Removed private methods

The following private methods were removed:

```
Behavior >> _primCompileMethod:symbolList:category:oldLitVars:
   intoMethodDict:intoCategories:intoPragmas:environmentId:
Behavior >> _primitiveCompileMethod:symbolList:category:
   oldLitVars:intoMethodDict:intoCategories:intoPragmas:
   environmentId:
Class >> _validateOptions:withFormat:
GsHostProcess >> _waitChild
IndexManager class >> _loadHiddenSet:
IndexManager class >> _loadHiddenSet:fromFile:
```

```
Object >> __topazPerform0:withArguments:
Object >> _writeCypressJsonOn:
Object class >> _loadHiddenSet:fromFilesIn:
Object class >> _loadHiddenSet:fromFilesIn:withPattern:
PPEndOfFileParser >> acceptsEpsilon
PPEndOfFileParser >> exampleOn:
PPEndOfFileParser >> isNullable
PPEndOfFileParser >> nonEmpty
PPParser >> updateContext:
PPPluggableParser >> acceptsEpsilon
Repository >> _doListInstancesFrom:with:includeMemory:
Repository >> _fullBackupTo:MBytes:compressKind:bufSize:
Repository >> _listInstancesInMemory:
Repository:bufSize:privateDecryptionKey:passphrase:
Repository >> _primSecureFullBackupTo:MBytes:compressKind:
   bufSize:encryptKind:publicKeyCerts:signatureHashKind:
   signingKey:signingKeyPassphrase:
Repository >> _restoreBackups:scavPercentFree:bufSize:
SortedCollection class >> _loadHiddenSet:
SortedCollection class >> _loadHiddenSet:fromFile:
SymbolDictionary class >> _listClassesIn:
System class >> _conflictsReport:
System class >> _disallowSubsequentCommits
System class >> _remoteCachesListStartingAt:midLevelOnly:
System class >> _startGcCacheWarmer
```

## Upgrade support methods changed to use GsBitmaps

During upgrade and conversion from older versions, sorted collections must be resorted and indexes rebuilt. The methods that support this have been adjusted to use GsBitmaps rather than Hidden Sets. The following methods have been specifically replaced:

```
IndexManager class >> rebuildAllCharCollIndexes:hiddenSetForErr
   ors:
SortedCollection class >> convertAll:usingCache:hiddenSetForErr
   ors:
SortedCollection class >> convertInstancesFromFilesForGem:of:
   hsIdx:hsErrIdx:bmDirectoryPath:conversion:
SortedCollection class >> resortAll:hiddenSetForErrors:
SortedCollection class >> writeFailedConversionObjsToFileForG
   em:hiddenSetForErrors:
```

The added methods that replace them are:

```
IndexManager class >> rebuildAllCharCollIndexes:
SortedCollection class >> convertAll:usingCache:bmForErrors:
SortedCollection class >> convertInstancesFromFilesForGem:of:
   bm:errBm:bmDirectoryPath:conversion:
SortedCollection class >> resortAll:
SortedCollection class >> writeFailedConversionObjsToFileForGe
   m:bmForErrors:
```

Other methods that interact with these methods have internal changes.

# 3.9 Encryption Changes

## Support for SHA-3

The following methods have been added to CharacterCollection and ByteArray.

To compute SHA-3 of receiver, as a LargeInteger, use the following methods:

```
sha3_224Sum
sha3_256Sum
sha3_384Sum
sha3_512Sum
```

To compute SHA-3 of receiver, as hex string, use the following methods:

```
asSha3_224String
asSha3_256String
asSha3_384String
asSha3_512String
```

## Support for HMAC (Hash-based message authentication codes)

Methods have been added to compute the HMAC for md5, sha1, sha256, sha512, sha3-224, sha3-256, sha3-384, and sha3-512. This can be computed for Strings or ByteArrays, which are converted to a base64 String to compute. The argument *secretKeyString* must be a single byte object: a String, ByteArray or Unicode7.

The following instance methods have been added to ByteArray and CharacterCollection.

```
asMd5HmacWithKey: secretKeyString
asMd5HmacStringWithKey: secretKeyString
asSha1HmacWithKey: secretKeyString
asSha1HmacStringWithKey: secretKeyString
asSha256HmacWithKey: secretKeyString
asSha256HmacStringWithKey: secretKeyString
asSha512HmacWithKey: secretKeyString
asSha512HmacStringWithKey: secretKeyString
asSha3_224HmacWithKey: secretKeyString
asSha3_224HmacStringWithKey: secretKeyString
asSha3_256HmacWithKey: secretKeyString
asSha3_256HmacStringWithKey: secretKeyString
asSha3_384HmacWithKey: secretKeyString
asSha3_384HmacStringWithKey: secretKeyString
asSha3_512HmacWithKey: secretKeyString
asSha3_512HmacStringWithKey: secretKeyString
```

## Support for Symmetric-Key Encryption

Symmetric key encryption uses the same key to perform the encryption and the decryption (in contrast to asymmetric-key encryption, which use separate public and private keys).

AES encryption/decryption (Advanced Encryption Standard) is a block symmetric cipher, while ChaCha20 is a stream symmetric cipher.

AES supports several modes:

- CBC is an acronym for cipher block chaining.

- OCB is an acronym for Offset Cookbook Mode.
- GCM is an acronym for Galois Counter Mode.

ChaCha20 supports:

- Poly1305

OCB, GCM and Poly1305 are Authenticated Encryption with Associated Data (AEAD) modes. AEAD provides data authenticity, confidentiality, and integrity.

AEAD also supports Additional Authenticated Data (AAD). AAD is not encrypted and therefore not confidential, but its authenticity and integrity are guaranteed. If AAD is used, it is not included in the encrypted payload, but must be provided in order to decrypt the data. The additional data is optional, so the argument for this may be nil, in which cases it is not needed for decryption.

AES is performed using the OpenSSL open source package and the AES specification, available at: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

## Added methods

Methods have been added to CharacterCollection and ByteArray that allow you to sign and verify the signature of the receiver.

***Algorithm***`EncryptWith`***NNN***`BitKey:` *aKey* `salt:` *aSalt* `into:` *destObjOrNil* `tag:` *tag* `extraData:` *eData*
***Algorithm***`DecryptWith`***NNN***`BitKey:` *aKey* `salt:` *aSalt* `into:` *destObjOrNil* `tag:` *tag* `extraData:` *eData*

Encrypts/Decrypt the receiver using ***NNN*** bits and the algorithm ***Algorithm***.

***Algorithm*** may be:

- `aesOcb` for AES-OCB.
- `aesGcm` for AES-GCM.
- `chacha20Poly1305` for CHACHA20-Poly1305.

***NNN*** may be:

- 128. In this case *aKey* must be a ByteArray of size 16.
- 192. In this case *aKey* must be a ByteArray of size 24.
- 256. In this case *aKey* must be a ByteArray of size 32.

For example,

```
aesGcmEncryptWith128BitKey:salt:into:tag:extraData:
aesOcbDecryptWith256BitKey:salt:into:tag:extraData:
chacha20Poly1305DecryptWithKey:salt:into:tag:extraData:
```

and so on

*destObjOrNil* must be nil or an instance of a non-invariant byte object. If *destObjOrNil* is nil, the result of the operation will be placed into a new instance of ByteArray (encryption) or String (decryption); otherwise the result will be placed into the given byte object starting at offset 1.

The size of *destObjOrNil* will be modified to correctly contain all encrypted or decrypted data, and may differ from the size of the receiver due to the automatic addition or removal of padding by the cipher algorithm.

*aSalt* must be a ByteArray of size 12. *aKey* must be a ByteArray with the appropriate size for the method. The same key and salt must be used to decrypt as were used to encrypt.

During AEAD encryption, a tag is generated which is used during decryption to ensure data integrity. The tag data will be stored into the *aTag* argument, which must an instance of a byte object. *eData* must be nil or a byte object with a character size of one containing additional data to be used in generating the tag value.

For AEAD decryption, the tag argument *aTag* must be a byte object containing the tag bytes returned during encryption. *eData* must be a byte object containing the same bytes provided during encryption, or nil if no byte object was provided.

When encrypting a receiver that has a character size greater than one, data is placed into big-endian byte order before encryption.

When decrypting into a *destObjOrNil* object that has a character size greater than one, data is converted to big-endian byte order after decryption.

### Example

```
| myKey mySalt myTag encoded decoded |
myKey := ByteArray withRandomBytes: 32.
mySalt := ByteArray withRandomBytes: 12.
myTag := ByteArray new.
encoded := 'abc' aesOcbEncryptWith256BitKey: myKey
   salt: mySalt into: nil tag: myTag extraData: 'def'.
decoded := encoded aesOcbDecryptWith256BitKey: myKey
   salt: mySalt into: nil tag: myTag extraData: 'def'.
```

# 3.10 GsTlsCredentials

The classes GsTlsCredential with subclasses GsTlsPrivateKey, GsTlsPublicKey, and GsX509Certificate have been added. These classes encapsulate TLS (SSL) private keys, public keys, and X509 certificates, respectively. Instances contain a hidden reference to C pointer to the OpenSSL representation of the TLS object.

Instances of GsTlsPublicKey and GsTlsPrivateKey are used for digital signing and digital envelopes, as well as in X509-Secured external sessions.

GsX509CertificateChain, a subclass of Array, has also been added to support X509-Secured logins using the new classes GemStoneX509Parameters and GsX509ExternalSession.

## Creating a GsTlsCredentials

Instances are created by reading the PEM from a file or PEM-format string. The following new class methods can be used to instantiate any subclass of GsTlsCredential:

```
newFromPemFile:
newFromPemString:
```

## Fetch public key from private key or certificate

To determine key type and extract a public key from a private key or from an X509 certificate, the following methods are available:

`GsX509Certificate >> asPublicKey`
Return the Public Key associated with the receiver.

`GsTlsPrivateKey >> asPublicKey`
Answers a new object which represents the public key for the receiver. The class of the new instance is the result of the message `#speciesForPublicKey`, which returns GsTlsPublicKey.

`GsTlsKeyPublicKey class fromCertificate:` *aGsX509Certificate*
Extract the public key from the argument *aGsX509Certificate* and return a new instance of the receiver.

## Verifying public/private key pairs

The following method can be used to verify that two keys or X509 certificates form a matching public/private key pair:

`matches:` *anotherKey*
Return true if the receiver and *anotherKey* match each other as a valid public-private key pair, false otherwise. If the receiver is a public key, *anotherKey* is expected to be a private key, and vice versa.

If the receiver or *anotherKey* is an instance of GsX509Certificate, the public key is extracted from the certificate and used for comparison.

RSA and DSA key pairs match if both keys use the same modulus. Elliptic curve key pairs match if both keys use the same curve and the same point on that curve.

## Querying a GsTlsCredential for its algorithm

OpenSSL 1.1.1 supports a number of encryption and signing algorithms. To query an instance of a kind of GsTlsCredential for the algorithm, the following methods are available:

`algorithm`
Answers a Symbol indicating the type of high-level PKI (Public Key Infrastructure) algorithm the receiver uses. The high-level PKI algorithms supported are:
#RSA - Rivest-Shamir-Adleman
#DSA - Data Signature Algorithm
#EC - Elliptic Curve Cryptography

All high-level algorithm have various sub-types. Use the `sslAlgorithm` method to obtain information about the specific PKI algorithm of the receiver.

`sslAlgorithm`
Answers a Symbol indicating the SSL type of PKI algorithm the receiver uses. See the image comment for details. If the receiver is an instance of GsX509Certificate, the result is the algorithm of its public key.

`description`
Answers a String obtained from SSL describing the receiver. The contents and format of the string vary depending on the receiver's class.

## 3.11 Support for Digital Signatures

Methods have been added to CharacterCollection and ByteArray that allow you to sign and verify the signature of the receiver.

The methods are of the form:

`signWith`***AlgorithmAndKeyTypePadding***`PrivateKey:` *aGsTlsPrivateKey* `into:` *aByteArrayOrNil*
Hashes the receiver using *Algorithm* and signs the resulting hash with the *KeyType* and *Padding*. Returns a ByteArray containing the resulting signature.

`verifyWith`***AlgorithmAndKeyTypePadding***`PublicKey:` *aGsTlsPublicKey* `signature:` *aByteArray*
Hashes the receiver using *Algorithm* and verifies the resulting hash using the *KeyType* and *Padding*. Returns true if the signature is correct. Otherwise raises an exception.

Where the following ***AlgorithmAndKeyTypePadding*** are available:

`Ec`
Elliptic Curve key

`Sha1AndDsa`
#PKCS1 padding

`Sha1AndRsaPss`
PSS padding

`Sha1AndRsa`
#PKCS1 padding

`Sha256AndDsa`
#PKCS1 padding

`Sha256AndRsaPss`
PSS padding

`Sha256AndRsa`
#PKCS1 padding

`Sha3_224AndRsaPss`
PSS padding

`Sha3_224AndRsa`
#PKCS1 padding

`Sha3_256AndRsaPss`
PSS padding

`Sha3_256AndRsa`
#PKCS1 padding

`Sha3_384AndRsaPss`
PSS padding

`Sha3_384AndRsa`
#PKCS1 padding

`Sha3_512AndRsaPss`
PSS padding

`Sha3_512AndRsa`
#PKCS1 padding

`Sha512AndDsa`
#PKCS1 padding

`Sha512AndRsaPss`
PSS padding

`Sha512AndRsa`
#PKCS1 padding

For example,

```
signWithSha3_256AndRsaPssPrivateKey:into:
verifyWithSha3_256AndRsaPublicKey:signature:
signWithSha512AndDsaPrivateKey:into:
verifyWithSha512AndDsaPublicKey:signature:
```

### GsTlsCredential Added Methods relating to Digital Signatures

Instances of the class GsTlsCredential and its subclasses, GsTlsPublicKey, GsTlsPrivateKey, and GsX509Certificate, have the following added methods.

`supportsDigitalSignatures`
: Answers a Boolean indicating if the receiver supports digital signatures. The type of support offered, either signing or verifying, depends upon the specific class of the receiver.

`canCreateDigitalSignatures`
: Returns true if the receiver is an instance of a GsTlsPrivateKey, false for GsTlsPublicKey and GsX509Certificate.

`canVerifyDigitalSignatures`
: Returns true if the receiver is an instance of a GsTlsPrivateKey, false for GsTlsPublicKey and GsX509Certificate.

## 3.12 Support for Digital Envelopes

GemStone has added support for GsDigitalEnvelopes, which allow both encryption and signing in a single operation.

A GsDigitalEnvelope has the following security features:

- **Confidentiality** -- the message is encrypted using a randomly generated AES encryption session key and initialization vector. The session key is then encrypted with the provided public encryption key.
- **Integrity --** authenticated encryption guarantees the cipher text has not been altered. The digital signature guarantees the encrypted key, initialization vector and tag have not been altered.
- **Authentication** -- the receiver of the envelope is assured that the sender signed the envelope with the private key matching the public key used to successfully verify the signature.

NOTE: In order to guarantee authentication, the receiver must confirm that the public verification key actually belongs to the sender. Normally this confirmation is done by verifying that an X509 certificate containing the sender's public key has been signed by a reputable certificate authority.

GsDigitalEnvelope does NOT do this public key/certificate signature verification; it is up to the envelope’s recipient to ensure that the public key or X509 certificate used to verify the signature is trustworthy.

### Creating the GsDigitalEnvelope

Digital Envelopes are created using the following methods:

```
GsDigitalEnvelope class >> encryptMessage: messageBytes
   withPublicEncryptionKey: publicEncryptionKey
   cipherId: cipherOpCode
   withPrivateSigningKey: privateSigningKey
```

```
GsDigitalEnvelope class >> encryptMessage: messageBytes
   withPublicEncryptionKeys: arrayOfPublicEncryptionKey
   cipherId: cipherOpCode
   withPrivateSigningKey: privateSigningKey
```

With the following arguments:

*messageBytes*

The message to be encrypted may be any kind of String or ByteArray. The name of the class is stored in the envelope so when the envelope is opened, an instance of the same kind of object is returned. Internally, the string is converted into big-endian form, but this is transparent to the user. It is an error if the class of object that was encrypted into the envelope is not resolvable at the destination.

*publicEncryptionKey* and *privateSigningKey*

The message is encrypted using a public key, and signed using a private key. The envelope can only be "opened" (decrypted) by a recipient with the private key that matches the public encryption key, and the public signing key.

Providing multiple encryption keys allows a message to encrypted by multiple public keys into multiple envelopes in one operation. Each envelope can be opened by the single matching private key, and the public signing key.

The encryption and signing keys can be instances of GsTlsPublicKey or GsX509Certificate.

*publicEncryptionKey* or the elements in *arrayOfPublicEncryptionKey* must be RSA.

*privateSigningKey* may be any of the following:

| | | |
|---|---|---|
| DSA | RSA | Ed25519 |
| ECDSA | EC | Ed448 |

For signing algorithms that require a message digest algorithm, the SHA2-256 message digest is used. Signatures generated with RSA keys are padded using the RSA_PKCS1_PSS_PADDING scheme

*cipherOpCode*

The `cipherId:` argument accepts a numeric code specifying the AEAD (Authenticated Encryption with Additional Data) of the cipher; legal values are in the following table.

| opCode | Cipher | Mode | bits/Bytes |
|---|---|---|---|
| 4 | AES | OCB | 128/16 |
| 5 | AES | OCB | 192/24 |
| 6 | AES | OCB | 256/32 |
| 7 | AES | GCM | 128/16 |
| 8 | AES | GCM | 192/24 |
| 9 | AES | GCM | 256/32 |
| 10 | CHACHA20 | Poly1305 | 256/32 |

Ciphers corresponding to 6 and 10 are considered the most secure and are recommended for most applications.

## Using the GsDigitalEnvelope

Support has also been added to allow GsDigitalEnvelopes to be passivated into string form, for ease of transmission to the destination.

Once a GsDigitalEnvelope object is created, it can be converted to a passive object (String) for transmission to the recipient. Decrypting ("opening") the given envelope requires the private key matching the argument public key, and the public signing key matching the private key used to sign the envelope. The recipient activates the GsDigitalEnvelope object, and then opens it using the matching keys.

A detailed example is provided in class methods in GsDigitalEnvelope. The example follows the following form:

### On the originating server

```
"GsDigitalEnvelope created"
envelope := GsDigitalEnvelope
      encryptMessage: messageText
      withPublicKey: <gsTlsPubEncrKey>
      cipherId: 10
      withPrivateSigningKey: <gsTlsPrivSigKey>.
"envelope serialized to a string for sending"
stream := WriteStream on: String new.
PassiveObject passivate: envelope toStream: stream.
passiveString :=  stream contents.
```

### On the destination server

```
"GsDigitalEnvelope instance recreated from the serialized form"
envelope := (PassiveObject newWithContents: passiveString)
   activate.
"actual decryption"
decryptedMsg := envelope decryptWithPrivateKey: <gsTlsPrivEncrKey>
   withPublicVerificationKey: <gsTlsPubSigKey>.
```

Chapter

4

# Changes in the GCI Interfaces

## 4.1 GCI changes

### Added GCI methods

#### GciClassRemoveAllMethods_

A variant of `GciClassRemoveAllMethods`, which takes a single additional argument, the environmentId in which to remove methods.

#### GciX509Login

This method added to support X509-Secured logins:

```
(BoolType) GciX509Login(
    GciX509LoginArg *args
    );
```

A C++ class has been added:

```
GciX509LoginArg
```

For further details, see *GemStone/S 64 Bit X509-Secured GemStone System Administration Guide*.

### Linker symbols now weak on Linux

Support for GCI application on Ubuntu 18.04 required some changes in the way GemStone shared libraries are built. The shared libraries now contain weak symbols, and the linker flags for linking a GCI application have changed.

Otherwise, this change should have no impact on customer application.

# 4.2 Foreign Function Interface (FFI) related changes

## Pre-parsing of C header files

Previously, the FFI parsing of external C library headers could fail with unexpected token errors, depending on definitions within the header file and its include files. Now, before the FFI performs parsing, the native C preprocessor is automatically run on the header file.

### C Preprocessor must be available

On Linux and AIX, `cpp` must be available, and on Darwin, `clang` must be available, in order for FFI to parse C header files.

### Comment style requirements change

Note that the C preprocessor on AIX only supports ANSI-compliant C; C++ style comments using // are not accepted by the preprocessor.

### Comments no longer included in output

A side effect of this change is that comments in the header files are no longer preserved in the output from FFI processing. Inclusion of C comments had been somewhat unreliable since associating the declaration with the comment depended on the convention of the comment preceding the declaration and applying only to a single function, which was not always the case.

## GciLibrary instance methods no longer include comments

GciLibrary instance methods are generated using the FFI, and as a result of the preprocessing changes, these methods longer include the comment.

Refer to the *GemBuilder for C* manual, or the text in `$GEMSTONE/include/gci.hf`, for function details.

## CHeader specification of a library search path

The `CHeader class >> path:` method previously required a full path to the C header file that you intended to parse. Header files that are included by the specified file are also parsed, and these libraries were found by searching on specific system search path directories.

This did not allow you to use specific versions of include files other than the installed versions. For example, when parsing a header file that includes the OpenSSL headers, parsing would find the installed system versions of OpenSSL, rather than the intended development version.

Now, path specification is more flexible. You may pass only the name of the file to `path:`, in which case it searches on the system path for both that file and included files.

You can also specify the library search path for a given header file name or path, using the following methods:

`CHeader class >> path:` *filePathString* `searchPath:` *searchPathString*

`CHeader class >> path:` *filePathString* `searchPaths:` *anArrayOfPathStrings*

These methods use the specified search path or paths to find the specified file *filePathString*, and any files referenced by #include directives from that file. If the header files are not found, it will look in the system search paths; the given search paths are searched before the system include directories.

A new method, `CPreprocessor >> allSearchPaths`, returns the full set of search paths for an instance of CPreprocessor.

## Subclassing FFI classes

In earlier versions, FFI methods hard-coded some class names, making subclassing difficult. Species methods have been added to allow subclasses to override behavior. Added methods include:

```
CPreprocessor >>  cPreprocessorTokenSpecies
CPreprocessor >>  cDeclarationSpecies
CHeader >> cByteArraySpecies
CHeader >> cDeclarationSpecies
CHeader class >> cPreprocessorSpecies
CDeclaration >> cCalloutSpecies
CDeclaration >> cDeclarationSpecies
CDeclaration >> cPreprocessorSpecies
```

## Other added and removed methods

The following method have been added by the code changes.

```
CDeclaration >> byteSizeRounded
CPreprocessor >> insertSearchPath:
CPreprocessor >> searchForInclude:excluding:
CPreprocessorToken >> defineParameters:expansionTokens:
CPreprocessorToken >> expandArguments:
```

The following methods have been removed by code changes:

```
CPreprocessor >> concatenate:definition:arguments:
CPreprocessor >> createStringLiteralFrom:
CPreprocessor >> createStringLiteralsIn:from:
CPreprocessorToken >> defineParameters
CPreprocessorToken >> defineParameters:string:
CPreprocessorToken >> defineString
CPreprocessorToken >> isVarargsToken
CPreprocessor class >> clearDefaultDefinitions
CPreprocessor class >> defaultDefinitions
```

# 4.3 GCI and FFI Bugs Fixed

## SIGTERM handler not chained to installed handler in linked GCI

In the libgcilnk library, but not in the libgcirpc library, a SIGTERM handler is not chained to a previously installed handler. (#47862)

## Reported bytes sent/received incorrect

### Initial value incorrect

When setting up logging using GciDbgEstablishToFile(), the bytes sent and bytes received were not initialized. This results in, for example, GBS call tracing to report a large number of bytes sent and received by the first server call. (#47996)

### Printing incorrect on 32-bit Windows

With 32-bit clients on Windows, integer printing may incorrectly print 32-bit values as if they were 64 bit. (#47997)

## GciErrSType class >> new incorrect malloc size

The implementation was configured to malloc 2163 bytes, which did not allow for the round-up to 32-bit word boundaries, and should have been 2168. (#47304)

## Problems with GciTsFetchUtf8Bytes/GciTsFetchUtf8

The thread-safe GCI call `GciTsFetchUtf8Bytes` hung in the transport layer, and `GciTsFetchUtf8` returned an incorrect size result in *requiredSize. (#47694)

## CHeader macros do ## concatenation before argument substitution

When parsing a C header with an expression that performs both ## concatenation and argument substitution, the operations were done in the incorrect order. (#47192)

# 4.4 Updated Compile and Link Information

## Linux Compile and Link Information

### Complier version

Red Hat 6.x: gcc/g++ 4.4.7

Red Hat 7.x: gcc/g++ 4.8.5

Ubuntu Linux 16.04: gcc/g++ 5.4.0

Ubuntu Linux 18.04: gcc/g++ 7.3.0

SUSE Linux 12: gcc/g++ 4.8.5

### Debugger version

Red Hat 6.x: GNU gdb 7.2-92.el6

Red Hat 7.1: GNU gdb 8.2.1

Ubuntu Linux 16.04: GNU gdb 7.11.1

Ubuntu Linux 18.04: GNU gdb 8.1.0.20180409-git

SUSE Linux 12: GNU gdb 7.9.1

### Compiling a user action or GCI application

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

The following warn flags are recommended for compilation:

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

### Linking a user action library

```
g++ -shared -Wl,-Bdynamic,-hlibuserAct.so userCode.o
   $GEMSTONE/lib/gciualib.o -o libuserAct.so -m64
   -Wl,--no-as-needed -lpthread -Wl,--as-needed -lcrypt -ldl -lc
   -lm -lrt
```

### Linking a GCI application

```
g++ userCode.o $GEMSTONE/lib/gcirtlobj.o -Wl,-traditional
   -Wl,-z,lazy -m64 -Wl,--no-as-needed -lpthread -Wl,--as-needed
   -lcrypt -ldl -lc -lm -lrt -o userAppl
```

# Solaris on x86 Compile and Link Information

### Complier version

CC: Studio 12.5 Sun C++ 5.14 SunOS_i386 2016/05/31

### Debugger version

Sun DBX Debugger 7.7 SunOS_i386 2009/06/03

### Compiling a user action or GCI application

```
CC -xO4 -m64 -xarch=generic -Kpic -mt -D_REENTRANT
   -D_POSIX_PTHREAD_SEMANTICS -I$GEMSTONE/include
   -features=no%except -c userCode.c -o userCode.o
```

### Linking a user action library

```
CC -m64 -xarch=generic -G -Bsymbolic -h libuserAct.so -i userCode.o
   $GEMSTONE/lib/gciualib.o -o libuserAct.so -Bdynamic -lc
   -lsendfile -lpthread -ldl -lrt -lsocket -lnsl -lm -lpam -lCrun
   -z nodefs
```

### Linking a GCI application

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

# AIX Compile and Link Information

### Complier version

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

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

### Debugger version

dbx

### Compiling a user action or GCI application

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

Depending on your version of AIX, you need to include -DFLG_AIX_VERSION=61, -DFLG_AIX_VERSION=71, or -DFLG_AIX_VERSION=72.

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

### Linking a user action library

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

### Linking a GCI application

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

## DARWIN Compile and Link Information

### Complier version

Apple LLVM version 10.0.0 (clang-1000.10.44.4)

### Debugger version

lldb-1000.0.38.2, Swift-4.2

### Compiling a user action or GCI application

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

The following warn flags are recommended for compilation:

```
   -Wno-format -Wtrigraphs -Wcomment -Wsystem-headers -Wtrigraphs
   -Wno-aggregate-return -Wswitch -Wshadow -Wunused-value
   -Wunused-variable -Wunused-label -Wno-unused-function
   -Wchar-subscripts -Wno-conversion -Wmissing-braces -Wmultichar
   -Wparentheses -Wsign-compare -Wsign-promo -Wwrite-strings
   -Wreturn-type -Wno-nullability-completeness
   -Wno-expansion-to-defined
```

### Linking a user action library

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

### Linking a GCI application

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

## Windows Compile and Link Information

### Complier/Debugger version

Microsoft Visual Studio 2010 Version 10.0.30319.1 RTMRel

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

## Compiling a GCI application

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

## Linking a GCI application

```
link /LIBPATH:"VisualStudioInstallPath\VC\lib\amd64"
   /LIBPATH:"C:\Program Files (x86)\Microsoft
   SDKs\Windows\v7.0A\Lib\x64" -RELEASE
   /OPT:REF /INCREMENTAL:NO /MAP /nologo /MANIFEST
   /MANIFESTFILE:userAppl.exe.manifest
   /MANIFESTUAC:"level='asInvoker'" userCode.obj
   %GEMSTONE%\lib\gcirpc.lib ws2_32.lib netapi32.lib advapi32.lib
   comdlg32.lib user32.lib gdi32.lib kernel32.lib winspool.lib
   Secur32.lib /out:userAppl.exe
```

Chapter

# 5 Bug Fixes

The following bugs that were present in v3.4.3 are fixed in this version.

## Symbol Garbage Collection not working reliably

The process that implemented the garbage collection of unreferenced Symbols could commit interior nodes of the SymbolDictionary structure such that many Symbols were not actually garbage collected. (#47891)

## Statmonitor issues

### Statmonitor automatic start failed with -X

Starting up a statmonitor -X using configuration parameters such as STN_STATMONITOR_ARGS did not work, as these methods automatically include the stone or cache name as an argument. The -X option provides host-only monitoring and no stone or cache name is permitted as an argument. The -X option is now handled, but note that there are limitations in the log file names permitted within the -F argument in this mode. (#47885)

### Automatic start of statmonitor did not occur during pageaudit

Setting a value for STN_STATMONITOR_ARGS means that statmonitor should start automatically when the stone runs; this was not occurring during pageaudit. (#47674)

## Missing localhost entry in /etc/hosts file not handled

The Stone failed to start with a missing entry for localhost in the machine host file /etc/hosts. Now, localhost is referenced internally by ::1 or 127.0.0.1 rather than relying on host translation. (#47202)

## Host Memory Banner incorrect on the Mac

The amount of physical memory reported for the host of a GemStone process was not reported correctly on the Mac; the number provided was always 2048 MB. (#47905)

## GsExternalSession Issues

### Infinite recursion in GsExternalSession from GciError >> stack

`GciError >> stack` may have gone into infinite recursion if an external session gets an error during the stack operation. (#47656)

### GsExternalSession failed with gemNRS: string argument

GsExternalSession login parameters can accept an instance of GsNetworkResourceString, or a String containing NRS as an argument. The use of strings for the `gemNRS:` parameter was broken. (#47868)

### Logout problems in certain configurations

The tests to determine if Gem was correctly logged out did not take into account the possible existence of a stone with the same name running on the Gem host but not used to login the External session; nor a Gem running on a remote node. While the logout proceeded correctly, the return would not complete and the logout would appear to hang. (#47941, #47663)

## Hot Standby issues

### Logreceiver race condition could cause tranlog read failure

There is a race condition in logreceiver logic that can cause the slave stone to attempt to read from a newly transferred tranlog before any data has been permanently flushed to disk. This will generate errors claiming that the file is not a GemStone transaction log file. (#47844)

### Logsender added padding to tranlog records when compressing

During compression, the logsender added padding to the tranlog records. This was unnecessary; and the cumulative padding could cause buffer overrun. (#47916)

## NonTranloggedGlobals can cause tranlog restore to fail

Under certain sequences of operations, when using NonTranloggedGlobals and when the oopHighWaterMark is increasing, the oopHighWaterMark was not updated correctly. This did not cause the restoreFromLogs to fail, however, the stone could not subsequently start up. (#47679)

## GsSysLog writeErrorMessage fails with ArgumentError

Execution of the method `GsSysLog >> writeErrorMessage:` failed with an Invalid priority ArgumentError. (#47878)

## Incorrect error from before: and after: on out of range

When argument to `SequentialCollection >> before:` was the first element, or the argument to `after:` was the last element, the error incorrectly stated that the argument was not found in the collection. (#48072)

## PositionableStreamPortable>>match: incorrect with legacy streams

Applications may install either Portable or Legacy stream hierarchies, which differ in the indexing of the start at 0 or 1. With legacy streams installed, which is the default for applications that originated earlier then v2.0, explicitly calling the `PositionableStreamPortable>>match:` would error. (#47285)

## instVarAt:put: did not work for a class instance variables

Using `instVarAt:put:` to assign a class instance variables value reported an error. (#47430)

## Issues related to Upgrade

### After upgrade, GcGems not running

During the Stone restart during upgrade, the Reclaim and Admin GcGem/s were started, but could not login due to a version mismatch, and no further startup attempts were made after upgrade. This left the upgraded system without the GcGems running. (#47884)

### Upgrade failed with "Attempt to modify invariant object"

The upgrade process attempted to add a dynamic instance variable #name to existing ObjectSecurityPolicies, for later use in displaying descriptive information about the security policy.

### Upgrade's removeallmethods incorrectly removed special-case methods

The upgrade process executes removeallmethods to ensure old methods that have been removed in the new version are removed. This execution of `removeallmethods` should only apply to persistent methods in environment 0, the standard GemStone Smalltalk environment. It was incorrectly removing methods in environments above 0, and removing session methods. Now, upgradeImage will not remove methods that have been installed in other environments. (#47821), or transient session methods (#46802)

### Upgrade logging for PositionableStream handling

Upgrade logging output now includes more detailed information on the handling of the PositionableStreamLegacy vs. PositionableStreamPortable hierarchies.

## NRS with gemnetobject did not handle spaces in pathnames

When using #dir: or #log: directives in the NRS that passed a gemnetobject into GemStone login parameters, the use of a directory name with spaces resulted in errors. Directory and file names with spaces are now supported. (#40211)

## enableUnixAuthenticationWithAlias: failed in Unicode Mode

In a repository that has enabled Unicode Comparison Mode, the method `UserProfile >> enableUnixAuthenticationWithAlias:` fails with a primitive failure. (#47251)

## Current user SymbolList used instead of session SymbolList

As described in the *Programming Guide*, the transient and persistent SymbolLists have different functions. Several methods referred to the current User's SymbolList, where they should refer to the session SymbolList. (#48050) The following methods have been updated:

```
GsDigitalEnvelope >> decryptWithPrivateKey:
   withPublicVerificationKey:
GsNMethod >> _literalValToAssocations:
SortedCollection >> resortAll:
SymbolDictionary >> _listClassesIn:matching:
```

## Unnecessary warnings about log birth time

The Stone log may have included warnings of the following pattern, which were unnecessarily alarming. (#47686)

```
Warning, found a birth time mismatch, log to Repository.
log birth time = 1207767503, Repository birth time = 1530037449
```

## GsPackagePolicy test for session methods incomplete

The test to determine if a SessionMethod needed to be created was imprecise, which resulted in SessionMethods being created unnecessarily. (#47887)

## Optimized code inconsistently handled step points and senders

When method code includes statements such as `true ifFalse:` *aBlock*, the statement can be optimized by excluding from the compiled executable method. The tools in the system that locates method sends and step points depends on the complied code. Previously, the optimization-out of such blocks was inconsistent, and the senders and step point information was correspondingly inconsistent. (#47985)

If you wish code to remain in the method and available to sendersOf and similar operations, do not use the `true ifFalse:` pattern.

## Spurious errors on inter-gem signalling with Gem terminating

When a Gem is in the process of shutdown, inter-gem signalling could result in an unnecessary error being signalled. (#48016)