

---

*GemStone*®

# **GemStone/S 64 Bit™**

## **Release Notes**

**Version 3.3.1**

June 2016



## **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.

This documentation, or any part of it, may not be reproduced, displayed, photocopied, transmitted, or otherwise copied in any form or by any means now known or later developed, such as electronic, optical, or mechanical means, without express written authorization from GemTalk Systems.

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 © 1986-2016 GemTalk Systems LLC. All rights reserved by GemTalk Systems.

## **PATENTS**

GemStone software is covered by U.S. Patent Number 6,256,637 "Transactional virtual machine architecture", Patent Number 6,360,219 "Object queues with concurrent updating", Patent Number 6,567,905 "Generational garbage collector with persistent object cache", and Patent Number 6,681,226 "Selective pessimistic locking for a concurrently updateable database". GemStone software may also be covered by one or more pending United States patent applications.

## **TRADEMARKS**

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

**VMware** is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions.

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

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

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

**Microsoft**, **MS**, **Windows**, **Windows 7**, **Windows 2008**, and **Windows 8** are registered trademarks of Microsoft Corporation in the United States and other countries.

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

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

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

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

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

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

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

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

---



# Preface

---

## About This Documentation

These release notes describe changes in the GemStone/S 64 Bit™ version 3.3.1 release. Read these release notes carefully before you begin installation, conversion testing, 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 3.3.1.

These documents are available on the GemTalk website, as described below.

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

This material is updated regularly; we recommend checking this site on a regular basis.

## Help Requests

You may need to contact Technical Support directly, if your questions are not answered in the documentation or by other material on the Technical Support site. Technical Support is available to customers with current support contracts.

Requests for technical assistance may be submitted online, by email, or by telephone. We recommend you use telephone contact only for more serious 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

When submitting a request, please include the following information:

- ▶ Your name and company name.
- ▶ The versions of GemStone/S 64 Bit and of all related GemTalk products, and of any other related products, such as client Smalltalk products.
- ▶ The operating system and version you are using.
- ▶ A description of the problem or request.
- ▶ Exact error message(s) received, if any, including log files 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. GemStone/S 64 Bit 3.3.1 Release Notes*

Overview . . . . .	7
Upgrade . . . . .	7
New required step to configuring ICU version . . . . .	7
Supported Platforms . . . . .	8
Platforms for Version 3.3.1 . . . . .	8
GemBuilder for Smalltalk (GBS) Versions . . . . .	8
VSD Version. . . . .	9
Changes and New Features . . . . .	9
Updated SSL libraries . . . . .	9
Community Edition Keyfile for Linux and Darwin . . . . .	9
Handling of ICU shared libraries. . . . .	9
Distribution changes . . . . .	10
Image changes. . . . .	10
Library load changes . . . . .	10
Added Configuration Parameter . . . . .	11
Other changes related to sorting and index-related changes . . . . .	11
Improved support for SortBlock recompile . . . . .	11
ExecBlock recompile . . . . .	11
sortWithBlock:* methods accepting IcuCollator. . . . .	12
replaceFrom:to:with:startingAt: now usable with CByteArray argument . . . . .	12
Increased limit for GsSocket read:into:startingAt: . . . . .	12
Additional printing for GCI errors . . . . .	12
Change in errors . . . . .	12
Handling C-level errno from CCallout. . . . .	12
Override limit on VM memory use. . . . .	13
Utility Changes . . . . .	13
topaz output pushnew limit increased . . . . .	13
startnetldi added option to skip reverse DNS lookups. . . . .	13

printlogs utility allows suppression of displaying strings . . . . .	13
gslst "exe deleted" status . . . . .	13
Added Cache Statistic. . . . .	13
GCI Changes. . . . .	14
GciPerformFetchBytes . . . . .	14
GciNbPerformFetchBytes. . . . .	14
GCI compilation flag . . . . .	14
Bugs Fixed . . . . .	15
Oop High Water Mark over 4 billion created problems . . . . .	15
Handling of reused PIDs in defunct lock files . . . . .	15
Infinite loop if gemstone.hostid exists but is not readable . . . . .	15
NetLDI may hang on EAI_AGAIN from reverse lookup. . . . .	15
STN_LOGIN_LOG_DIR did not expand environment variables . . . . .	15
Logins on Mac required locks directory on client node. . . . .	16
Issues related to Upgrade . . . . .	16
ICU version change caused problems in upgrade to 3.3 . . . . .	16
Upgrade to 3.3 did not handle SortedCollection recompile requirement . . . . .	16
Upgrade did not correctly handle epochGcPageBufferSize. . . . .	16
upgradeSeasideImage failed as a result of recently added class . . . . .	16
Corrupt object error during symbol conversion from 2.4.x. . . . .	16
DbfHistory extra line for 3.2.x-originated repositories . . . . .	16
User action load problems corrupted installed user action table . . . . .	16
Garbage Collection issues . . . . .	17
Class instance referenced from class prevented GC . . . . .	17
Gem crash due to excessive recursion in MFC . . . . .	17
Reclaim Gem may startup in transaction . . . . .	17
More information in cases where Admin Gem is not running . . . . .	17
Backup and Restore issues . . . . .	17
Repeated replay of transaction log can cause corruption . . . . .	17
Possible deadlock in multithreaded backup after file error. . . . .	17
.gz not appended to short backup file names . . . . .	17
Indexing bugs . . . . .	18
GsQuery reject: incorrect results . . . . .	18
GsQuery detect:ifNone: returned query instead of ifNone: result . . . . .	18
ReversedRangeIndexReadStream size incorrect . . . . .	18
Index Audit could miss invalid encryption keys. . . . .	18
Remote shared cache related issues . . . . .	18
Remote shared page cache monitor can get stuck spin lock . . . . .	18
After remote cache startup timeout, cache status may be stuck . . . . .	18
Page Manager crash on removing pages from remote cache . . . . .	18
GsFile write following a read may write at end of file . . . . .	18
GsSecureSocket>>readWillNotBlockWithin: wrong after partial read . . . . .	18
Large operations on hidden sets can run out of ME object memory space . . . . .	19
Configuration Parameter Changes . . . . .	19
KeyValueDictionary become: did not correctly reset parent references . . . . .	19
Invalid class instance variables does not prevent class creation . . . . .	19
GsMethod did not include environmentId method . . . . .	19

# GemStone/S 64 Bit

## 3.3.1 Release Notes

---

### Overview

GemStone/S 64 Bit 3.3.1 is a new version of the GemStone/S 64 Bit object server. This release adds a number of new features, and fixes a number of bugs in v3.3; we recommend everyone using or planning to use GemStone/S 64 Bit upgrade to this new version.

These release notes describe changes between the previous version of GemStone/S 64 Bit, version 3.3, and version 3.3.1. If you are upgrading from a version prior to 3.3, review the release notes for each intermediate release to see the full set of changes. In particular, if you are upgrading from version 2.4.x, note that there were substantial changes in v3.0 that impact your application.

For details about installing GemStone/S 64 Bit 3.3.1 or upgrading from earlier versions of GemStone/S 64 Bit, see the *GemStone/S 64 Bit Installation Guide* for v3.3.1 for your platform.

### Upgrade

Upgrade is supported from GemStone/S 64 Bit v2.4.x, 3.1.x, 3.2.x, and 3.3. For details on the conversion and upgrade, see the *GemStone/S 64 Bit Installation Guide* for v3.3.1.

### New required step to configuring ICU version

GemStone/S 64 Bit v3.3 included an updated version of the ICU libraries that support unicode string collation and Unicode Comparison Mode. If you are upgrading from a version 3.2.x, you may continue to use the older ICU library version; an additional step is required when upgrading to set your application to use the legacy ICU version.

The underlying issue and the changes in v3.3.1 are described below, under “ICU version change caused problems in upgrade to 3.3” on page 16 and “Handling of ICU shared libraries” on page 9. The updated *Installation Guides* for v3.3.1 provides the specific steps that are required as part of upgrade.

## Supported Platforms

### Platforms for Version 3.3.1

GemStone/S 64 Bit version 3.3.1 is supported on the following platforms:

- ▶ Solaris 10 and 11 on SPARC
- ▶ Solaris 10 on x86
- ▶ AIX 6.1 on POWER6 and POWER7 and AIX 7.1 on POWER7 and POWER8
- ▶ Red Hat Enterprise Linux Server 6.4, 6.5, 6.7, and 7.1; Ubuntu 12.04 and 14.04; and SUSE Linux Enterprise 12, all on x86
- ▶ OS X 10.9.5 (Mavericks), with Darwin 13.4.0 kernel, on x86

For more information and detailed requirements for each supported platforms, please refer to the GemStone/S 64 Bit v3.3.1 Installation Guide for that platform.

### GemBuilder for Smalltalk (GBS) Versions

GemStone/S 64 Bit version 3.3.1 requires GBS version 8.1 or later for VisualWorks Smalltalk, or version 5.4.3 or later for VA Smalltalk. Older versions of GBS cannot login to v3.3.1.

The following versions of GBS are supported with GemStone/S 64 Bit version 3.3.1:

#### GBS version 8.1

<b>VisualWorks 7.10.1 32-bit and 64-bit</b>
<ul style="list-style-type: none"> <li>▶ Windows 8, Windows 2008 R2 and Windows 7</li> <li>▶ Solaris 10 on SPARC</li> <li>▶ RedHat ES 6.4, 6.5, 6.7, and 7.1, SUSE ES 12, and Ubuntu 12.04 and 14.04</li> </ul>

#### GBS version 5.4.3

<b>VA Smalltalk 8.6.2</b>	<b>VA Smalltalk 8.5.2</b>
<ul style="list-style-type: none"> <li>▶ Windows 8, Professional or above</li> <li>▶ Windows 2008 R2</li> <li>▶ Windows 7, Professional or above</li> </ul>	<ul style="list-style-type: none"> <li>▶ Windows 2008 R2</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 v3.3.1 distribution includes VSD version 5.2. The previous version of GemStone/S 64 Bit, v3.3, included VSD v5.1.3.

VSD v5.2 includes many changes and bug fixes, especially in support for working with multiple files. For a description of the changes, see the [Release Notes for VSD v5.2](#).

## Changes and New Features

This listing includes enhancements between v3.3 and v3.3.1, and includes changes also introduced in versions 3.2.13, 3.2.14, or 3.2.15.

### Updated SSL libraries

The version of OpenSSL used by GemStone/S 64 Bit v3.3.1 has been updated to 1.0.2h.

### Community Edition Keyfile for Linux and Darwin

For some time, GemStone distributions on certain platforms have included a limited starter keyfile, intended primarily for use with Seaside/GLASS/GsDevKit applications and therefore located under the `$GEMSTONE/seaside` directory. These keyfiles are suitable for evaluation.

In v3.3.1, on Linux and Darwin, the GemStone distribution includes a starter keyfile under `$GEMSTONE/sys`, with the name `community.starter.key`. If there is no keyfile at the location `$GEMSTONE/sys/gemstone.key`, then the stone will look for a keyfile `$GEMSTONE/sys/community.starter.key` and use that for startup.

If the configuration parameter `KEYFILE` refers to an invalid file, or a file other than `$GEMSTONE/sys/gemstone.key` that does not exist, then the stone will not use the starter keyfile and will fail to start.

Note that the starter keyfile does not have permissions for GBS, GemConnect, or GBJ; if you set up a new installation and do not set your license keyfile, the stone will now start rather than failing with an error. Operations such as attempting to login from GBS, for example, will fail.

The keyfile that is distributed under the seaside directory continues to be included in the distribution in the old location on Linux and Macintosh, in addition to the new community keyfile.

### Handling of ICU shared libraries

New versions of ICU are released periodically to support changes in the official Unicode standard. For example, the ruble symbol (Character codePoint: 8381) is a new addition in Unicode 7.0, and was added to ICU in v53. The ICU library version included with GemStone/S 64 Bit v3.3 was ICU v54.1, while version 3.2.x included version 51.2. This created problems with SortedCollections and Indexes that involved ICU-based comparisons; see the description on page 16 for details.

To ensure correct sort ordering and query results, the upgrade process and internal configurations have been changed so that repositories will continue to use the same ICU library version over upgrade. To avoid inadvertent changes, it is now disallowed to login with an ICU library version other than the one expected by the repository.

## Upgrade process and results

The ICU version varies between the supported originating repositories for upgrade to 3.3.1, and 2.x and 3.0.x did not have the ICU feature. The upgrade scripts and *Installation Guide* instructions have been updated for each upgrade path.

The result of following these instructions for each version is as follows:

- ▶ Upgrade from 2.x or 3.0.x will use ICU v54.1.
- ▶ Upgrade from 3.1.x will use ICU v54.1; you must resort/rebuild for ICU collation, since 3.1.x used an older ICU version that is not supported for 3.3.x.
- ▶ Upgrade from 3.2.x will use ICU v51.2, you must edit the configuration file setting for `GS_LIB_ICU_VERSION`, but no resort/rebuild for ICU collation is required.
- ▶ Upgrade from 3.3 will use ICU v54.1. If you have previously upgraded to 3.3 from 3.2.x, and you have not already rebuilt/resorted for ICU collation, you are exposed to bug #46056 and will need to perform the rebuild/resorts.

GsDevKit\_home systems that are upgraded using the upgradeStone script do not have to take any further action; this script performs all necessary steps on all supported originating versions.

## Distribution changes

Both 51.2 and 54.1 ICU libraries are included in the distribution, and the library interfaces are designed so that Gems in version 3.2.14 or later 3.2.x versions, or version 3.3.1, may load either version.

## Image changes

A class variable and access method has been added to IcuCollator to request the version of the ICU libraries that are currently loaded:

```
IcuCollator class >> libraryVersion
```

A new entry has been added to Globals to store the version of the ICU libraries that the repository should be using:

```
Globals at: #IcuLibraryVersion
```

This is a string that is set to '54.1' or '51.2' on upgrade, depending on what version you are upgrading from.

## Library load changes

The new configuration parameter `STN_GEM_LIBICU_VERSION` has been added, and is used by the Gem during login to determine which version of `libcicu*.so` should be loaded. This configuration parameter is not usable on AIX due to the way libraries are loaded, and you must use the following environment variable.

The default setting for this is 54.1. **An additional manual step is required on upgrade from 3.2.x to set this configuration parameter to 51.2 in the Stone's configuration file.**

The new `GS_LIB_ICU_VERSION` environment variable can override the setting for the configuration parameter. This may be set to either '51.2' or '54.1'.

During login, if the loaded ICU library does not match the repository's configuration, if `IcuCollator libraryVersion ~= (Globals at: #IcuLibraryVersion)`, then

an error is signalled and further commits are disallowed by this session. This avoids any risk of committing incorrectly sorted results.

The version test on login can be bypassed for upgrades, if the user is SystemUser and the environment variable `GS_LIB_ICU_BYPASS` is defined.

After subsequent upgrades, the repository will automatically continue to use the same ICU version; you must manually upgrade to use a later ICU version.

## Added Configuration Parameter

### **STN\_GEM\_LIBICU\_VERSION**

A string specifying the version of the libicu\* libraries that a gem or topaz -l should load during login. This can be overridden by the `GS_LIB_ICU_VERSION` environment variable in the environment of a gem (i.e. environment of the startnetldi) or the environment of a topaz -l process.

Not changeable at runtime.

Runtime equivalent in gem (read-only): `#GemLibicuVersion`  
Default: "54.1"

## Other changes related to sorting and index-related changes

The method `SortedCollection class >> resortInstancesFromFilesForGem: gemNum of: totalGems` has been added. This method reads the files in `$upgradeLogDir/scbm` that contain OOPs of instances of `SortedCollection`, and performs a resort.

Code has been added to `IndexManager` to support rebuild of indexes that rely on ICU library collation.

This code is designed for use in future improved index management during upgrade and ICU version upgrade.

## Improved support for SortBlock recompile

When application code recompile is required, as with the 3.2.x to 3.3.x upgrade, `sortBlocks` must also be recompiled; see bug on page 16. `postconv` is now a required step following the 3.2.x to 3.3.x upgrade, and support for this has been improved in v3.3.1.

### **ExecBlock recompile**

The following methods have been added:

`needsRecompile`  
Returns true if the receiver's block needs recompilation, false otherwise.

`recompile`  
Attempts to recompile the receiver's method, assuming it is a simple block without references to self or method variables.

These methods test if block recompile is needed, and perform the simple block recompiles. They are invoked by `postconv` after a GemStone server upgrade that requires code recompile.

Complex blocks that have references to self or method variables cannot be automatically compiled, since the referents are in other methods. Handling such blocks requires individual attention; you may have to manually construct code that recreates the block.

### **sortWithBlock:\* methods accepting IcuCollator**

The `sortWithBlock:*` methods now provide variants with an added `collator:` argument. This allows you to pass in the `IcuCollator` as a third argument in a sort block. This allows `IcuSortedCollection` sortBlocks to be simple blocks, which can be automatically recompiled.

The class `SortBlockUnicodeNode` has been added to support this functionality. This is a subclass of `SortBlockNode`, used by `BlockSorter` to implement the sort used by `sortWithBlock:*`.

### **replaceFrom:to:with:startingAt: now usable with CByteArray argument**

The methods

```
ByteArray >> replaceFrom:to:with:startingAt:
String >> replaceFrom:to:with:startingAt:
```

now permit the `with:` argument to be a `CByteArray`. Note that the `startingAt:` argument is a 1-based offset into the otherwise 0-based instance of `CByteArray`.

### **Increased limit for GsSocket read:into:startingAt:**

The read limit for the method `GsSocket >> read:into:startingAt:` was previously 10000000 (10MB), now it is 500000000 (500MB)

### **Additional printing for GCI errors**

When a GCI protocol error occurs, GemStone will now print the recent GCI calls and associated packets to allow diagnosis.

#### **Change in errors**

Error number 2230 was previously `LGC_ERR_PACKET_KIND_BAD`; this is now `LGC_ERR_SERVER_PACKET_KIND`.

Error number 2231 was previously `LGC_ERR_EXPECTED_CONTINUE`; this is now `LGC_ERR_CLIENT_PACKET_KIND`.

### **Handling C-level errno from CCallout**

The way `CCallout` handles the C global `errno` has been changed in this release.

Now, the `errno` resulting from a C-level call via `CCallout` is cached in VM memory. Each time that `CCallout >> callWith:` is invoked, `errno` is set to the cached value before the function, and the cached value is set to `errno` after the function executes. This preserves the value against other GemStone internal calls.

`CCallout` has added class methods to access the cached value:

```
CCallout class >> errno
```

Returns the `SmallInteger` value of the C global variable `errno` that was saved by the most recent invocation of `CCallout >> callWith:.`

```
CCallout class >> errno: aSmallInteger  
  If aSmallInteger == -1, Returns the SmallInteger value of the cached errno value,  
  ffiErrno. If aSmallInteger >= 0, stores aSmallInteger into the session's cache.
```

## Override limit on VM memory use

GemStone automatically sets the limits on memory use using `setrlimit()`.

Now, this can be disabled using the environment variable `GEMSTONE_NO_SETRLIMIT`, for customers who want to configure memory limits manually.

## Utility Changes

### topaz output pushnew limit increased

Previously, up to 100 files could be specified; now, 1000 are allowed.

### startnetldi added option to skip reverse DNS lookups

The following option has been added to `startnetldi`:

- N Use numeric IP addresses for printing peer info in logs, do not do reverse DNS lookups to get hostnames of clients.

### printlogs utility allows suppression of displaying strings

The output of the `printlogs` utility is used for debugging and for analysis of repository transaction. When printing page information, the output can include data incidental to the actual issue being examined, in the form of strings that may include sensitive information.

Now, the added `nostrings` argument to `printlogs` suppresses printing the contents of all kinds of strings and similar byte collections, including ByteArrays.

### gslis "exe deleted" status

When a GemStone distribution is replaced while `netldi`, `stone`, or `cache` are running, `gslis` described these as killed; now it reports a new status "exe deleted".

## Added Cache Statistic

The following statistic has been added:

### `AlmostOutOfMemoryCount` (Gem)

`AlmostOutOfMemoryCount` is the number of times that either an `AlmostOutOfMemory` notification was signalled to the application, or `smalltalk stack` was printed to the gem log because temporary object memory was approaching full.

## GCI Changes

The following functions have been added:

### GciPerformFetchBytes

GciPerformFetchBytes is similar to GciPerform, with the addition of maxResultSize, the maximum number of bytes to be returned in \*result . The function returns the number of bytes returned in \*result, or -1 if an error is available to be fetched with GciErr.

```
(int64) GciPerformFetchBytes(  
    OopType receiver,  
    const char *selector,  
    const OopType *args,  
    int numArgs,  
    ByteType *result,  
    ssize_t maxResultSize  
);
```

### GciNbPerformFetchBytes

GciNbPerformFetchBytes is a nonblocking variant of GciPerformFetchBytes.

```
(void) GciNbPerformFetchBytes(  
    OopType receiver,  
    const char *selector,  
    const OopType *args,  
    int numArgs,  
    ByteType *result,  
    ssize_t maxResultSize  
);
```

### GCI compilation flag

Compile flags in gciemn.h now include GCI\_COMPILE\_evaluateWithoutSelf = 4, which disallows references to self, used to recompile ExecBlocks.

## Bugs Fixed

The following bugs in 3.3 are fixed in v3.3.1. Note that this list includes bugs that are also fixed in 3.2.13, 3.2.14, and 3.2.15.

### Oop High Water Mark over 4 billion created problems

In an internal C function, a variable was declared as uint32 while a uint64 is needed. This may have caused failures or infinite loops in operations such as markForCollection, object audit, and reclaim. (#46252)

### Handling of reused PIDs in defunct lock files

gslis looks at GemStone lock (..LCK) files to determine what GemStone server processes (Stones, NetLDIs, etc.) are running. The lock files include information about the server process, including the PID of the process.

In earlier versions, gslis only verified that the PID was running, not that the process using that PID was the same GemStone server process. If a process was killed, leaving the lock file behind, and that PID was reused, gslis assumed the PID was valid. The server name/lock file were stuck, and the lock file had to be manually deleted.

Now, on Linux only, lock files involving reused PIDs are handled correctly, by checking in `/proc/PID/`.

- ▶ gslis always checks for the correct process pid, and will report such servers as killed, not OK.
- ▶ `gslis -c` will clear the lock file
- ▶ `startnetldi` will clear the lock file and start the requested netldi.

On platforms other than Linux, you must still manually delete the lock file.  
(#46066)

### Infinite loop if gemstone.hostid exists but is not readable

Each host with a GemStone server node finds or creates a file `/opt/gemstone/locks/gemstone.hostid` to identify the host. If this file exists, but is not readable by a process such as linked topaz, the process will hang. (#46219)

### NetLDI may hang on EAI\_AGAIN from reverse lookup

If a reverse IP lookup returned an EAI\_AGAIN ("The name could not be resolved at this time. Try again later."), the NetLDI retried immediately until a valid response was returned. On systems that persistently return this error, it resulted in the NetLDI hanging. (#46156)

An option has been added to `startnetldi` to skip the reverse lookup and print the numeric IP address; see "startnetldi added option to skip reverse DNS lookups" on page 13.

### STN\_LOGIN\_LOG\_DIR did not expand environment variables

The environment variable `STN_LOGIN_LOG_DIR`, added in version 3.2.2, did not permit environment variables such as `$GEMSTONE`. (#46028)

## Logins on Mac required locks directory on client node

A login from a client on Mac to a server on another machine failed if there was no `/opt/gemstone/locks` directory on the client. (#46181)

## Issues related to Upgrade

### ICU version change caused problems in upgrade to 3.3

The ICU library version was updated for v3.3. This created issues with indexes and SortedCollections that used the ICU libraries for collation. (#46056).

The way ICU versioning is handle is changed substantially in this release, see “Handling of ICU shared libraries” on page 9.

### Upgrade to 3.3 did not handle SortedCollection recompile requirement

After upgrade to v3.3, all code requires recompile. This includes code in SortedCollection sortBlocks and other persistent blocks. The upgrade instructions did not include this requirement. (#46055)

The Installation Guides have been updated for v3.3.1, and the code that supports postConv now recompiles simple sortBlocks from 3.2.x.

### Upgrade did not correctly handle epochGcPageBufferSize

When upgrading from v2.x to 3.x, the setting for epochGcPageBufferSize was not being converted from a previous setting to a power of two. (#46009)

### upgradeSeasideImage failed as a result of recently added class

After additions of GLASS/Seaside/GsDevKit classes GsFailedMethodCompilationDuringCopyMethodToNewClass, and NumberParserGsDevKitIssue75Test, the upgradeSeasideImage step failed. (#46059, #46217)

### Corrupt object error during symbol conversion from 2.4.x

During the symbol conversion phase of the 2.4.x to 3.x conversion, a data page could end up with an out of range clusterId. This resulted in a corrupt object error. (#46032)

### DbfHistory extra line for 3.2.x-originated repositories

In repositories originating in 3.2 and later, the DbfHistory included lines both for a 3.2.0 filein and a filein specific to the actual originating version. (#45977)

## User action load problems corrupted installed user action table

If a problem was detected while a user action was being loaded, the user action table was left in an invalid state. This may have caused the Gem to crash when the system attempted to execute a user action. (#46131).



## Garbage Collection issues

### Class instance referenced from class prevented GC

If a application class's class variable referenced an instance of the class, but there are no other references to the class or any instance of the class, and the size of the instance is 0, the class may have been immune from garbage collection. (#46116)

### Gem crash due to excessive recursion in MFC

There was an optimization in the multithreaded MFC that performed a recursion; it was possible for excessive recursion to occur which resulted in running out of Cstack space. This optimization was determined to provide very little value and has been removed. (#46194)

### Reclaim Gem may startup in transaction

When the stone starts up and has to recover due to an unclean shutdown, and there were pages to reclaim, then the Reclaim Gem started up in transaction. This could cause a commit record backlog until recovery completed, or Reclaim Gems' maxTransactionDuration was reached. (#46005)

### More information in cases where Admin Gem is not running

If the Admin Gem is not running following an operation that holds the GC lock, a request for a markForCollection cannot get the gcLock and fails to start with an error. The error did not indicate the cause of the problem; now, the error message includes a warning that the Admin Gem is not running. (#46047)

## Backup and Restore issues

### Repeated replay of transaction log can cause corruption

If an incomplete transaction log was replayed, and then later the same transaction log was replayed after it had completed (and therefore contained more records), it was possible for some records in the later part of the transaction log to not be replayed. This resulted in the restored system being corrupted. (#46148)

### Possible deadlock in multithreaded backup after file error

If a file read or write error occurred while a multithreaded backup is being written, it attempted to write the error to the log without releasing a mutex, resulting in a deadlock. (#46243)

### .gz not appended to short backup file names

When creating a compressed backup using fullBackupCompressedTo:, a ".gz" should be automatically appended if it is not provided in the filename string. If the filename string was fewer than 4 characters, the ".gz" suffix was not appended. (#46169)

## Indexing bugs

### **GsQuery reject: incorrect results**

The code for GsQuery reject: included a bug and returned the wrong results. (#46184)

### **GsQuery detect:ifNone: returned query instead of ifNone: result**

Rather than returning the results of the `ifNone:` block, on a negative result it returned the query. (#46184)

### **ReversedRangeIndexReadStream size incorrect**

Sending `#size` to an indexing reverse read stream returned 0, rather than the number of elements in the results to be streamed over. (#46184)

### **Index Audit could miss invalid encryption keys**

Index audit did not catch all cases of invalid encryption keys in interior nodes. (#46069)

## Remote shared cache related issues

### **Remote shared page cache monitor can get stuck spin lock**

There was a code path in which a remote shared page cache could have deadlocked on a stuck spin lock. (#46175)

### **After remote cache startup timeout, cache status may be stuck**

If a remote cache fails to complete startup within the time period of `STN_REMOTE_CACHE_STARTUP_TIMEOUT`, the stone starts shutting down and cleaning up after the startup attempt. There were timing cases in which the remote cache could complete the startup and shut itself down, leaving the Stone status incorrect. (#46020)

### **Page Manager crash on removing pages from remote cache**

It was possible for the Page Manager thread in the stone to detect a fatal error and crash, when there was a mismatch in the number of pages that it had requested to remove from remote shared page caches. (#46171).

## GsFile write following a read may write at end of file

When a GsFile was open for update (that is, for both reading and writing), in certain cases if a read operation is followed by a write, the text to be written was appended to the end, not at the desired location. (#46120)

## GsSecureSocket>>readWillNotBlockWithin: wrong after partial read

If a read performed on a GsSecureSocket did not read all available characters, a subsequent call to `GsSecureSocket>>readWillNotBlockWithin:` incorrectly returned false. (#46174)

## Large operations on hidden sets can run out of ME object memory space

Operations on hidden sets, such as `System>>hiddenSetEnumerate:limit:`, against very large bitmaps could encounter a logic error that filled the ME space part of temporary object memory. In-memory garbage collection was not triggered when ME Space was full but other spaces did not need GC. This resulted in an error "VM temporary object memory is full , ME-space overflow" errors (4067), which could not be trapped using the `AlmostOutOfMemory` exception. (#46054, #46058)

## Configuration Parameter Changes

the maximum for `GEM_TEMPOBJ_MESPACE_SIZE` and `GEM_TEMPOBJ_POMGEN_SIZE` has been increased from 1000000 to 64000000

The minimum for `GEM_TEMPOBJ_MESPACE_SIZE` has been changed from 1000 to 2000.

## KeyValueDictionary become: did not correctly reset parent references

When a `KeyValueDictionary become:` occurs, any `CollisionBuckets` within the `KeyValueDictionary` should have their parent reset to the new containing dictionary. This step was not being done. Operations on collision buckets do not normally invoke the parent, so no consequences of this bug are known. (#46194)

## Invalid class instance variables does not prevent class creation

If the `classInstanceVariable:` argument to a class creation method is invalid, an error is returned. Due to the way class instance variables are added after class creation, the class was still created, but was not made invariant. (#46166)

Several private subclass creation methods have been modified to include a `classInstanceVariable:` argument, and the private methods without this argument have been removed.

## GsMethod did not include environmentId method

This method was not provided (#46095)

