

---



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

## **Release Notes**

**Version 3.6**

November 2020



## **INTELLECTUAL PROPERTY OWNERSHIP**

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

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

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

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

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

## **COPYRIGHTS**

This software product, its documentation, and its user interface © 1986-2020 GemTalk Systems LLC. All rights reserved by GemTalk Systems.

## **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**, **POWER7**, **POWER8**, **POWER9** 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 3.6 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 3.6.

## Terminology Conventions

The term “GemStone” is used to refer to the server products GemStone/S 64 Bit and GemStone/S, and the GemStone family of products; the GemStone Smalltalk programming language; and may also be used to refer to the company, now GemTalk Systems LLC, previously GemStone Systems, Inc. and a division of VMware, Inc.

## Technical Support

### Support Website

#### **[gemtalksystems.com](http://gemtalksystems.com)**

GemTalk’s website provides a variety of resources to help you use GemTalk products:

- ▶ **Documentation** for the current and for previous released versions of all GemTalk products, in PDF form.
- ▶ **Product download** for the current and selected recent versions of GemTalk software.
- ▶ **Bugnotes**, identifying performance issues or error conditions that you may encounter when using a GemTalk product.

- ▶ **Supplemental Documentation** and **TechTips**, providing information and instructions that are not in the regular documentation.
- ▶ **Compatibility matrices**, listing supported platforms for GemTalk product versions.

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

## Help Requests

GemTalk Technical Support is limited to customers with current support contracts. Requests for technical assistance may be submitted online (including by email), or by telephone. We recommend you use telephone contact only for urgent requests that require immediate evaluation, such as a production system down. The support website is the preferred way to contact Technical Support.

**Website:** [techsupport.gemtalksystems.com](http://techsupport.gemtalksystems.com)

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

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

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

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

Technical Support is available from 8am to 5pm Pacific Time, Monday through Friday, excluding GemTalk holidays.

## 24x7 Emergency Technical Support

GemTalk offers, at an additional charge, 24x7 emergency technical support. This support entitles customers to contact us 24 hours a day, 7 days a week, 365 days a year, for issues impacting a production system. For more details, contact GemTalk Support Renewals.

## Training and Consulting

GemTalk Professional Services provide consulting to help you succeed with GemStone products. Training for GemStone/S is available at your location, and training courses are offered periodically at our offices in Beaverton, Oregon. Contact GemTalk Professional Services for more details or to obtain consulting services.

---



# Table of Contents

---

## *Chapter 1. Release Notes for 3.6*

Overview . . . . .	11
New keyfiles required . . . . .	11
Supported Platforms. . . . .	12
Platforms for Version 3.6 . . . . .	12
GemBuilder for Smalltalk (GBS) Versions. . . . .	12
VSD Version. . . . .	13
GemBuilder for Java (GBJ) Version. . . . .	13
GemConnect Version . . . . .	13
Upgrade . . . . .	13
Upgrading Seaside . . . . .	13
Documentation Changes . . . . .	13
1.1 Library and Distribution changes . . . . .	14
Updated library versions . . . . .	14
\$GEMSTONE/upgrade packaging. . . . .	14
Obsolete scripts no longer distributed . . . . .	14
1.2 Optimizations . . . . .	14
AIO Page Server and Free Frame Page Servers no longer separate processes. . . . .	14
NetLDI is now multithreaded. . . . .	15
Default Repository size smaller than keyfile maximum . . . . .	15
Changes in transaction logs recorded in milliseconds . . . . .	15
Transparent huge pages on linux for Gem temporary object memory . . . . .	15
On restore, data pages automatically compacted . . . . .	15
FindFreeFrame handling . . . . .	15

## Chapter 2. Changes in Administration

2.1 Encrypted Extents and Transaction Logs . . . . .	17
Encrypted Extents . . . . .	17
Internal handling for other sessions . . . . .	18
Transaction Logs . . . . .	18
Restart and Recovery . . . . .	18
Programmatic Backups . . . . .	19
Extent Copy Backup . . . . .	19
Solo logins . . . . .	19
Generating keys . . . . .	19
Example Setting up Encrypted Extents . . . . .	20
Encrypted Extents with Backup and Restore . . . . .	20
Restoring Backups . . . . .	21
Restoring transaction logs with no changes in encryption keys . . . . .	21
Restoring transaction logs with changes in encryption keys . . . . .	21
Page Audit . . . . .	23
Hot Standby . . . . .	23
printlogs/searchlogs . . . . .	23
Programmatic Information about encryption . . . . .	24
Modifying Encrypted Files . . . . .	24
Examples of modifying encrypted files . . . . .	24
To encrypt an extent . . . . .	25
To start a stone on an encrypted extent . . . . .	25
To decrypt an extent or extent snapshot backup . . . . .	25
To decrypt an extent or extent snapshot backup . . . . .	25
To change the key of an encrypted extent or extent snapshot backup . . . . .	25
To change the key of an encrypted transaction log . . . . .	26
To change the encryption key of an encrypted fullbackup . . . . .	26
To decrypt a fullbackup . . . . .	27
Encrypting and Decrypting GsTls* keys . . . . .	27
2.2 Changes in Users and ObjectSecurityPolicies . . . . .	28
CodeLibrarianUser no longer needed . . . . .	28
DataCurator no longer has permission to modify Globals . . . . .	28
SystemUser no longer subject to privilege checks . . . . .	28
SystemUser now exposes sub-dictionaries . . . . .	28
Changes in handling of transient symbolList . . . . .	28
Whitelist for performOnServer: . . . . .	29
Example . . . . .	29
GsSysLog . . . . .	30
2.3 Changes in Backup, Restore and Hot Standby . . . . .	30
Tranlog version numbers following restore . . . . .	30
Hot Standby Failover Changes . . . . .	30
Added methods . . . . .	31
New Planned Failover Process in Detail . . . . .	31
2.4 Other Changes related to administrative operations . . . . .	33
GsObjectInventory changes . . . . .	33

Improved thread handling. . . . .	33
Removed methods profileObjectsInBm: and profileObjectsInFile: . . . . .	33
Deprecated findAllReferencePaths feature removed. . . . .	33
System currentSession* includes logsender, logreceiver. . . . .	33
2.5 Changes in Errors . . . . .	34
AlmostOutOfMemoryError . . . . .	34
AlmostOutOfStackError . . . . .	34
ExitClientError . . . . .	34
UncontinuableError . . . . .	34
Other added Exceptions . . . . .	34
Removed Exceptions . . . . .	34
2.6 Changes in Utilities . . . . .	35
copydbf . . . . .	35
Secure extent support options added. . . . .	35
Previous, non-working -s has new meaning. . . . .	35
copydbf appends zip/secure extensions as needed. . . . .	36
Changes in file naming with copydbf to directory target . . . . .	36
copydbf semantics change for zip status. . . . .	36
Required to use -u to copydbf a compressed file into raw partition . . . . .	36
copydbf output to stderr vs. stdout. . . . .	36
copydbf no longer accepts NRS for remote access, -n removed . . . . .	36
Record-level compression argument -c removed . . . . .	37
Testing option -d added . . . . .	37
netldebug utility added . . . . .	37
printlogs and searchlogs. . . . .	37
pstack . . . . .	37
startlogsender, startlogreceiver . . . . .	37
startlogreceiver -D argument for additional logging . . . . .	37
startlogsender -u option removed . . . . .	38
startlogsender option to specify tranlog prefix . . . . .	38
startnetldi . . . . .	38
Added option to configure threads. . . . .	38
Resolving hostnames using reverse DNS lookup . . . . .	38
startstone . . . . .	38
Resolving hostnames using reverse DNS lookup . . . . .	38
Underscores removed from prompts for username and password . . . . .	38
startstone and pageaudit secure database options added . . . . .	39
updatesecuredbf utility added . . . . .	39
2.7 Topaz changes . . . . .	39
Prompt change when input nested files . . . . .	39
Added commands . . . . .	39
TFILE. . . . .	39
DISPLAY/OMIT ALLOOPS. . . . .	40
DISPLAY/OMIT DECIMALBYTES . . . . .	40
SET SINGLECOLUMN ON/OFF. . . . .	40
Changes in commands. . . . .	40
STRINGS / STRINGSIC searches class comments . . . . .	40

SET CLASS to nil . . . . .	40
ALLSTACKS limit removed . . . . .	40
QUIT / EXIT with file input . . . . .	40
DEFINE no longer updates ErrorCount. . . . .	40
2.8 Configuration Parameter Changes . . . . .	41
Added Configuration Parameters . . . . .	41
GemDebuggerActive . . . . .	41
StnReverseDns . . . . .	41
Changed Configuration Parameter . . . . .	41
STN_NUM_LOCAL_AIO_SERVERS . . . . .	41
2.9 Changes in Statmonitor and Cache Statistics . . . . .	42
Statmonitor command line details. . . . .	42
Default process cache names. . . . .	42
Topaz linked sessions distinguished in statmonitor data. . . . .	42
FailedCommitCount includes commits that succeeded on retry. . . . .	42
Changed Cache Statistics . . . . .	43
Added Cache Statistic. . . . .	43
Removed Cache Statistics . . . . .	43

### *Chapter 3. Changes in GemStone Smalltalk*

3.1 Changes in Classes, compilation, and code management . . . . .	45
Instance variable name length limit. . . . .	45
Literal Arrays may now embed literal ByteArrays . . . . .	45
Block not optimized no longer reports CompilerWarning . . . . .	45
Change in handing of transient symbol list . . . . .	46
Changes in Class Categories . . . . .	46
Private method to rename an instance variable . . . . .	46
3.2 Changes in Numeric/Time classes . . . . .	47
Added Specials . . . . .	47
SmallDate. . . . .	47
SmallDateAndTime . . . . .	47
SmallScaledDecimal. . . . .	47
SmallTime . . . . .	47
DateAndTime changes . . . . .	48
Special care when adding methods to kernel classes DateAndTime, Date, Time, and ScaledDecimal. . . . .	48
Other added methods. . . . .	49
3.3 ExternalSession Changes. . . . .	49
GsTsExternalSession changes . . . . .	49
Methods added to both. . . . .	50
3.4 Other Changes and Enhancements . . . . .	51
Behavior >> isSpecial should not be used; replaced by Behavior >> areInstancesSpecial . . . . .	51
Added commit failure types . . . . .	51
Exiting the Client from Smalltalk . . . . .	51



ClassOrganizer searches class comments, return arrays changed . . . . .	51
3.5 Deprecated and removed classes and methods. . . . .	52
Removed Classes . . . . .	52
Find All Reference Path removed methods . . . . .	52
Other removed methods: . . . . .	53
Private removed methods . . . . .	53

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

4.1 GCI changes . . . . .	55
Added Functions and Macros. . . . .	55
GciCheckSigTerm. . . . .	55
GciFetchDateAndTime. . . . .	55
GciFetchDateAndTimePosix . . . . .	55
GciNewDateAndTime . . . . .	55
GciNewDateAndTimePosix . . . . .	56
GCI_OOP_IS_SmallDate. . . . .	56
GCI_OOP_IS_SmallDateAndTime . . . . .	56
GCI_OOP_IS_SMALL_DOUBLE . . . . .	56
GCI_OOP_IS_SmallScaledDecimal . . . . .	56
GCI_OOP_IS_SmallTime . . . . .	56
4.2 GCI thread-safe changes. . . . .	56
Added Functions . . . . .	56
GciTsNbLogin. . . . .	56
GciTsNbLoginFinished. . . . .	57
Removed Functions . . . . .	57
4.3 Foreign Function Interface (FFI) related changes. . . . .	57
Support for Callout symbol version . . . . .	57
Race condition in setting FFI errno . . . . .	57
Added methods. . . . .	58
4.4 UserAction changes . . . . .	58
Handling Errors in user actions. . . . .	58
Invoking a client useraction from non-blocking GCI call could freeze Gem. . . . .	58
4.5 Updated Compile and Link Information . . . . .	59
Linux Compile and Link Information . . . . .	59
Solaris on x86 Compile and Link Information . . . . .	60
AIX Compile and Link Information . . . . .	60
DARWIN Compile and Link Information . . . . .	61
Windows Compile and Link Information . . . . .	62

## ***Chapter 5. Bug Fixes***

Slow stone startup with large possibleDead. . . . .	63
Socket blocking issues . . . . .	63
Session waiting in GsSocket >> accept does not respond to stopSession. . . . .	63

GsSocket read or accept may block after Socket closed . . . . .	63
readWillNotBlockWithin: and writeWillNotBlockWithin: result could be incorrect. . . . .	63
Hot standby bugs . . . . .	63
logsender did not respect not-default STN_TRAN_LOG_PREFIX. . . . .	63
logsender login failures caused excessive error messages . . . . .	64
Repository>>continuousRestoreFromArchiveLogs: did not work on extent copy . . . . .	64
startlogreceiver did not expand environment variables in -T argument .	64
CHep memory leaks in failed X509 logins . . . . .	64
Restore from tranlogs may not replay clustering . . . . .	64
Repository scan may fault in dead objects . . . . .	64
DBF_PRE_GROW did not accept list with empty elements . . . . .	64
Stone may not be responsive while disposing large CR backlog . . . . .	64
Out of Memory in code_gen space with many classes . . . . .	65
Error reporting for Stone not found. . . . .	65
GsHostProcess executable-not-found error not helpful. . . . .	65
Statmonitor issues. . . . .	65
Deleting a statmonitor file in use may corrupt the new file. . . . .	65
In auto-restart mode, file name may be reused . . . . .	65
Topaz issues . . . . .	65
Possible to redefine the ErrorCount predefined variable . . . . .	65
Display bytes with multi-byte characters did not linebreak correctly . . .	65
Time and Date related issues . . . . .	66
TimeZone fromPath: attempted to look for zoneinfo on client. . . . .	66
Duration printing of subseconds did not round up to second . . . . .	66
DateAndTime fromString: without timezone was incorrect . . . . .	66
GciTsLibrary version: did not handle argument flexibly . . . . .	66
Removed reference to JISCharacter in RangeEqualityIndex>>isBasicClass: . . .	66
GemStoneParameters did not handle invariant password string . . . . .	66

# Release Notes for 3.6

---

## Overview

GemStone/S 64 Bit™ 3.6 is a new version of the GemStone/S 64 Bit object server. Version 3.6 includes a number of important new features, such as the ability to encrypt the on-disk representation of extents and other dbf files.

These Release Notes include changes between the previous version of GemStone/S 64 Bit, v3.5.4, and v3.6. If you are upgrading from a version prior to 3.5.3, review the release notes for each intermediate release to see the full set of changes.

For details about installing GemStone/S 64 Bit 3.6 or upgrading from earlier versions of GemStone/S 64 Bit, see the *GemStone/S 64 Bit Installation Guide* for v3.6 for your platform.

### Note for Beta users

If you have been testing with a beta version of v3.6, and if you are using encrypted extents, these beta-encrypted extents are not compatible with the release executables; there have been root page changes that make the encrypted extents incompatible. You must re-encrypt your beta-test extent dbfs, before these can be used with the 3.6 release executables.

## New keyfiles required

The keyfiles for v3.5.x and earlier cannot be used with v3.6; new keyfiles are required for this release. To obtain a new keyfile for GemStone/S v3.6, write to [keyfiles@gemtalksystems.com](mailto: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.

## Supported Platforms

### Platforms for Version 3.6

GemStone/S 64 Bit version 3.6 is supported on the following platforms:

- ▶ Red Hat Enterprise Linux Server and CentOS Linux 7.8 and 8.1; and Ubuntu 16.04, 18.04, and 20.04  
GemStone performs testing on a mixture of Red Hat and CentOS servers; both are fully certified platforms. Any reference to Red Hat applies to both distributions.
- ▶ Solaris 10 and 11.4 on x86
- ▶ AIX 7.1 and 7.2
- ▶ OS X 10.14.6 (Mojave) with Darwin 18.7.0 kernel, and OSX 10.15.6 (Catalina) with Darwin 19.6.0 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 v3.6 Installation Guide* for that platform.

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

### GemBuilder for Smalltalk (GBS) Versions

GemStone/S 64 Bit version 3.6 requires GBS version 8.5 or later for VisualWorks Smalltalk, or version 5.4.6 or later for VA Smalltalk.

The following versions of GBS are supported with GemStone/S 64 Bit version 3.6:

#### GBS version 8.5

VisualWorks 8.3.2 32-bit and 64-bit	VisualWorks 8.2.1 32-bit and 64-bit
<ul style="list-style-type: none"> <li>▶ Windows 10</li> <li>▶ RedHat ES 7.8 and 8.1;</li> <li>▶ Ubuntu 18.04 and 20.04</li> </ul>	<ul style="list-style-type: none"> <li>▶ Windows 10</li> </ul>

GBS 8.5 is expected to work with VisualWorks 9.0, but is a "preview" platform and not fully supported.

#### GBS version 5.4.6

VA Smalltalk 9.2.2	VA Smalltalk 8.6.3
<ul style="list-style-type: none"> <li>▶ Windows 10</li> </ul>	<ul style="list-style-type: none"> <li>▶ Windows 10</li> <li>▶ Windows 7</li> </ul>

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 v3.6 distribution includes VSD version 5.5.1. The previous version of GemStone/S 64 Bit, v3.5.4, included VSD v5.5.

VSD version 5.5.1 includes a number of new features and bug fixes. For details on the changes, see the [Release Notes for VSD v5.5.1](#).

With GemStone/S 64 Bit v3.6, **statmonitor** now writes additional information to the statmonitor file: the command line used to invoke statmonitor. As a result, statmonitor files from v3.6 cannot be read by versions of VSD earlier than v5.5. VSD 5.5 can read statmonitor files generated in older versions of GemStone/S 64, 32-bit GemStone, and GBS, as well as those generated by GemStone/S 64 Bit v3.6.

VSD 5.5.1 is included with the GemStone distribution, and can also be downloaded as a separate product from <https://gemtalksystems.com/vsd/>

## GemBuilder for Java (GBJ) Version

The most recent version of GBJ, v3.1.3, can be used with GemStone/S 64 Bit v3.6.

The GBJ shared library that is distributed with the server, libgbjgci313\*, includes basic support for the new special data types in 3.6 (SmallTime, SmallDate, SmallDateAndTime, SmallScaledDecimal). Further support will be added in the next GBJ release.

## GemConnect Version

The most recent version of GemConnect, v2.4, can be used with GemStone/S 64 Bit v3.6.

## Upgrade

Upgrade is supported from all 3.3.x, 3.4.x, and 3.5.x versions. To upgrade from earlier versions, upgrade to a supported upgrade origin version, and then upgrade to 3.6.

Note that **upgradeImage** no longer overwrites the file `$upgradeLogDir/upgradeImage.sh`. If this file exists, **upgradeImage** creates `$upgradeLogDir/upgradeImage_1.sh`, and if this also exists, `$upgradeLogDir/upgradeImage_2.sh`, and so on.

### Upgrading Seaside

The **upgradeseasideimage** script now supports the **-W** argument, which enables the topaz command **topazwaitfordebug**. When **-Wis** specified, a remote gem can attach to debug the upgrade process via the topaz **debuggem** command.

## Documentation Changes

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 chapter has been added to the *System Administration Guide* that is specific to using encrypted extents.

- ▶ Materials on Date and Time classes, including the new specials added in v3.6, is added to the *Programming Guide* numerics chapter.
- ▶ The chapters on SymbolLists and ExternalSessions in the *Programming Guide*, and on Hot Standbys in the *System Administration Guide* have been extensively revised.
- ▶ Using **debuggem** and **topazwaitfordebug** to attach to and debug from a separate Gem process is now described in the *Topaz User's Guide*.

The *GemStone/S 64 Bit X509-Secured GemStone System Administration Guide* has not been updated for this release.

## 1.1 Library and Distribution changes

### Updated library versions

The version of OpenLDAP has been updated to 2.4.55.

The ZoneInfo (TimeZone database) has been updated from 2019a to 2020a.

### **\$GEMSTONE/upgrade** packaging

The way GemStone image code is packaged in the distribution for upgrade has been changed, as part of the development towards tonel-based source code management.

This change should be transparent to the user, but the contents of **\$GEMSTONE/upgrade** will appear very different.

A number of Classes whose names begin with Upgrade, are present in the Globals SymbolDictionary. These classes implement methods used during the upgrade process.

### Obsolete scripts no longer distributed

The upgradeComments, postconv, and convprep22 scripts are obsolete, and no longer included in the distribution.

## 1.2 Optimizations

### AIO Page Server and Free Frame Page Servers no longer separate processes

The AIO Page Server and Free Frame Page Servers are now threads within the stone, rather than separate processes.

The way these are configured and tuned is unchanged; the configuration parameters have not been renamed, to avoid impact during upgrade.

The individual log files previously created by these processes are no longer written; these were seldom useful.

Page server cache statistics, as recorded by statmonitor, reports statistics for these page server threads with only minor changes, as AioThreadN and FFThreadN.

The supporting scripts, runaiopgsvr and runffpgsvr, have been removed.

## NetLDI is now multithreaded

The NetLDI is now multithreaded. The number of threads defaults to 10; it can be configured using the `startnetldi -t` option. With X509-Secured GemStone, in secure mode, or if the NetLDI is running as root, the NetLDI always runs single-threaded.

## Default Repository size smaller than keyfile maximum

For limited GemStone licences, one of the limits is the repository size. In previous releases, the default maximum size for a repository was the limit set in a keyfile; however, if the repository ran out of space (for example, a temporary increase due to a commit record backlog), there was no ability to recover, other than an emergency temporary keyfile from GemTalk Technical Support.

Now, when a license keyfile has a limit on the repository size, the default maximum size of the repository is 80% of the keyfile limit. This default is used if no size is explicitly set. If the repository reaches the default limit, it will shut down, but by manually configuring the size to a larger value that is still lower than the licence limit, the system can be restarted easily and the problem resolved.

## Changes in transaction logs recorded in milliseconds

The timestamps on records in the transaction logs were previously in seconds, and are now in milliseconds.

## Transparent huge pages on linux for Gem temporary object memory

Starting with version 3.5.1, the shared page cache is able to take advantage of transparent huge pages (2MB) pages, provided that this is enabled in the Linux kernel (see the Installation Guide for details). No special configuration in GemStone is needed to use transparent huge pages.

With 3.6, if `GEM_TEMPOBJ_CACHE_SIZE` is configured to more than 400MB, Gem temporary object memory can also use transparent huge pages. No specific action is required, other than enabling transparent huge pages in the linux kernel.

## On restore, data pages automatically compacted

When restoring a backup, formerly, the repository could contain many partially empty data pages. Now, data pages are automatically compacted during restore.

Clustering is respected, and performance appears to be the same or better than without compaction.

## FindFreeFrame handling

When a Gem requires new frame, but the shared page cache does not have frames on the free list, it scans the cache looking for a frame that can be made available. It is possible that no frames can be made available. Now, if it does not find a free frame after 60 seconds, the Gem will get a fatal cache error.





# 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:

Encrypted Extents and Transaction Logs . . . . .	17
Changes in Users and ObjectSecurityPolicies . . . . .	28
Changes in Backup, Restore and Hot Standby . . . . .	30
Other Changes related to administrative operations . . . . .	33
Changes in Errors . . . . .	34
Changes in Utilities . . . . .	35
Topaz changes . . . . .	39
Configuration Parameter Changes . . . . .	41
Changes in Statmonitor and Cache Statistics . . . . .	42

## 2.1 Encrypted Extents and Transaction Logs

GemStone provides user and object level security to protect your application data from unauthorized access via applications or Smalltalk code execution. However, the repository extents contain all repository objects, and the tranlogs record recent changes; these may include readable data, and are normally protected by disk permissions from unauthorized access.

With v3.6, you may also configure your system to keep the extents and transaction logs encrypted on disk at all times.

### Encrypted Extents

With encrypted extent files, the on-disk extents are encrypted at all times. Transaction logs written by a Stone using encrypted extents are always written in encrypted form, using the same keypair as the Stone.

To use encrypted extents, you must manually encrypt the repository extent files prior to starting the Stone, using a public key or X509 certificate. The **startstone** command takes additional arguments, specifying the private key. If the repository has more than one extent, all extents must be encrypted using the same key. Only one keypair can be associated with encrypted extents.

The encryption step is done using a public key; to start the stone on these extents, the private key corresponding to the public key (and the passphrase for the private key, if configured) are required. Encryption is done using AES-XTS symmetric encryption with a key length of 128 or 256 bits.

In an encrypted extent, all data in the extent is encrypted, except for a small number of records with metadata information. This unencrypted information also allows **copydbf -i** to report information without needing the key.

Once the Stone has started, the private key is not needed again until the next time the Stone is restarted. Pages from the extents are decrypted before reading into the cache, and encrypted on write back to disk. Since the Stone may need restart at any time if an unexpected outage occurs, the private key must be kept readily available.

## Internal handling for other sessions

The Stone uses the private key argument provided by **startstone** to decrypt the session key stored in each of the extents. Each extent has its own unique session key. This is stored in private stone memory, and so is not accessible to other sessions on the shared page cache. Processes that read or write to the extents need access to this key.

For gems, which may be on remote hosts over unsecured networks, a random key is generated and used to encrypt the keys, which are provided in shared memory for local gems, or over an encrypted SSL socket for remote gems.

The first gem to attach to a remote shared page cache stores the keys in the encrypted extent key table in the remote cache, under spin lock control.

## Transaction Logs

Transaction logs written by a Stone using encrypted extents are written in encrypted form, using the same keypair as the extents. When the Stone is started, if recovery is needed (that is, on unclean shutdown), it will be able to recover from transaction logs since they were encrypted using the same key as was used to start the Stone. Special care is needed when updating the encryption key of the Stone on an extent that has not been cleanly shutdown.

If the encryption key of the extents have changed, on startup after a clean shutdown, the Stone will start a new transaction log that is automatically encrypted with the new keys. In this case, you must take care to keep the older keypairs, or update the encryption keys for the older tranlogs, if you want to be able to restore these older tranlogs into an older backup.

## Restart and Recovery

If the stone unexpectedly shuts down, the private key is needed to restart the stone. Automatic recovery from transaction logs requires that all required transaction logs for recovery be encrypted using the same key as the stone. If necessary, the transaction logs may have their keys updated using **updatesecuredbf**.

## Programmatic Backups

Making a programmatic backup of a repository running with encrypted extents does **not** automatically create an encrypted backup. You must use the backup creation methods that specifically create encrypted backups.

- ▶ the standard methods `Repository >> fullBackupTo:...` make unencrypted backups, regardless of whether the running Stone's extents are encrypted or not.
- ▶ the secure backup methods `Repository >> secureFullBackupTo:...` make encrypted or unencrypted/signed backups, using the key/s provided to that method, regardless of whether the running Stone's extents are encrypted or not.

## Extent Copy Backup

An extent copy backup of a repository with encrypted extents is, of course, encrypted.

Ensure that the private key remains available in order to decrypt or restore. If the certs and keys need to be updated, you can use **updatesecuredbf** on the extent copy backups, but you will need to use the **-O** option, since the extents were not cleanly shutdown.

## Solo logins

Solo login sessions, sessions that use an extent file directly to login, without starting a Stone process, are disallowed with encrypted extents.

## Generating keys

The following process can be used to generate your own public/private key pairs. The example uses `$GEMSTONE/bin/openssl`, but you may use any openssl executable.

1. Generate the private key:

```
unix> openssl genrsa -out privateKeyName 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
```

2. Extract the public key from the private key:

```
unix> openssl rsa -in privateKeyName -pubout -out publicKeyName
writing RSA key
```

3. Optional but strongly recommended: protect the private key with a passphrase.

```
unix> openssl rsa -aes256 -in privateKeyName -out privatePPKeyName
writing RSA key
Enter PEM pass phrase: <enter a passphrase>
Verifying - Enter PEM pass phrase: <enter passphrase again>
Then delete the unprotected private key:
unix> rm privateKeyName
```

## Example Setting up Encrypted Extents

The following example shows the steps to encrypt a two-extent repository with a key from the GemStone distribution. Note that this does not provide security; you must generate your own private key meeting your own security requirements.

**Step 1.** Stop the Stone cleanly. While it is possible to encrypt an extent that is not cleanly shutdown, you would also need to encrypt the tranlogs needed for recovery.

```
unix> stopstone stoneName administrativeUser password
```

**Step 2.** Make encrypted copies of your extents using copydbf.

```
unix> copydbf $GEMSTONE/data/extent0.dbf
           $GEMSTONE/data/secextent0.sdbf -e server_1_servercert.pem
           -s 256 -K $GEMSTONE/examples/openssl/certs
```

```
unix> copydbf $GEMSTONE/data/extent1.dbf
           $GEMSTONE/data/secextent1.sdbf -e server_1_servercert.pem
           -s 256 -K $GEMSTONE/examples/openssl/certs
```

**Step 3.** Edit the configuration file used by the Stone to specify the new extent names.

```
DBF_EXTENT_NAMES = $GEMSTONE/data/secextent0.sdbf,
                  $GEMSTONE/data/secextent1.sdbf;
```

**Step 4.** Execute startstone using the private key.

```
unix> startstone -D server_1_serverkey.pem
           -J $GEMSTONE/allcerts/server_1_server_passwd.txt
           -K $GEMSTONE/examples/openssl/certs
           -K $GEMSTONE/examples/openssl/private gs64stone
```

On startup, the Stone will immediately start a new transaction log that is encrypted with the key `server_1_servercert.pem`.

Using the stone, and stopping the stone, now proceed as with any GemStone installation.

## Encrypted Extents with Backup and Restore

How to make and restore the various kinds of backups, including encrypted backups, is described in the *System Administration Guide*, Chapter 11.

Making a backup in a repository configured with secure extents is no different than making a backup with ordinary extents. From a repository configured with encrypted extents, you may make:

- ▶ Online or offline extent snapshot backups, which remain encrypted with the same keys as the working system extents.
- ▶ Ordinary programmatic backups, which are not encrypted. An ordinary backup from a repository using encrypted extents is not automatically encrypted.
- ▶ Secure programmatic backups, which are signed and may either be encrypted or not encrypted. Encryption keys are provided as argument to the secure backup methods, so the backup may be encrypted using the same keypair as the working system's extents, or an entirely different keypair.

While restoring a programmatic fullbackup or an extent snapshot back into a repository with extent encryption is not substantially different from restoring into a repository with regular unencrypted extents, if the encryption keypairs that are used to encrypt the extents has changed, this will impact restoring transaction logs.

## Restoring Backups

When restoring a programmatic fullbackup to a system that is using encrypted extents, you will need to start with a clean, **encrypted** extent. This requires an additional step to use `copydbf` to encrypt a clean, empty extent.

Alternately, you can restore into an unencrypted extent, and then shut down the restored stone, encrypt the extent files, and restart.

Of course, all **startstone** invocations will require the appropriate key for the encrypted extents of the Stone you are restoring.

The restore process is similar to the processes described in the *System Administration Guide*, Chapter 11.

## Restoring transaction logs with no changes in encryption keys

When using encrypted extents, the transaction logs are also encrypted with the same keypair as the Stone. As long as all required transaction logs for restore are encrypted with the same keypair as the newly restored Stone (or not encrypted), then restoring transaction logs into a Stone with encrypted extents follows the same process as restoring into a Stone with regular (non-encrypted) extents.

## Restoring transaction logs with changes in encryption keys

When using encrypted extents, over time you may change the keypair used to encrypt the extents (using `updatesecuredbf` or `copydbf` twice to decrypt and re-encrypt). When the stone is restarted with the new key, it will create a new transaction log that is encrypted using the new keypair. Since the full series of transaction logs will be encrypted with multiple keys, special handling is required to allow all the transaction logs to be restored.

There are several options:

- ▶ Use `updatesecuredbf` to update the encryption key for every transaction log so it is using the current Stone's encryption key. Any tranlogs that do not have the Stone's encryption key cannot be read, and restore will stop there.
- ▶ Use the method `Repository >> setTranlogPrivateKeysForRestore:` to provide the stone with the private key for all of the transaction logs that you will be restoring. Any transaction logs using keys that are not provided to the stone cannot be read, and restore will stop there.
- ▶ Always make backups immediately after changing the encryption key, and set up your systems for backup and restore, such that you will never need to restore older encrypted repositories. For example, you may decrypt the older backups and transaction logs and archive these as insecure files using your own secure storage mechanisms.

## Update transaction log keys

The following examples update tranlogs to the new Stone's key, `server_3_servercert.pem`.

In this example `tranlog1.sdbf` was encrypted with `server_1_servercert.pem`, and `tranlog2.sdbf` was encrypted with `server_2_servercert.pem`.

```
unix> updatesecuredbf tranlog1.sdbf -e server_3_servercert.pem
-D server_1_serverkey.pem
-J $GEMSTONE/examples/openssl/certs/server_1_server_passwd.txt
-K $GEMSTONE/examples/openssl/certs
-K $GEMSTONE/examples/openssl/private

unix> updatesecuredbf tranlog2.sdbf -e server_3_servercert.pem
-D server_2_serverkey.pem
-J $GEMSTONE/examples/openssl/certs/server_2_server_passwd.txt
-K $GEMSTONE/examples/openssl/certs
-K $GEMSTONE/examples/openssl/private
```

## Providing private keys to the Stone in Smalltalk code

Rather than updating all the keys for existing transaction logs, you may pass into the stone all the private keys for all the transaction logs that you will be restoring, using the method `Repository >> setTranlogPrivateKeysForRestore:` and creating instance of `GsTlsPrivateKey`.

```
Repository >> setTranlogPrivateKeysForRestore: arrayOfGsTlsPrivateKey
Specifies a list of instances of GsTlsPrivateKey representing private keys, that will be used to read encrypted tranlogs during a restoreFromArchiveLogs operation. The private key used to start the stone does not need to be included.
```

During a `restoreFromArchiveLogs` operation, the stone searches *arrayOfGsTlsPrivateKey* for a match to the public key stored in the encrypted tranlog. An error is raised if no match is found, and the restore operation fails.

```
GsTlsPrivateKey class >> newFromPemFile: pemFilename
Create an instance of GsTlsPrivateKey for the private key in the given filename. This key must not require a passphrase.
```

```
GsTlsPrivateKey class >> newFromPemFile: pemFilename
withPassphraseFile: passphraseFilename
Create an instance of GsTlsPrivateKey for the private key in the given filename. The key in this file is assumed to require a passphrase, which is in the given passphrase file.
```

There are other instance creation methods on `GsTlsPrivateKey`; refer to the image for other options.

Once `GsTlsPrivateKeys` are provided to the Stone, this list of keys is saved in the Stone, so you may execute this expression multiple times with new arguments, without losing key information.

### Example 2.1 Setting tranlog keys for restore

---

```
topaz 1> run
SystemRepository setTranlogPrivateKeysForRestore: {
(GSTlsPrivateKey newFromPemFile:
  '$GEMSTONE/examples/openssl/private/server_1_serverkey.pem'
withPassphraseFile:
  '$GEMSTONE/examples/openssl/private/server_1_server_passwd.txt')
.
(GSTlsPrivateKey newFromPemFile:
  '$GEMSTONE/examples/openssl/private/server_2_serverkey.pem'
withPassphraseFile:
  '$GEMSTONE/examples/openssl/private/server_2_server_passwd.txt')
}
%
```

---

After this is executed, the Stone can restore transaction logs encrypted with `server_1_servercert.pem` or `server_2_servercert.pem`, as well any tranlogs created with the current Stone's encryption key `server_3_servercert.pem`.

### Page Audit

To run **pageaudit** on a system with encrypted extents, you must include the same private key information as when starting a stone.

For example, when the extent files for a repository are encrypted using the key `server_2_servercert.pem`, then the following invocation will allow pageaudit to execute:

```
unix> pageaudit -D server_2_serverkey.pem
-J $GEMSTONE/examples/openssl/private/server_2_server_passwd.txt
-K $GEMSTONE/examples/openssl/private
```

### Hot Standby

You may use encrypted extents within a warm or hot standby system. The logsender and logreceiver do not require any special configuration; they pass along the records without decrypting them.

The slave stone must be started with the same encryption key as the master stone, so that the process restoring the tranlogs is able to decrypt them. If the encryption key on the master is changed, you should also update the encryption key on the slave's extents.

Alternatively, you can provide the new key to the slave system using `setTranlogPrivateKeysForRestore:`; however, this is not retained over shutdown/restart of the slave system, since it is in restore mode. After a restart of the slave system you will need to execute `setTranlogPrivateKeysForRestore:` again.

### printlogs/searchlogs

Tranlog analysis works with encrypted transaction logs; see "printlogs and searchlogs" on page 37.

## Programmatic Information about encryption

Methods have been added to System class to allow you to fetch information about encryption status.

```
System class >> stoneCommandLineArgs
  Returns a String representing the command line arguments used to start the stone.

System class >> repositoryEncryptionScheme
  Answers a symbol describing the algorithm used to encrypt the repository (extents
  and tranlogs) or #none if the repository is not encrypted.

System class >> repositoryIsEncrypted
  Answers a boolean indicating if the repository is encrypted.

System class >> repositoryPrivateKey
  If the repository is encrypted and the current user is SystemUser, returns an
  instance of GsTlsPrivateKey which represents the private key used to decrypt the
  repository. Returns nil if the repository is not encrypted. Raises an exception if this
  method is invoked by any user except SystemUser.

System class >> repositoryPublicKey
  If the repository is encrypted, returns an instance of GsTlsPublicKey or
  GsX509Certificate which represents the public key used to encrypt the repository.
  Returns nil if the repository is not encrypted.
```

## Modifying Encrypted Files

The utilities **copydbf** and **updatesecuredbf** allow you to perform a number of operations on encrypted extents, transaction logs, and backup files.

**copydbf** allows:

- ▶ copy the file without changing encryption state. This does not require cert files.
- ▶ encrypt an extent or transaction log file (backups cannot be encrypted using copydbf)
- ▶ decrypt an extent, transaction log, or backup to a regular file

**updatesecuredbf** allows:

- ▶ change the encryption key of an extent or transaction log, or one of the encryption keys of an encrypted backup, to a different key.

Note that it is possible to create secure backups that are signed, but not encrypted. The signing key for these backups cannot be changed, and these backups cannot be converted into encrypted nor into regular files.

## Examples of modifying encrypted files

Keys are not needed to make a copy of an encrypted dbf. **copydbf** on an encrypted file, without using decryption keys, results in a file that is encrypted with the same key as the source file.



## To encrypt an extent

This example encrypts the distribution extent into `secextent0.sdbf`.

```
unix> copydbf $GEMSTONE/bin/extent0.dbf
      $GEMSTONE/data/secextent0.sdbf -e server_2_servercert.pem
      -s 256 -K $GEMSTONE/examples/openssl/certs
```

If you have multiple extents, you must encrypt **all** extents with the same key, to be able to start a Stone.

The same process can be used to encrypt an extent snapshot backup. Unencrypted programmatic fullbackups cannot be encrypted.

You may also encrypt an unencrypted transaction log, although normally encrypted transaction logs are created by a Stone using encrypted extents.

## To start a stone on an encrypted extent

When the Stone's configuration file specified extent files that are encrypted using the key `server_2_servercert.pem`, then the following invocation will start the stone on those extents.

```
unix> startstone -D server_2_serverkey.pem
      -J $GEMSTONE/examples/openssl/private/server_2_server_passwd.txt
      -K $GEMSTONE/examples/openssl/private gs64stone
```

## To decrypt an extent or extent snapshot backup

This example decrypts `secextent0.sdbf`, which was encrypted with `server_2_servercert.pem`, into `decrypextent0.dbf`.

```
unix> copydbf secextent0.sdbf decrypextent0.dbf
      -D server_2_serverkey.pem
      -J $GEMSTONE/examples/openssl/private/server_2_server_passwd.txt
      -K $GEMSTONE/examples/openssl/certs
      -K $GEMSTONE/examples/openssl/private
```

If you have multiple extents, you must decrypt **all** extents to be able to start a Stone.

## To decrypt an extent or extent snapshot backup

This example decrypts `sectranlog1.sdbf` into `decryptranlog1.dbf`.

```
unix> copydbf sectranlog1.sdbf decryptranlog1.dbf
      -D server_2_serverkey.pem
      -J $GEMSTONE/examples/openssl/private/server_2_server_passwd.txt
      -K $GEMSTONE/examples/openssl/certs
      -K $GEMSTONE/examples/openssl/private
```

Decrypted transaction logs can be restored without special handling, including in a series with other transaction logs that are encrypted.

## To change the key of an encrypted extent or extent snapshot backup

This example updates the original key `server_2...` to `server_3...`. The modification is done to `secextent0.dbf` in place; we recommend making a copy before using `updatesecuredbf`.

The stone using this extent must have been shut down cleanly.

```
unix> updatesecuredbf secextent0.sdbf
      -e server_3_servercert.pem -D server_2_serverkey.pem
      -J $GEMSTONE/examples/openssl/private/server_2_server_passwd.txt
      -K $GEMSTONE/examples/openssl/certs
      -K $GEMSTONE/examples/openssl/private
```

If you have multiple extents, you must update the keys for **all** extents to the same key to be able to start a Stone.

## To change the key of an encrypted transaction log

This example updates the original key `server_2...` to `server_3...`. The modification is done to `sectranlog1.dbf` in place; we recommend making a copy before using **updatesecuredbf**.

```
unix> updatesecuredbf sectranlog1.sdbf
      -e server_3_servercert.pem -D server_2_serverkey.pem
      -J $GEMSTONE/examples/openssl/private/server_2_server_passwd.txt
      -K $GEMSTONE/examples/openssl/certs
      -K $GEMSTONE/examples/openssl/private
```

Transaction logs are normally part of a series that are retained to allow restore. Before a transaction log can be restored, it must either be using the same key as the extents for the Stone into which it is being restored, or the keys must be supplied using `setTranlogPrivateKeysForRestore::`.

## To change the encryption key of an encrypted fullbackup

Encrypted programmatic fullbackups are always signed; you will need to supply the signing key (and passphrase, if required), as well as the previous and new encryption keys.

Encrypted backups may be encrypted with up to 8 different public keys, and the private key corresponding to any of these public keys can be used to decrypt. An invocation **updatesecuredbf** changes a single one of these public key encryptions, leaving the others unchanged.

```
unix> updatesecuredbf encryptedBackup1.sdbf
      -e backup_encrypt_2_clientcert.pem
      -J $GEMSTONE/examples/openssl/private/backup_encrypt_1_clientcert_passwd.txt -D backup_encrypt_1_clientcert.pem
      -S backup_sign_3_clientkey.pem
      -T $GEMSTONE/allcerts/backup_sign_3_client_passwd.txt
      -K $GEMSTONE/examples/openssl/certs
      -K $GEMSTONE/examples/openssl/private
```

For a multi-file fullbackup, you must update the keys for **all** backup files to allow them to be used for restore.

## To decrypt a fullbackup

This example decrypts `encryptedBackup1.sdbf` into `Backup.dbf`, using the private encryption key `backup_encrypt_2_clientkey.pem`.

```
unix> copydbf encryptedBackup1.sdbf Backup.dbf
-D backup_encrypt_2_clientkey.pem
-J $GEMSTONE/examples/openssl/private/backup_encrypt_2_client_p
asswd.txt -K $GEMSTONE/examples/openssl/certs
-K $GEMSTONE/examples/openssl/private
```

For a multi-file fullbackup, you should decrypt **all** backup files to allow them to be used for restore.

## Encrypting and Decrypting GsTls\* keys

The following methods allow some ability to encrypt and decrypt `GsTlsPublicKey`, `GsX509Certificate`, and `GsTlsPrivateKey`.

`GsTlsPublicKey` >> `encrypt: aByteObj`

Encrypts `aByteObj` using the receiver which must be an RSA public key or X509 certificate object. This method supports RSA keys only. Encryption is performed using the RSA-OAEP padding algorithm.

Only small amounts of data may be encrypted using asymmetric keys. The maximum allowed size of `aByteObj` is dependent on the number of bits in the receiver's public key. For 2048 bit keys, the maximum size is 214 bytes. For 4096 bit keys, the maximum size is 470 bytes.

`aByteObj` must be an instance of a byte class, typically `ByteArray`. Byte objects with character sizes greater than 1 are not supported.

`GsX509Certificate` >> `encrypt: aByteObj`

Encrypts `aByteObj` using the receiver which must be an RSA public key or X509 certificate object. This method supports RSA keys only. Encryption is performed using the RSA-OAEP padding algorithm.

Only small amounts of data may be encrypted using asymmetric keys. The maximum allowed size of `aByteObj` is dependent on the number of bits in the receiver's public key. For 2048 bit keys, the maximum size is 214 bytes. For 4096 bit keys, the maximum size is 470 bytes.

`GsTlsPrivateKey` >> `decrypt: aByteArray`

Decrypts `aByteArray`, which was previously encrypted with the receiver's matching public key. Only RSA keys are supported for encrypting and decrypting.

Examples are provided in the following methods:

```
GsTlsPrivateKey class >> encryptDecryptBase64Example
```

```
GsTlsPrivateKey class >> encryptDecryptExample
```

## 2.2 Changes in Users and ObjectSecurityPolicies

### CodeLibrarianUser no longer needed

The CodeLibrarianUser, and the associated CodeLibrarianObjectSecurityPolicy, is no longer needed in 3.6, with the changes in the way GemStone code is stored and built which bring GemStone further into alignment with Rowan code management.

This UserProfile, and the object security policy, are removed in new repositories. Upgraded repositories will contain CodeLibrarianUser in AllDeletedUsers, and the object security policy will remain but is not used.

### DataCurator no longer has permission to modify Globals

In previous versions of GemStone, the Globals SymbolDictionary had the DataCuratorObjectSecurityPolicy. As of v3.6, the Globals object has the SystemUserObjectSecurityPolicy; so, only SystemUser is able to add, remove, or modify entries in Globals.

While necessary to close the security hole, this change means that some operations that were previously possible, such as adding or removing keys from Globals, including installing Portable or Legacy Streams, will now encounter a security error, and should be performed by SystemUser rather than DataCurator.

### SystemUser no longer subject to privilege checks

SystemUser is intended to be a super-user, without restrictions to its powers. While methods that removed privileges from SystemUser were disallowed, there were ways to modify the SystemUser's privileges, which left the SystemUser without the ability to perform the privileged tasks. (#48905)

Now, SystemUser is not subject to SecurityErrors when performing privileged tasks, regardless of its permissions.

### SystemUser now exposes sub-dictionaries

The SystemUser's SymbolList now includes SymbolDictionaries that were previously only referenced from keys within Globals and thus were less visible. This includes ObsoleteClasses, GsCompilerClasses, the Streams hierarchies, and new Rowan Kernel SymbolDictionary.

### Changes in handling of transient symbolList

Previously, each session always had a transient symbolList, which was a copy of the symbolList for that UserProfile, and the transient symbolList was used for symbol resolution. Now, by default the persistent symbolList is used, making it easier for applications that do not need to distinguish between transient and persistent symbolLists. Querying for or assigning a transient symbol list, creates a copy of the persistent symbolList and provides the previous behavior.

In particular, previously UserProfiles without CodeModification privilege could not easily pick up changes in the persistent symbolList.

The new handling of the transient symbolList is described in more detail under "Change in handling of transient symbol list" on page 46.

## Whitelist for performOnServer:

Accessing OS functions using `performOnServer:` (and the related `GsHostProcess` functionality) is very useful, and for some applications it may be necessary for users that should not have general OS command access, to be able to execute specific OS commands.

This can be set up via a per-user command whitelist. To allow a user with `NoPerformOnServer` privilege to execute a specific command, the full path (starting with `/`) to the command is put on a whitelist for that specific user. The full path must then be specified on the `performOnServer:` or `GsHostProcess` method argument; the paths and commands must match exactly for the execution to be permitted.

There are no checks or restrictions on arguments to the commands; only the commands themselves are examined and allowed (or not).

The following methods add one or more commands to the whitelist:

```
UserProfile >> addPerformOnServerCommand: fullPathToCommand
UserProfile >> addPerformOnServerCommands: arrayOfCommands
```

To report the existing whitelist:

```
UserProfile >> performOnServerCommands
```

To remove commands from the whitelist:

```
UserProfile >> removeAllPerformOnServerCommands
UserProfile >> removePerformOnServerCommand: aString
UserProfile >> removePerformOnServerCommands: arrayOfString
```

The requirement to use full paths in the whitelist and in the argument to `performOnServer:` applies to operating system commands such as `ls`. While users without `NoPerformOnServer` can pass commands such as `'pwd'` and `'ls'` to `performOnServer:`, and rely on the path lookup, users with `NoPerformOnServer` must configure the whitelist to include the full path to the specific executable that is allowed, such as `'/bin/pwd'` and `'/bin/ls'`, and pass the same full path to `performOnServer:.`

### Example

For example, as `DataCurator` or another privileged user, update a non-privileged user `ReadingUser` by executing:

```
(AllUsers userWithId: 'ReadingUser')
  addPrivilege: 'NoPerformOnServer';
  addPerformOnServerCommand: '/usr/bin/git'.
System commitTransaction.
```

Now, as `ReadingUser`, the following will succeed:

```
System performOnServer: '/usr/bin/git log'
```

But `ReadingUser` cannot perform other OS commands, such as `ls` or `cp`. Each individual executable to be allowed must be specifically entered in the whitelist.

## GsSysLog

Writing to the System log using GsSysLog is also controlled by NoPerformOnServer, and can be whitelisted with a the string ' /GsSysLog/writeLogMessage ', or using special protocol:

```
UserProfile >> disableSystemLogAccess
UserProfile >> enableSystemLogAccess
UserProfile >> hasSystemLogAccess
```

## 2.3 Changes in Backup, Restore and Hot Standby

### Tranlog version numbers following restore

The way tranlog version numbers are handled during `commitRestore` has changed in this release.

Sending `commitRestore` following restoring a fullbackup and transaction logs normally creates a tranlog with the number that is one higher than the last transaction index in the backup/restored tranlogs (this number is reported by `copydbf -i` on backups and transaction logs).

In previous releases, if a tranlog with this name already existed, it would instead choose the number equivalent to the next unused tranlog filename. Starting in v3.6, it will now always write to a filename with a number one higher than the restored repository; it is an error if a tranlog with that number exists, and the `commitRestore` will fail.

Generally, you should use care in managing transaction logs locations to avoid confusion in the sequence, or risking fork-in-time scenarios.

### Hot Standby Failover Changes

In a hot standby system, there is a master Stone that continuously updates a slave Stone. On a planned failover, the master Stone must complete the update of the slave Stone; and then, usually, the slave Stone is made into the master, and the former master becomes the slave.

This process has been improved in v3.6.

Rather than using `System class >> suspendCommitsForFailover`, a new method, `System class >> failOverToSlave`, has been added.

The old method `suspendCommitsForFailover` prepared for a failover to the slave system, but did not prepare the master to become the new slave. While the process to failover to the slave itself is not different, the additional steps to setup the old-master to act as the new-slave are much simplified.

The new method `failOverToSlave`, when executed on the master Stone, performs the same steps as `suspendCommitsForFailover`, and in addition puts the master stone into restore from logs mode. In this mode, the master slave will not write any more tranlog entries, and is ready to be the new slave Stone.

When `failOverToSlave` is used on the master system, then once the previous-slave Stone executes `commitRestore`, the old slave and old master are effectively reversed. The logreceiver and logsender are started on the new slave and new master, respectively; the

old master/new slave executes `continuousRestoreFromArchiveLogs`:. The old slave system is now the master, and the old master system is now the slave.

## Added methods

`Repository >> failOverStatus`

To use on a master system to determine the `restoreStatus` of a slave system. A logreceiver periodically transmits `restoreStatus` to the logsender, and the logsender stores that status into the primary shared cache. The `failOverStatus` methods fetch that restore status from the primary cache and report on it, thus allowing a session on a primary stone to observe the restore status of the slave system.

For example:

```
topaz 1> exec SystemRepository failOverStatus %
This repository last checkpoint file 2 record 4155
Other repository restored to checkpoint file 2 record 4155
```

`Repository >> failOverToSlave`

For use on the master system of a hotstandby pair. Suspends commits, starts a checkpoint, and waits for completion of the checkpoint. Then puts this repository into restore from logs mode.

## New Planned Failover Process in Detail

**Step 1.** Confirm that all tranlog records have been transmitted to the slave system, and that the slave system has restored all transaction log records.

On the master system, before failing over, execute

```
SystemRepository failOverStatus
```

and verify that the master and slave are on the same log and record.

**Step 2.** On the master system, execute:

```
SystemRepository failOverToSlave
```

which will suspend commits, and perform a checkpoint, and enter restore from logs mode. Commits are disallowed. Do not shut down the master Stone.

Executing this method sets up the master system to automatically become a slave after the previous slave, new master has performed the `commitRestore`, and the logsender and logreceiver are set up on the different nodes.

**Step 3.** Stop the logsender process on the master system using the **stoplogsender** utility command.

```
stoplogsender -P 57222
```

**Step 4.** On the slave system, stop the logreceiver process using the **stoplogreceiver** utility command.

```
stoplogreceiver -P 57222
```

**Step 5.** On the slave system, send the message

```
Repository >> stopContinuousRestore to exit continuous restore mode, then
Repository >> commitRestore to terminate the restore process and enable
logins. The slave system is now ready for use as the new master.
```

**Step 6.** If you have changes in the configuration parameters for the slave system, you should reset these to the values that you want for the master, in the configuration file or files used by the former-slave system. Depending on the specific parameters, you may need to restart the former-slave system for these changes to take effect.

You will also need to enable any automated processes that you normally run on the master, such as monitoring, on the former-slave system.

**Step 7.** Client applications will have to reconnect to the former-slave system, which now becomes the primary (master) system.

Note that, when using `failOverToSlave` in v3.6, it is no longer necessary to make a backup of the new master and restore it into the new slave. All that is needed to complete the setup of the old master as the new slave, is to setup the logsender/logreceiver and put the new slave into continuous restore mode.

**Step 8.** Start the logsender process on the former slave, new master system using the `startlogsender` utility.

```
startlogsender -P 57222 -A masterListeningAddress -s masterStone
```

**Step 9.** Start the logreceiver process on the former master, new slave system using the `startlogreceiver` utility.

```
startlogreceiver -P 57222 -A masterNode -T tranlogDirectories  
-s slaveStone
```

**Step 10.** Put the former master, now slave Stone into continuous restore mode. To do this, log in and execute `Repository >> continuousRestoreFromArchiveLogs:`, passing in the list of directories.

After entering continuous restore mode, this method will exit and you can log out. The hot standby is now operational in the reverse direction.



## 2.4 Other Changes related to administrative operations

### GsObjectInventory changes

#### Improved thread handling

GsObjectInventory profiling methods (other than `fastProfileObjects*`) previously used two threads to perform the scan. Now, you can control the number of threads using the standard way to control threads in a repository scan; that is, by executing

```
SystemRepository setDefaultNumThreads: numThreads
```

before executing the profiling method.

In addition, the following method has been added:

```
profileObjectsIn: aCollectionOrGsBitmap threads: numThreads
```

Same as `profileObjectsIn:` except that it allows specifying the number of threads to use for the scan.

#### Removed methods `profileObjectsInBm:` and `profileObjectsInFile:`

The methods `GsObjectInventory >> profileObjectsInBm:` and `GsObjectInventory >> profileObjectsInFile:` were unnecessarily added in v3.5.3, and have been removed.

`GsObjectInventory >> profileObjectsIn:*` now accepts a `GsBitmap` directly. To analyze objects in a file, read the objects from the file into a `GsBitmap` first.

### Deprecated `findAllReferencePaths` feature removed

The methods `Repository >> findAllReferencePathsToObjects*`, were intended to find multiple reference paths to objects within a single scan. These were deprecated and unsupported as of v3.4, since they failed to scale well in practice, and have been removed in this release.

### System `currentSession*` includes logsender, logreceiver

The report output by `System class >> currentSessionsReport`, `System class >> currentSessionNames`, and related methods, now includes a logsender or logreceiver process that is attached (that is, a logsender or logreceiver that was started using the `-s` key).

## 2.5 Changes in Errors

### AlmostOutOfMemoryError

In addition to the class `AlmostOutOfMemory/6013`, a new exception class, `AlmostOutOfMemoryError/6022`, has been added.

- ▶ `AlmostOutOfMemory` is a Notification that is enabled by default, although you must enable signalling or set a handler block before any action is taken.
- ▶ `AlmostOutOfMemoryError` is an Error that is disabled by default, but once enabled does not require additional actions.

Note that the threshold setting is shared between these two classes.

### AlmostOutOfStackError

In addition to the class `AlmostOutOfStack/2059`, a new exception class, `AlmostOutOfStackError/2519`, has been added.

- ▶ `AlmostOutOfStack` is a Notification that is enabled by default.
- ▶ `AlmostOutOfStackError` is an Error that is disabled by default. If it is enabled, then `AlmostOutOfStackError` will be signalled instead of a `AlmostOutOfStack`.

### ExitClientError

The class `ExitClientError/3004` has been added.

`ExitClientError` allows you to exit the client from within GemStone Smalltalk code. See “Exiting the Client from Smalltalk” on page 51 for more details.

### UncontinuableError

The Exception class `UncontinuableError ERR_EXC_RETURN_DISALLOWED/2758` has been added. This is used when return disallowed, a return from `on:do:` would cross frame of C primitive, user action, or FFI call.

### Other added Exceptions

The follow Exceptions have been added:

**ABORT\_ERR\_CommitsSuspended/3002**

**ERR\_illegalRamSelf/2156**

### Removed Exceptions

`StDBErrStackTooBig / STDB_ERR_STACK_TOO_BIG/1036`

`StDBErrMaxLits / STDB_ERR_MAX_LITS/1039`

`StDBErrMaxSelectors / STDB_ERR_MAX_SELECTORS/1040`

`StDBErrTooManyBlocks / STDB_ERR_TOO_MANY_BLOCKS/1055`

`ERR_RUBY_SystemExitError/2752`

`#repErrNoFreePages/REP_ERR_NO_FREE_PAGES/4003`

## 2.6 Changes in Utilities

### copydbf

**copydbf** has been rewritten in this release, separating the codebase from other utilities. In addition to the new features added to support encrypted extents, the semantics of copy has been made more intuitive, and there are a number of other changes.

The output printed by a **copydbf** command has changed significantly. The output is written to stderr rather than stdout, includes additional information, and some of the information has been rearranged within the output.

#### Secure extent support options added

The following options have been added to **copydbf** to support working with secure extents. See the section “Encrypted Extents and Transaction Logs” on page 17 for details.

- D** *privKey*  
Use *privKey* to decrypt the source extent, backup or tranlog.
- e** *cert*  
Encrypt the destination extent or tranlog with the given public key or cert.
- F**  
Use `fsync(2)` to flush all data to the destination disk before **copydbf** exits.
- j** *passphrase*  
Use the given passphrase text to read the private key specified by **-D**.
- J** *pfFile*  
Use passphrase contained in file *pfFile* to read the private key specified by **-D**.
- O**  
Some operations on secure extents require a clean shutdown. Specifying this option overrides that requirement.
- s** *keyLength*  
Specifies the encryption key length in bits. *keyLength* may be 128, meaning AES-XTR-128, or 256, for AES-XTR-256.

When using encrypted files:

- ▶ **copydbf** does not support creating files that are both encrypted and compressed. Compressing an encrypted file has little benefit since the encrypted bytes appear random.
- ▶ **copydbf** on encrypted files, or operations to encrypt or decrypt dbf files, cannot be combined with changes to the byte ordering. **-l**, **-m**, and **-p** commands are disallowed in these cases. All encrypted files are written in the byte order of the machine that the command is executing on.
- ▶ **copydbf -i** does not require encryption keys to provide the basic information about an encrypted extent. If you pass in the private key and keyring, the it will verify that the given key can be used to decrypt the extent.

#### Previous, non-working **-s** has new meaning

The **-s** argument to **copydbf** previously preallocated size for the new copy of the database file. This is not useful with modern systems. The **-s** has been repurposed to specify the encryption key size of a secure backup.

## copydbf appends zip/secure extensions as needed

When **copydbf** is used to copy a file to a target filename, the correct extension (`.gz`, `.lz4`, `.dbf`, or `.sdbf`) is appended to the target filename, if needed according to the format of the destination file. It is recommended to omit the extension from the target filename.

## Changes in file naming with copydbf to directory target

When the **copydbf** copy destination is an existing directory rather than a filename, and the `-f newbasename` argument is not specified, then **copydbf** composes the name of the target file.

In earlier versions, the new file was named `tranlog`, `backup`, or `extent`, followed by the file's id, ending with `.dbf`. Now, the base name of the original file is preserved, with the appropriate extension added if needed.

## copydbf semantics change for zip status

In previous releases, a **copydbf** operation wrote the destination file using the compression specified in the **copydbf** command, with the understanding that if no compression flag was included, the target would not be compressed. The compression state of the originating file was not considered; all state in the destination was as specified in the command line.

With v3.6, this has changed, and the compression flags are considered as transformations to apply to the originating file. In 3.6:

- ▶ it is now **not** legal to specify a compression flag when the originating file is already compressed (unless the type of compression changes).
- ▶ **copydbf** of a compressed file by default will now result in a compressed file; you must use the `-u` if you wish the destination file to be uncompressed.

## Required to use -u to copydbf a compressed file into raw partition

You may not put a compressed file onto a raw partition. With the new semantics, in order to **copydbf** a compressed file onto a raw partition, you must include the `-u` flag to explicitly decompress during copy.

## copydbf output to stderr vs. stdout

**copydbf** status output (details about the copied file) now goes to `stderr`, rather than `stdout`.

This allows you to pipe the **copydbf** copy operation output to `stdout`. Support for this is added either by specifying `stdout` as the **copydbf** target name, or by using the added `-a` option. That is, either of these operations:

```
unix> copydbf extent0.dbf stdout > thecopy1.dbf
unix> copydbf extent0.dbf -a > thecopy2.dbf
```

## copydbf no longer accepts NRS for remote access, -n removed

The arguments to **copydbf** should be filenames, not NRS, and **copydbf** of remote files requires that the file system be mounted. The `-n` option to specify a NetLDI has been removed from **copydbf**.

## Record-level compression argument -c removed

The **-c** argument controlled record-level compression. This is no longer needed and has been removed.

## Testing option -d added

The **-d** (dry run) option allows you to verify the results of a copy, without actually making a copy.

## netldidebug utility added

A new utility has been added, which allows you to modify the debug state of a running NetLDI (that is, the state resulting from omitting or including **-d** with **startnetldi**), without the need to shut down and restart the NetLDI.

```
netldidebug netldiNameOrPort enabledebug | extraddebug | disabledebug
where:
```

```
netldiNameOrPort -- the name or port of a running NetLDI
enabledebug -- enable debugging (equivalent to startnetldi -d)
extraddebug -- enable extra debugging, for argument expansion
disabledebug -- turn off all debugging
```

## printlogs and searchlogs

The **printlogs** and **searchlogs** utilities have been updated to work with encrypted tranlogs; you must include the private key, the passphrase if needed, and the keyring dir, using the following keyword arguments:

```
keyring privKeyPath privkey keyFileName privkeypf keyPassphrase
privkeypffn fileContainingPassphrase
```

Only one of **privkeypf** and **privkeypffn** are needed.

For example,

```
unix> printlogs keyring $GEMSTONE/examples/openssl/private/
privkey server_2_serverkey.pem
privkeypffn $GEMSTONE/examples/openssl/private/server_2_server_
passwd.txt tranlog4.sdbfpstack
```

## pstack

**pstack** now prints output to stderr, rather than stdout

## startlogsender, startlogreceiver

### startlogreceiver -D argument for additional logging

The **startlogreceiver** utility now accept the **-D** argument. This provides logging similar to the **-d** argument, but also prints commands sent between the logsender and logreceiver. The logreceiver's log file will be much larger.

## startlogsender -u option removed

**startlogsender** no longer accepts the **-u** option; logs records are always compressed for transmission.

## startlogsender option to specify tranlog prefix

The **-g** *tranlogFilePrefix* argument has been added to **startlogsender**. This is only needed if not using the **-s** option to specify the stone name, and if the master stone's transaction logs are not named at the default, 'tranlog' (that is, a non-default STN\_TRAN\_LOG\_PREFIX). If using the **-s** argument, the argument to **-g** is ignored, and the Stone's setting is used.

## startnetldi

### Added option to configure threads

The NetLDI is now multithreaded, with a default of 10 threads, or 1 for a X509-Secured NetLDI or a NetLDI running as root.

To control the number of threads, the **-t** option has been added to **startnetldi**.

**-t** *numberOfThreads*

The number of threads that netldi will use, default 10, min 1, max 500. If **-S** specified, or netldi running as root, only uses one thread.

### Resolving hostnames using reverse DNS lookup

Previously, the NetLDI would by default lookup IP address to fetch the host names of peers, using reverse DNS lookup; this could be disabled using the **-N** argument.

These names are rarely printed, and configuration issues in the reverse lookup could cause timeouts. Now, by default, there is no lookup of IP address. The **-N** argument remains but has no effect.

To enable printing of hosts by name, the new option **-H** has been added.

**-H** Do reverse DNS lookup and print peer hostnames in addition to IP address in logs.

The IP addresses are now printed in the NetLDI log when the NetLDI is in debug mode.

## startstone

### Resolving hostnames using reverse DNS lookup

Previously, the Stone would by default lookup IP address to fetch the host names of peers, using reverse DNS lookup. Since this is subject to issues if DNS is not configured correctly, this has been changed, and by default, there is no lookup of IP address.

To enable printing of hosts by name, the new option **-H** has been added.

**-H** Do reverse DNS lookup and print peer hostnames in addition to IP address in logs.

### Underscores removed from prompts for username and password

If you enter stopstone without the username/password, the prompts for the username and password began with an underscore. These underscores have been removed.

## startstone and pageaudit secure database options added

The following have been added to startstone and pageaudit:

**-D** *privateKeyFile*

Specifies the private key file used to access the encrypted database. The file must be in PEM format. If the private key has a passphrase, then either **-j** or **-J** are also required. The file name must not include a path. Use **-K** to specify the path to the private key file.

**-j** *passphrase*

Specifies the passphrase for the private key specified by **-D**.

**-J** *pffile*

Specifies a file containing the passphrase for the private key specified by **-D**.

**-K** *dir*

Specifies one or more keyring directories. Directories may be separated by colons and/or **-K** may be specified multiple times.

See the section “Encrypted Extents and Transaction Logs” on page 17 for details.

## updatesecuredbf utility added

The **updatesecuredbf** utility allows you to modify the encryption key for an encrypted extent file, transaction log, or backup file. **updatesecuredbf** can only be used with already encrypted dbf files; use **copydbf** to encrypt or decrypt a dbf file.

For more information, see “Modifying Encrypted Files” on page 24, “Examples of modifying encrypted files” on page 24, and the *System Administration Guide* chapter 12 and Appendix B.

## 2.7 Topaz changes

### Prompt change when input nested files

The **input** command allows you to input a series of topaz commands from an file. These files may themselves invoke **input** on other files.

To make it easier to see the level of file that is being input, the topaz prompt now includes a + (plus sign) for each level of file input nesting, up to 20 (the limit for nesting input files).

For example,

```
topaz 1> input aFile
topaz 1 +> <commands in aFile>
topaz 1 +> input nestedFile
topaz 1 ++> <commands in nestedFile>
topaz 1 +> <further commands in aFile>
```

### Added commands

#### TFILE

This command allows you to input a tonel format class definition or class extension file. The methods within the file are all compiled. The Class specified in the file must already exist (i.e. class definition execution is not attempted).

## DISPLAY/OMIT ALLOOPS

The command **display/omit classoops** was not correctly named, since it also enabled display of the oops of specials; it has been replaced by the new subcommands **display/omit alloops**. The command under the legacy name is still available.

## DISPLAY/OMIT DECIMALBYTES

**display decimalbytes** is similar to **display bytes**, but displays the bytes of the contents as decimal rather than hexadecimal. Executing **omit bytes** or **omit decimalbytes** disables printing of any numeric byte contents.

## SET SINGLECOLUMN ON/OFF

When stacks are printed, methods that have more than eight method arguments and temporary variables print their variables in four columns. With this new command, you can disable this feature, and display each variable on a single line.

**set singlecolumn** is off by default, providing the behavior in previous releases.

## Changes in commands

### STRINGS / STRINGSIC searches class comments

The **strings** and **stringsic** commands now include string search matches in class comments, as well as in class and instance method source code.

### SET CLASS to nil

You can now use **set class nil** to clear the selected class.

### ALLSTACKS limit removed

The **allstacks** command previously was limited to 1000 frames per process. This limit has been removed.

### QUIT / EXIT with file input

The **quit** or **exit** command is now ignored when reading a file using **input**, when stdin is a tty.

If topaz stdin is redirected to a file and the file does not end with a **quit** or **exit** command, topaz considers EOF on stdin to be an error, that contributes to non-zero exit status of topaz.

### DEFINE no longer updates ErrorCount

The predefined **ErrorCount** variable returns the number of errors. Topaz predefined variables should not be updated; **define ErrorCount** has been disallowed.



## 2.8 Configuration Parameter Changes

### Added Configuration Parameters

#### **GemDebuggerActive**

GemDebuggerActive is a runtime-only configuration option for the Gem, enabled by:

```
System gemConfigurationAt:#GemDebuggerActive put: true
```

This controls debugging of exceptions signaled from within user actions.

When true, exceptions for which no handler is found are delivered with a complete stack, including the frames of the user action. When true, an `AbstractException >> return` disallowed from within a user action, will be signaled as not trappable so the debugger in the GCI client can debug a complete stack.

Default: false

For more details, see “Handling Errors in user actions” on page 58.

#### **StnReverseDns**

Runtime-only configuration parameter for the Stone. If false, the IP addresses of other nodes are reported in the Stone log as IP addresses only. If true, the Stone uses reverse DNS lookup to get the node’s name, and reports both IP address and node name.

Default: true if Stone started with **-H**, false otherwise

See also “Resolving hostnames using reverse DNS lookup” on page 38

### Changed Configuration Parameter

#### **STN\_NUM\_LOCAL\_AIO\_SERVERS**

The configuration parameter `STN_NUM_LOCAL_AIO_SERVERS` now configures the number of AIO threads in the Stone, rather than the AIO page servers; the name has been retained for simplicity.

Historically, this was limited to 1 per extent. With modern hardware, this restriction is no longer valid. The default has been changed to -1, which allows the system to calculate the number of AIO page server threads.

With a setting of -1:

- ▶ If the repository size is > 100GB, and the host has at least 20 cores, the number of threads is the larger of 5 or number of extents
- ▶ if the repository size is > 10GB and host has at least 8 cores, the number of threads is the larger of 2 or number of extents
- ▶ otherwise, 1 per extent up to a maximum of 4

## 2.9 Changes in Statmonitor and Cache Statistics

### Statmonitor command line details

With 3.6, statmonitor writes an additional line to the statistics data file. This is displayed in VSD, in the **Main Window > File > Info** dialog.

This feature required changes to VSD as well as GemStone; the changes to support this additional line in the data file were made in the previously released VSD v5.5.

**VSD 5.5 or later is required to read statmonitor data files written by GemStone/S v3.6.**

### Default process cache names

There are some changes in the default names for Gems in v3.6.

- ▶ Previously, Topaz Linked and RPC sessions both displayed with the default name 'Topaz'. Now, Gems from RPC logins from Topaz, by default, are named 'TopazR', and Gems from Linked topaz logins are now named 'TopazL'. These names can be changed using **-u** or **set cachename**.
- ▶ Other applications, such as GBS, previously wrote the name as 'GemNN', where NN was the cache slot index. Now, the default name is 'Gem'.

### Topaz linked sessions distinguished in statmonitor data

When a topaz linked process logged out, and logged in again, it was previously treated as a single continuous session. However, this produced incorrect or misleading statistics, since values that were reset over logout and login were treated as continuous. (#48650)

Now, each topaz linked login through subsequent logout will be recorded as a separate session in statmonitor data.

### FailedCommitCount includes commits that succeeded on retry

If a commit fails, but is retried and succeeds, previously this was counted as a failed commit and included in the cache statistic FailedCommitCount, although from the point of view of the end-user this was a successful commit. Now, commits that succeed on retry are not included in FailedCommitCount (#48387)

New related cache statistics include:

#### **FailedAttemptedCommitCount** (Gem)

The number of attempts to commit or continue a transaction which failed.

#### **LastFailedCommitReasonCode** (Gem)

A reason code identifying the cause of the most recent failed commit:

- 1 = Reduce Conflict (RC) Failure
- 2 = Dependency Map Failure
- 3 = General commit failure (usually write-write conflicts)
- 4 = Retry Failure
- 5 = Commit Disallowed
- 6 = Retry Limit Exceeded
- 7 = Symbol Failure
- 8 = Lock Failure

## Changed Cache Statistics

**PagesWaitingForRemovalInStoneCount** has been removed.

It is replaced by the following, which track the Page Manager thread's work queue:

**PagesWaitingForPagemgr** (Stone)

The number of pages in the Stone in the Page Manager's work list.

**PagesWaitingForRemoval** (Stone)

The number of pages in the Stone that are waiting to be removed from the Shared Page Cache by the Page Manager.

**CodeCacheSizeBytes** has been renamed to **CodeCacheSizeKBytes**, and is now in units of KB.

## Added Cache Statistic

**CacheSerialNum** (All)

The serial number assigned by the shared page cache monitor to each process that attaches to the shared page cache.

## Removed Cache Statistics

The following cache statistics have been removed:

**CHeapSizeKB**

**PinnedPrivatePagesCount**



# Changes in GemStone Smalltalk

---

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

Changes in Classes, compilation, and code management ..	45
Changes in Numeric/Time classes .....	47
ExternalSession Changes .....	49
Other Changes and Enhancements.....	51
Deprecated and removed classes and methods .....	52

## 3.1 Changes in Classes, compilation, and code management

### Instance variable name length limit

Previously, instance variable names were limited to 64 characters; this has been increased. Now, class names, instance variable names, class variable names, class instance variable names, method temporaries, and method arguments all have a limit of 1024 characters.

### Literal Arrays may now embed literal ByteArrays

The Array literal forms `#( element1 element2 ...)`, previously could embed most other literal kinds with the exception of literal ByteArrays `#[ int1 int2 ...]`. Now, ByteArrays can also be embedded.

For example, this is now legal:

```
#( #[1 2 3 4] 'test' 1 2.0 )
```

### Block not optimized no longer reports CompilerWarning

The compiler cannot inline blocks under some conditions, when the blocks are complex. Previously, this resulted in Warnings, such as "WARNING: at line 21, to:do: not optimized". These Warnings were not useful and are no longer produced.

## Change in handing of transient symbol list

Historically, on login a copy of the UserProfile's symbolList is made available as the transient symbolList, and the transient symbolList is then used for all lookups for classes and other variables. This allows a separation between the shared symbolList for that UserProfile, and any changes the user is making while logged in. The transient symbolList was kept in session state, and was not affected by commit or abort. If there were changes in the UserProfile's persistent symbolList (made by other sessions), a session's transient symbolList did not automatically pick these changes up on commit or abort.

In 3.6, this behavior has been modified to make it easier to keep the transient symbolList in sync with the persistent symbolList.

Now, on login, the transient symbolList is a reference to, not a copy of, the persistent symbolList. Applications that do not need to modify the transient symbolList will automatically pickup changes by commit or abort, without extra effort.

Executing any of the following methods will create a copy of the transient symbolList in session state. This results in the same behavior as in previous releases.

```
System class >> refreshTransientSymbolList
GsCurrentSession >> transientSymbolList: aSymbolList
GsCurrentSession >> transientSymbolList
```

## Changes in Class Categories

GemStone Classes have a class category, which are not visible in GBS but are displays in tools such as Jade. There are a number of changes in the categories of classes in 3.6.

## Private method to rename an instance variable

A method has been added to allow you to rename an instance variable in place, without versioning the class.

Note that this is not considered a safe operation. This is not intended for use in critical systems, or under production conditions. Use caution and perform thorough testing, since there may be unexpected consequences.

```
Class >> unsafeRenameInstVarNamed: oldName to: newName
```

See method comments in the image for more details.

## 3.2 Changes in Numeric/Time classes

### Added Specials

Specials are kinds of objects in which the value of the object is encoded in the OOP itself, and thus do not require an entry in the object table, space on a data page, nor a fetch to fault into memory.

The following new Specials are subclasses of existing classes, and, like SmallDouble and Float, transparently interoperate to represent the most commonly used subsection of the full range of the superclass's values.

#### SmallDate

SmallDates represents instances of Date; the SmallDate range covers all supported Dates.

Creating an instance of Date in the supported Date range will return an instance of SmallDate.

The technical requirement is that the year range is limited to  $-2^{31}$  (-2147483648) to  $2^{31} - 1$  (2147483647). The OOP uses 9 bits for an unsigned integer for the dayOfYear, and 47 bits for the 32 bit signed integer year.

#### SmallDateAndTime

A SmallDateAndTime is a DateAndTime representing a time (in microseconds) in the range of years 2001-2072, for times with offset from UTC (in hours) in the range -16..15).

Creating an instance of DateAndTime, when the result is in the range of SmallDateAndTime, will return an instance of SmallDateAndTime.

The OOP uses 5 bits signed offset from UTC in hours, and 51 bits of unsigned integer for the time since 2001 in units of microseconds.

#### SmallScaledDecimal

SmallScaledDecimal is used to represent ScaledDecimals that have scale  $\leq 31$  and mantissa in the range  $-2^{50}$  (-1125899906842624) to  $2^{50} - 1$  (1125899906842623).

Creating an instance of ScaledDecimal, when the result is in the range of SmallScaledDecimal, will return an instance of SmallScaledDecimal.

The OOP uses 51 bits for a signed integer mantissa and 5 bits unsigned scale.

#### SmallTime

SmallTime is used to represent instances of Time as the local time since midnight in units of microseconds.

Creating an instance of Time, when the result is in the range of SmallTime, will return an instance of SmallTime.

The OOP uses 56 bits for an unsigned integer. The maximum value of local time is 86399999999.

## DateAndTime changes

To retain common behavior between the range of `SmallDateAndTime` and `DateAndTime`, `DateAndTime` has been adjusted. The following changes describe the behavior of instances of `SmallDateAndTime` as well as instance of `DateAndTime`.

- ▶ `DateAndTime` instances are now invariant.
- ▶ The method `offset:`, which modified and returned the receiver, now returns a new instance of `DateAndTime` with the new offset. Please examine your code to ensure that you are not relying on `offset:` to modify the receiver; this method is ANSI and could not be deprecated or return an error. `SmallDateAndTime` instances, since the value is encoded in the OOP, are inherently not modifiable.
- ▶ The method `beRounded` modified the receiver, and will throw an invariant object error for `DateAndTime` and a `shouldNotImplement` for `SmallDateAndTime`. Code should use the method `rounded`, which returns a new instance.
- ▶ `DateAndTime` instances are always created with an internal `SmallScaledDecimal` with 6 decimal places of precision, equivalent to microseconds, for equivalence with `SmallDateAndTime`.
- ▶ `DateAndTime` `asSeconds` returns an instance of `SmallScaledDecimal`, not a `SmallDouble`.
- ▶ `DateAndTime` instances no longer print based on the nature of the data in the seconds instance variable. `DateAndTime` instances print using `printString` with microseconds (6 digits of fractional seconds), and as integral seconds using `asString`.
- ▶ The class method `nowWithScale:` is deprecated, and returns the same value as `now`. The `DefaultScale` class variable is obsolete and no longer used.
- ▶ Class and method comments for `DateAndTime` and `DateAndTimeANSI` have been edited to remove boilerplate and general information, making the useful information easier to see.
- ▶ Some instance creation errors now signal an `OutOfRange` error rather than an `Error`.

## Special care when adding methods to kernel classes `DateAndTime`, `Date`, `Time`, and `ScaledDecimal`

Application extension methods that are added to `DateAndTime`, `Date`, `Time`, or `ScaledDecimal` may directly reference instance variables on that class; this is normal, and these methods will compile correctly. However, these methods are problematic when inherited by `SmallDateAndTime`, `SmallDate`, `SmallTime`, and `SmallScaledDecimal`, since specials by definition do not have instance variables.

If you have added methods directly to GemStone kernel classes `DateAndTime`, `Date`, `Time`, or `ScaledDecimal`, you must review your code and ensure that any direct accesses to instance variables are changed to use accessor methods. Methods with direct instance variable references will compile correctly, but will get `Error 2710/instVar access not allowed in a special object on execution`. Methods with direct references that are not recompiled will cause the Gem to crash.



## Other added methods

`Time >> asMicroseconds`

Returns a `SmallInteger` that represents the receiver in units of microseconds since midnight, local time.

`Time >> subtractMilliseconds: anInteger`

Returns a `Time` or `SmallTime` (in practice, a `SmallTime`) that describes a time of day *anInteger* milliseconds earlier than that of the receiver.

`Time class >> fromMicroseconds: anInteger`

Create an instance of `Time` or `SmallTime` (in practice, a `SmallTime`) that is *anInteger* microseconds since midnight, local time.

## 3.3 ExternalSession Changes

The class `GsTsExternalSession` was added in v3.5.3. `GsTsExternalSession` uses the thread-safe GCI (`gcits.hf`), rather than the traditional GCI interface (`gci.hf`) that is used by `GsExternalSession`.

In this release, `GsTsExternalSession` has been associated with `GsExternalSession` under a common superclass, with adjustments to the API to improve the ability to move between these classes.

In addition, the class `GsTsX509ExternalSession` has been added, which allows X509-secured external sessions using the thread-safe GCI.

This is the new hierarchy:

```
AbstractExternalSession
  GsExternalSession
    GsLegacyExternalSession
    GsX509ExternalSession
  GsTsExternalSession
    GsTsX509ExternalSession
```

`GsExternalSession` is now legacy; new code should use `GsTsExternalSession`.

`GsExternalSession` may have issues in complicated environments with multiple threads.

The overhead of making calls using `GsTsExternalSession` is 1/2 to 1/4 the overhead required by calls to `GsExternalSession`.

### GsTsExternalSession changes

`GsTsExternalSession` is now a subclass of `AbstractExternalSession`, and a number of methods are now implemented in the superclass.

In addition, many new methods have been added to make the transition between `GsExternalSession` and `GsTsExternalSession` more transparent.

- ▶ You can now create an instance of `GsTsExternalSession` using `newDefault`, which creates an instance with the same library version as the current session.
- ▶ `GsTsExternalSession` has added instance variables `stoneSessionSerial` and `logger`.

- ▶ The logging feature from `GsExternalSession` is now available in `GsTsExternalSession`; however, the thread-safe API does not support user actions, so logging to the GCI client is not supported.
- ▶ The methods `quiet` and `quietLogout` are now implemented in `GsTsExternalSession`; however, they have no effect (`GsTsExternalSession` is always quiet).
- ▶ Methods to create a new instance, or update an existing instance, by setting specific parameter values have been added. The following methods are added:

```
GsTsExternalSession >> gemNRS:
GsTsExternalSession >> hostPassword:
GsTsExternalSession >> hostUsername:
GsTsExternalSession >> password:
GsTsExternalSession >> stoneNRS:
GsTsExternalSession >> username:
GsTsExternalSession class >> gemNRS:stoneNRS:username:password:
GsTsExternalSession class >> gemNRS:stoneNRS:username:aUsername:hostUsername: hostPassword:
```

- ▶ The following other methods have also been added:

```
hardBreak
isRemoteServerBigEndian
nbLogin
nbLoginFinished
softBreak
stoneSessionSerial
waitForResultForSeconds:
waitForResultForSeconds:otherwise:
```

## Methods added to both

The following methods have been added to both `GsExternalSession` and `GsTsExternalSession`:

```
begin
  Abort current transaction and begin a transaction

commitOrError
  Commit the current transaction in the remote session and return true, or signal an error if the commit fails.
```

## 3.4 Other Changes and Enhancements

### Behavior >> isSpecial should not be used; replaced by Behavior >> areInstancesSpecial

The method `Object >> isSpecial` can be sent to any object to determine if it is a special (the OOP encodes the value, such as a `Character` or `SmallInteger`). Historically, this method was overridden by `Behavior >> isSpecial`, which returned if instance of the class would be `Specials`, rather than the status of receiver itself.

The method `Behavior >> areInstancesSpecial` has been added, and should be used to test if instances of a class are `Specials`.

In v3.6, `Behavior >> isSpecial` has been removed, and the behavior is inherited now from `Object`. Sending `isSpecial` to Classes such as `Character`, which previously returned true, will now return false. Examine your application senders of `isSpecial` to ensure that this does not have unexpected results; change your code as needed to use `areInstancesSpecial`.

### Added commit failure types

After an attempt to commit a transaction fails, `System class >> transactionConflicts` returns the details of the commit failure. In addition to the previous possible return values, this may also now include `#symbolFailure` or `#lockFailure`.

### Exiting the Client from Smalltalk

The class `ExitClientError` has been added, allowing you to exit from a client process (usually `topaz`) with a specific error code.

```
topaz 1> run
ExitClientError
  signal: 'Disallowed Operation'
  status: 34
%
ERROR 3004 , a ExitClientError occurred (error 3004), ,
  Disallowed Operation (ExitClientError)
Logging out session 1.
--- 08/11/2020 15:21:17.137 PDT Logging out
```

### ClassOrganizer searches class comments, return arrays changed

String search in `ClassOrganizer` (and via `topaz`) now searches the text of class comments, as well as method source.

The Arrays returned from methods to perform search were previously of size 2, the methods containing the search string and the offsets into these methods, are now are of size 3. The new third element is a list of classes whose class comment includes the search string.

Affected methods are:

```
ClassOrganizer>>substringSearch:
ClassOrganizer>>substringSearch:in:
ClassOrganizer>>substringSearch:in:ignoreCase:
```

## 3.5 Deprecated and removed classes and methods

The following methods are newly deprecated:

```
Class >> byteSubclass:classVars:classInstVars:poolDictionaries:
  category:inDictionary:instancesInvariant:
Class >> indexableSubclass:instVarNames:classVars:classInstVars
  :poolDictionaries:category:inDictionary:instancesInvariant:i
  sModifiable:
Class >> subclass:instVarNames:classVars:classInstVars:poolDict
  ionaries:category:inDictionary:constraints:instancesInvariant:
  isModifiable:
DateAndTime class >> nowWithScale:
DateAndTime class >> setDefaultScale:
System class >> reclaimGemSessionIds
System class >> _deadNotReclaimed
```

### Removed Classes

The following classes were removed, and are now in Globals in the sub-dictionary ObsoleteClasses.

```
CanonicalStringDictionary
RubyBreakException
RubyCextData
RubyConstantRef
RubyHash
RubyProc
RubySocket
RubyThrowException
RubyTime
UnimplementedFloat1
UnimplementedFloat2
```

### Find All Reference Path removed methods

The following methods related to the previously deprecated, now removed find all reference paths feature have been removed:

```
Repository >> findAllReferencePathsToObject:
Repository >> findAllReferencePathsToObject:maxPaths:
Repository >> findAllReferencePathsToObjects:
Repository >> findAllReferencePathsToObjects:limitObjArray:prin
  tToLog:
Repository >> findAllReferencePathsToObjects:limitObjArray:prin
  tToLog:maxPaths:
Repository >> findAllReferencePathsToObjects:maxPaths:
Repository >> findAllReferencePathsToObjects:printToLog:
Repository >> findAllReferencePathsToObjects:printToLog:maxPath
  s:
Repository >> _findAllReferencePathsToObjects:limitObjArray:pri
  ntToLog:maxPaths:
```

## Other removed methods:

```
AbstractCharacter >> asJISCharacter
Behavior >> isSpecial
Character >> asJISCharacter
CharacterCollection >> asJISString
CPreprocessor >> defaultAix61Definitions
DecimalFloat class >> on:do:
GsObjectInventory class >> profileObjectsInBm:
GsObjectInventory class >> profileObjectsInFile:
Integer >> asJISCharacter
Object >> asEUCString
System class >> fastFindObjectsLargerThan:limit:
System class >> findObjectsLargerThan:limit:
```

## Private removed methods

The following private methods (including methods that were temporarily added during tonel code development), have been removed:

```
AbstractDictionary >> at:ifPresent:
BlockClosure >> setPackage:withInMetacelloConfig:
CharacterCollection >> tonelSubstrings:
Class >> _newKernelSubclass:instVarNames:inDictionary:
Class >> _nonInheritedOptions
Class >> _optionsForDefinition
CLibrary >> _oneArgPrim:arg:
CLibrary class >> _oneArgPrim:arg:
Collection >> asMetacelloAttributeList
Collection >> setForDo:withInMetacelloConfig:
Collection >> setIncludesInMetacelloPackage:
Collection >> setLoadsInMetacelloProject:
Collection >> setRequiresInMetacelloPackage:
Collection >> sorted:
CPreprocessor >> _defaultAix61DefinitionsOld
CZstream class >> _initializeConstants
DateAndTimeANSI >> _secondsUTC:offset:
Dictionary class >> newFromPairs:
GciLibrary >> GciHeapCheck
GciLibrary >> GciUtlExit_:
GciLibrary class >> initializeFunction_GciHeapCheck_inLibrary:
GsBitmap >> _basicNew
GsBitmap >> _reportString
GsNMethod >> _debugInfoHasFileAndLine
GsNMethod >> _lineNumberBias
GsObjectInventory class >> _objInventory:waitForLock:pageBufSize:percentCpuActiveLimit:showHiddenClasses:aHiddenSet:listInstances:toFile:inMemoryOnly:
GsQuery class >> initialize
GsQueryFormula class >> initialize
GsQueryTokenParser class >> initialize
IO class >> new:mode:
```

```
IO class >> _forgetFileDescriptor:
IO class >> _rememberFileDescriptor:obj:
Object >> flag:
Object class >> fixReferencesAfterConversionFromDirectory:
ProcessorScheduler >> _endYield
ProcessorScheduler >> _findReadyProcess:
ProcessorScheduler >> _reapEvents:honorOob:
ProcessorScheduler >> _reschedule:
ProcessorScheduler >> _rescheduleHonorOob:
Repository >> _allReferencesByParentClass:
Repository >> _fastAllReferencesByParentClass:
Repository >> _primRestoreSecureBackups:scavPercentFree:bufSize
:privateDecryptionKey:passphrase:numThreads:
Repository >> _restoreBackups:scavPercentFree:bufSize:numThread
s:
SequenceableCollection >> writeStream
String >> addToMetacelloRepositories:
String >> asObsoleteSymbol
String >> execute:against:
String >> fetchRequiredForMetacelloMCVersion:
String >> loadRequiredForMetacelloMCVersion:
String >> mergeIntoMetacelloRepositories:
String >> packageFileSpecFor:
String >> recordRequiredForMetacelloMCVersion:
String >> removeFromMetacelloRepositories:
String >> resolvePackageSpecsNamedForMetacelloMCVersion:visited
:ifAbsent:
String >> setIncludesInMetacelloPackage:
String >> setLoadsInMetacelloProject:
String >> setRequiresInMetacelloPackage:
Symbol >> asMetacelloAttributeList
Symbol >> asMetacelloAttributePath
Symbol >> setForDo:withInMetacelloConfig:
Symbol >> setForVersion:withInMetacelloConfig:
Symbol >> setPostLoadDoItInMetacelloSpec:
Symbol >> setPreLoadDoItInMetacelloSpec:
System class >> _breakSerialization
Time >> _initialize:
UserProfile >> _createDictionary:at:
UserProfile >> _renameDictionary:to:
```

# Changes in the GCI Interfaces

---

## 4.1 GCI changes

### Added Functions and Macros

The following functions and macros have been added to the standard GCI, `gci.hf`.

#### **GciCheckSigTerm**

```
(BoolType) GciCheckSigTerm(  
    );
```

#### **GciFetchDateAndTime**

```
(BoolType) GciFetchDateAndTime(  
    OopType dtObj,  
    GciDateAndTime *result  
    );
```

#### **GciFetchDateAndTimePosix**

```
(BoolType) GciFetchDateAndTimePosix(  
    OopType dtObj,  
    GciPosixDateAndTime *result  
    );
```

#### **GciNewDateAndTime**

```
(OopType) GciNewDateAndTime(  
    const GciDateAndTime *arg  
    );
```

**GciNewDateAndTimePosix**

```
(OopType) GciNewDateAndTimePosix(
    int64 microseconds,
    int offsetSeconds
);
```

**GCI\_OOP\_IS\_SmallDate**

```
BoolType GCI_OOP_IS_SmallDate(theOop)
```

**GCI\_OOP\_IS\_SmallDateAndTime**

```
BoolType GCI_OOP_IS_SmallDateAndTime(theOop)
```

**GCI\_OOP\_IS\_SMALL\_DOUBLE**

```
BoolType GCI_OOP_IS_SMALL_DOUBLE(theOop)
```

**GCI\_OOP\_IS\_SmallScaledDecimal**

```
BoolType GCI_OOP_IS_SmallScaledDecimal(theOop)
```

**GCI\_OOP\_IS\_SmallTime**

```
BoolType GCI_OOP_IS_SmallTime(theOop)
```

## 4.2 GCI thread-safe changes

The thread-safe GCI is provided in gcits.hf.

### Added Functions

**GciTsNbLogin**

```
(GciSession) GciTsNbLogin(
    const char *StoneNameNrs,
    const char *HostUserId,
    const char *HostPassword,
    BoolType hostPWIsEncrypted,
    const char *GemServiceNrs,
    const char *gemstoneUsername,
    const char *gemstonePassword,
    unsigned int loginFlags,
    int haltOnErrNum,
    int *loginPollSocket /* output */
);
```

Starts a GciTsLogin in a separate thread. You must call GciTsLoginFinished to determine if login is complete. While the login is in progress, any GciTs call other than GciTsLoginFinished will return an error.



While the login is executing, the file descriptor returned in `*loginPollSocket` can be polled, when read-ready the login is complete.

Returns NULL if `malloc()` failed or `socketpair()` failed.

### **GciTsNbLoginFinished**

```
(int) GciTsNbLoginFinished(  
    GciSession sess,  
    BoolType *executedSessionInit,  
    GciErrSType *err  
);
```

Should be called at least once after `GciTsNbLogin` is called. Argument `sess` must be result of a `GciTsNbLogin` call. Function result is:

- ▶ 0 if previous `GciTsNbLogin` not finished, and `executedSessionInit` is invalid
- ▶ -1 if previous `GciTsNbLogin` failed, with details in `*err`
- ▶ 1 if previous `GciTsNbLogin` finished; `*err` may include a warning

### **Removed Functions**

The following functions have been removed:

```
GciTsForkContinueWith  
GciTsForkExecute  
GciTsForkLogin  
GciTsForkPerform  
GciTsForkStoreTravDoTravRefs  
GciTsX509ForkedLogin
```

## **4.3 Foreign Function Interface (FFI) related changes**

### **Support for Callout symbol version**

You can now specify the symbol version for an FFI callout, to prevent accidentally calling the wrong version of the routine.

```
CCallout >> version
```

```
CCallout >> version: aString
```

The argument *aString* will be used as third argument to `dlvsym()`

### **Race condition in setting FFI errno**

Previously, the FFI `errno` was in session state, and could be accessed incorrectly if the `GsProcess` was preempted at the wrong time. Now, the FFI `errno` is part of the `GsProcess` state. This does not affect customer code. (#48524)

## Added methods

The following methods have been added:

```
CByteArray >> utf16From: zeroBasedStart to: zeroBasedEnd
    Return a new instance of Utf16 containing the specified bytes of the receiver.

CByteArray >> utf8From: zeroBasedStart to: zeroBasedEnd
    Return a new instance of Utf8 containing the specified bytes of the receiver.

CByteArray class >> withAll: anObject nullTerminate: aBoolean
    anObject may be a String, ByteArray, Utf8, Utf16 or CByteArray. Returns a new
    instance of the receiver containing a copy of the bytes of the argument. If aBoolean
    is true, the resulting CByteArray contains the bytes of the argument plus a
    terminator byte (or Utf16 codepoint) with value zero. If argument is a Utf16, the
    result has code points in CPU byte order.
    Note that ByteArray usually stores integers in big-endian order, not CPU native
    byte order.
```

## 4.4 UserAction changes

### Handling Errors in user actions

In earlier releases, the error handling block of an `on:do:` could not handle an error in a user action, and then return from the error handling block. This case resulted in infinite recursion signaling an `UncontinuableError`.(#49040)

An error from a user action now signals the new error 2758, `ERR_EXC_RETURN_DISALLOWED`: A return from `on:do:` would cross frame of C primitive, user action, or FFI call. This is handled correctly by the `on:do:.`

To allow debugging of the original error, a new Gem runtime parameter, `GemDebuggerActive`, allows you to retrieve a stack trace for the original error. When `GemDebuggerActive` is set to true, the `on:do:` does not handle error 2758, and control returns to the command line. At this point, in topaz, you can fetch the stack using commands such as **where**.

### Invoking a client useraction from non-blocking GCI call could freeze Gem

In an execution that was started from a `GciNbPerform*` or `GciNbExecute*` call, if the code invokes a client useraction, the gem would freeze until a further call such as `GciNbEnd`. (#48658)

User actions are now disallowed during non-blocking GCI execution. User actions have always been disallowed when using the thread safe GCI, `GciTs`.

The flag `GCI_PERFORM_NON_BLOCKING` was renamed to `GCI_PERFORM_noClientUseraction`.

## 4.5 Updated Compile and Link Information

### Linux Compile and Link Information

#### Compiler version

Red Hat Linux ES 8.1: gcc/g++ 8.3.1

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

Ubuntu 20.04 LTS: gcc/g++ 9.3.0

Ubuntu 18.04 LTS: gcc/g++ 8.4.0

Ubuntu 16.04 LTS: gcc/g++ 5.4.0

#### Debugger version

Red Hat Linux ES 8.1: gdb 8.2-11

Red Hat Linux ES 7.4: gdb 7.6.1-100

Ubuntu 20.04 LTS: gdb 9.1

Ubuntu 18.04 LTS: gdb 8.2.50

Ubuntu 16.04 LTS: gdb 8.2.50

#### 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 -Wno-format-truncation
-Wuninitialized
```

#### Linking a user action library

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

#### Linking a GCI application

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

## Solaris on x86 Compile and Link Information

### Compiler version

CC: Studio 12.5 Sun C++ 5.14 SunOS\_i386 2016/05/31

### Debugger version

Sun DBX Debugger 8.1 SunOS\_i386 2016/06/01

### 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 -h libuserAct.so -i userCode.o
$GEMSTONE/lib/gciualib.o -o libuserAct.so -Bdynamic -lc
-lpthread -ldl -lrt -lsocket -lnsl -lm -lpam -lCrun -lsendfile
-z nodefs
```

### Linking a GCI application

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

## AIX Compile and Link Information

### Compiler version

IBM XL C/C++ for AIX, V16.1

### 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
-029:1540-1103:1540-2907:1540-0804:1540-1281:1540-1090:1540-2988
-qnoeh -I$GEMSTONE/include -c userCode.c -o userCode.o
```

Depending on your version of AIX, you need to include `-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 -LcompilerInstallDir/lib
-lpthread -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/compilerInstallDir/lib -lpthread -lc_r -lC_r -lm -ldl
-lbsd -lpam -Wl,-brtllib -o userAppl
```

## DARWIN Compile and Link Information

### Compiler version

Apple clang version 11.0.3 (clang-1103.0.32.62)

### Debugger version

lldb

### 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

### Compiler/Debugger version

Microsoft Visual Studio 2010 Version 10.0.30319.1 RTMRel

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

### Compiling a GCI application

```
c1 /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
```

---

The following bugs were present in v3.5.4 and are fixed in this version.

## Slow stone startup with large possibleDead

If the stone is shut down after the `markForCollection`, but before `promoteToDead`, and the `possibleDead` set is very large, the restart of the stone may take a long time (30 minutes was observed with a 4 billion size `possibleDead`). (#49109)

## Socket blocking issues

### Session waiting in `GsSocket >> accept` does not respond to `stopSession`

When a session is waiting on a `GsSocket >> accept`, it did not respond to a `stopSession` out-of-band signal. (#49120)

### `GsSocket read` or `accept` may block after `Socket` closed

A blocking read or accept operation on a `GsSocket` may continue to block, even if the socket itself is closed. (#49041, #49081)

### `readWillNotBlockWithin:` and `writeWillNotBlockWithin:` result could be incorrect

The methods `GsSocket>>readWillNotBlockWithin:` and `writeWillNotBlockWithin:` could return true when the socket was not ready, so the subsequent read or write would block. (#49044)

## Hot standby bugs

### logsender did not respect not-default `STN_TRAN_LOG_PREFIX`

The logsender did not work correctly if the transaction logs generated by the master stone did not have the default name `tranlogNN.dbf`; that is, if the setting for `STN_TRAN_LOG_PREFIX` was not left at the default. (#48706)

### **logsender login failures caused excessive error messages**

If a running logsender attempts to login to a restarted stone, and the login fails (such as incompatible Gem version), there was excessive printing to the stone log, as well as to the logsender log. Now, only a single login failure message is printed to the Stone log, and logsender prints messages less frequently. (#48752, 48957)

### **Repository>>continuousRestoreFromArchiveLogs: did not work on extent copy**

When a hotstandby system was started using an extent copy backup, the `continuousRestoreFromArchiveLogs`: did not error, but it failed to replay tranlogs. (#48340)

### **startlogreceiver did not expand environment variables in -T argument**

If the argument to `-T` included an environment variable, this was not expanded. (#49057)

## **CHep memory leaks in failed X509 logins**

Linked topaz and HostAgents may have excessive CHep memory use after failed X509 logins. (#49230)

## **Restore from tranlogs may not replay clustering**

When a clustering operation in a tranlog is being replayed, and the object was already modified by another tranlog replay and is not yet committed, clustering of that object to a different cluster bucket may be ignored, if the new cluster bucket does not relate to the object's current page. (#48509)

## **Repository scan may fault in dead objects**

When a repository scan operation is running while sessions are voting on dead objects, there is a small risk that objects that were faulted into memory that are being promoted to dead. This could potentially allow references to dead objects to be committed. (#49051) Now, this will cause a repository scan to error with "An atomic promote detected during operation, results are compromised - please try again".

## **DBF\_PRE\_GROW did not accept list with empty elements**

Configuration parameters that take lists should allow empty elements (that is, two comma dividers with nothing or white space between them). This was not working for `DBF_PRE_GROW`. (48967)

## **Stone may not be responsive while disposing large CR backlog**

When there is a large commit record backlog to dispose, the Stone may be unresponsive until the backlog is completely disposed. (#48755)



## Out of Memory in code\_gen space with many classes

The thresholds that managed garbage collection in the code\_gen space of temporary object memory were not configured correctly, so cases that placed pressure on code\_gen could encounter OutOfMemory errors. (#49143)

## Error reporting for Stone not found

When the Stone named in topaz login parameters is not found, the error message returned to topaz included a number of secondary and duplicate errors. (#49032)

## GsHostProcess executable-not-found error not helpful

GsHostProcess requires a full path to the executable, it does not do a path lookup. The error message when the path was omitted was not helpful in understanding the problem. (#48761).

## Statmonitor issues

### Deleting a statmonitor file in use may corrupt the new file

If the file to which statmonitor is writing is deleted, statmonitor opens a new file and continues writing. However it may write the buffered data to the new file before it writes the statmonitor header; which means VSD cannot read the file. (#49233)

### In auto-restart mode, file name may be reused

When running with **statmonitor -r** or **-R**, which automatically create new files periodically, a new filename is generated containing an incremented sequence number. However, if the current file is moved or deleted, a file with that same sequence number was generated, rather than the next in the logical sequence. (#49235)

## Topaz issues

### Possible to redefine the ErrorCount predefined variable

Topaz includes a number of predefined variables, which should not be redefined. The ErrorCount variable was not previously disallowed in **define**, which could cause unexpected behavior; for example, from **errorcount** or **exit/quit**. (#48977)

### Display bytes with multi-byte characters did not linebreak correctly

When executing with display bytes, the output for byte objects such as Strings include both the text, broken up into 20-byte lines, and the corresponding bytes. When displaying multi-byte characters, the line breaks were inserted off by one. (#49190)

## Time and Date related issues

### **TimeZone fromPath: attempted to look for zoneinfo on client**

The code in `TimeZone fromPath:` first looks for zoneinfo on the given path on the server; previously, if this was not found, it would attempt to find the given path on the client. Now, it searches only on the server. (#49009)

### **Duration printing of subseconds did not round up to second**

Duration printing prints in milliseconds resolution. If this printing would normally round up to a full seconds (0.9999, for example), this did not increment the whole second, but left the subseconds as 0. Rounding up in now capped at .999. (#48802)

### **DateAndTime fromString: without timezone was incorrect**

`DateAndTime fromString:` produced an oddly incorrect time, if the timezone portion of the argument was omitted. Note that `fromString:` is not expected to work with poorly formatted input strings. (#48804).

## **GciTsLibrary version: did not handle argument flexibly**

The argument to `GciTsLibrary >> version:` was limited to specific versions, and failed to match some legal version numbers. (#48879)

## **Removed reference to JISCharacter in RangeEqualityIndex>>isBasicClass:**

The source code for the image method `RangeEqualityIndex>>isBasicClass:` contained a reference to the class `JISCharacter`, which was moved to `ObsoleteClasses` in an earlier version; this method could not be recompiled. The reference has been removed. (#48343)

## **GemStoneParameters did not handle invariant password string**

If the `password:` argument to an instances of a `GemStoneParameters` was invariant (such as a literal string), an error was thrown. (#48898)