GemStone/S 64 Bit™ Release Notes

Limited Distribution
Special Release

Version 3.4.6.1

February 2021

GemTalk™ Systems

INTELLECTUAL PROPERTY OWNERSHIP

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

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

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

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

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

COPYRIGHTS

PATENTS

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

TRADEMARKS

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

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

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

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

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

Linux is a registered trademark of Linus Torvalds and others.

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

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

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

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

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

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

Raspberry Pi is a trademark of the Raspberry Pi Foundation

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

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

Table of Contents

Chapter 1. GemStone/S 64 Bit 3.4.6.1 Release Notes

Chapter

1

GemStone/S 64 Bit 3.4.6.1 Release Notes

Overview

GemStone/S 64 Bit™ 3.4.6.1 is a limited distribution special release version of the GemStone/S 64 Bit object server. This release includes a workaround for a critical bug, as well as other critical bug fixes and other changes.

The previous release, v3.4.6 (also a limited distribution special release), contained an incomplete implementation of STN_TRAN_INCREMENTAL_LOGGING; some code paths remained in which incremental transaction logging could occur. These release notes describe all changes that are in v3.4.6.

Supported Platforms

GemStone/S 64 Bit version 3.4.6.1 is supported on the following platforms:

- Red Hat Enterprise Linux Server/CentOS 6.10, 7.8, and 8.1; and Ubuntu 16.04, 18.04, and 20.04, on x86.

Workaround for Critical Bug

Modifications to kinds of IdentityBag in nested transactions may not tranlog correctly

These issues are not entirely fixed in v3.4.6.1; if you are using nested transactions with affected collections, you should disable incremental tranlogging, using the new configuration parameter STN_TRAN_INCREMENTAL_LOGGING.

If changes are made to an IdentityBag or a subclass of IdentityBag from within a nested transaction, the changes may not be logged correctly in the transaction logs.

If the database is later restored from backup and the transaction logs replayed, or if the transaction logs are applied to a warm or hot standby, the collection will be incorrect in the replayed repository.

One case of this corruption is fixed in v3.4.6.1, affecting additions. The added elements were missing from the collection in the restored repository, and the value for the variable that tracks the size of the collection was incorrect (#49092).

A further problem still exists. With incremental tranlogging, only specific changes to collections are logged to the transaction logs, for efficiency. These incremental tranlog entries are not done correctly for kinds of IdentityBag from within nested transactions. In the restored repository, the collection contained nils instead of valid objects. (#49382)

To workaround this problem, set the configuration parameter STN_TRAN_INCREMENTAL_LOGGING to false. This avoids incremental tranlogging and the bug cannot manifest. Note, however, that the transaction logs will be larger.

Added configuration parameter STN_TRAN_INCREMENTAL_LOGGING

To avoid bug #49382, the configuration parameter STN_TRAN_INCREMENTAL_LOGGING has been added. The default is true, providing behavior from previous release. When false, operations such as changes to collections that are usually incrementally logged are instead logged in full.

This will result in larger tranlog volume.

STN_TRAN_INCREMENTAL_LOGGING
FALSE shuts off the incremental logging of additions or deletions to nodes of large objects, with side effect of more bytes written to tranlogs.

Default TRUE.

Changes in this Release

GemStone revision numbering

GemStone internal development has moved from SVN to Git. Build numbers within the GemStone product (such as log file headers) that previously showed the 5-digit revision, now include the Git SHA; for completeness the entire 40 characters are included.

For example:

```
VERSION: 3.4.6.1, Sun Feb 14 17:14:11 2021
COMMIT: 2021-02-14T16:20:54-08:00 1e6baccce1ad342954012f3c76d1342af5c2627c
```

Updated library versions

- The version of OpenSSL has been updated to 1.1.1h.
- The version of Kerberos has been updated to v1.18.1.
- The version of OpenLDAP has been updated to 2.4.55.

Updated Linux platform support

Support has been added for Red Hat 8.1 and Ubuntu 20.04. Note that previous versions of GemStone 3.4.x cannot run on Ubuntu 20.04 due to OS changes in mmap().

GemStone on Mac "Catalina"

While GemStone is supported (for development) on Mac OSX 10.15.x (Catalina), the most recent security features are triggered when the software is downloaded from a website. To avoid GemStone executables being blocked as security risk, download the GemStone distribution using curl, wget or ftp. (#48841)

Support for UUIDs

Universally Unique Identifiers are 128-bit numbers that, while not relying on a registry, provide practically unique numbers.

UUIDs are provided by the new class GsUuidV4. GsUuidV4 is an implementation of a version 4 UUID as specified in RFC 4122, A **U**niversally **U**nique **ID**entifier (UUID) URN Namespace. UUIDs are generated randomly using the secure OpenSSL random number generator.

Instances of GsUuidV4 are invariant and therefore cannot be modified.

To create an instance of UUID:

`GsUuidV4 class >> new`

`GsUuidV4 class >> fromString:` *aUuidString*
aUuidString must be a valid UUID version 4 string in the following format:
xxxxxxxx-xxxx-4xxx-Vxxx-xxxxxxxxxxxx
where x is any valid lower-case hex digit and V is one of 8, 9, a or b.

See the image for additional protocol.

GsHostProcess added methods

The following methods have been added to simplify use of GsHostProcess:

`GsHostProcess >> execute:` *commandLineString*
Executes *commandLineString* in a child process. If child returns non-zero exit status, signals an Error using contents of stderr from the child. Otherwise returns a String containing stdout from the child. Lookup in the PATH environment variable is not performed, *commandLineString* must specify a complete path to an executable or script. Requires that the executing UserProfile not have NoPerformOnServer privilege.

`GsHostProcess >> execute:` *commandLineString* `input:` *stdinString*
Similar to `GsHostProcess >> execute:`, but writes *stdinString* to the stdin of the child. *stdinString* may be nil, in which case nothing is written to stdin of child.

GsFile added methods

The following method has been added, allowing to read a file and return the contents in a single operation.

`GsFile class >> getContentsOfServerFile:` *filePathAndName*
Return the contents of the named file in a single operation. Use caution when reading a large file as doing so may cause an out of memory error.

GsSecureSockets support for SNI

Server Name Indication (SNI) provides a way for a client to provide a hostname along with the other connection parameters. This allows the server to present multiple certificates in response to connections on the same IP address/port.

`GsSecureSocket >> setServerNameIndication`
Answer a string indicating the SNI name assigned to the receiver, or nil if no SNI name has been assigned.

`GsSecureSocket >> setServerNameIndication:` *aString*
Sets the SNI name for the receiver to be *aString*. This method is only valid for client sockets which have not yet securely connected to a peer.

GsSocket closeAll no longer closes GsSockets created by fromFileHandle:

`GsSocket class >> closeAll` previously closed all GsSockets. Now, GsSockets created by `GsSocket >> fromFileHandle:` are not closed by `closeAll`.

Removed Ruby class and associated methods

The Ruby-support class Regexp has been removed, and is now in ObsoleteClasses. A number of methods that reference this class have been updated.

The following method has been removed:

```
Object >> _isRegexp
```

Utility Script Changes

Script to verify secure backup renamed

The script that is used to verify a secure backup, **verify_backup_with_openssl.sh**, has been renamed to omit the trailing .sh; it is now **verify_backup_with_openssl**.

This script also now supports the **-h** argument to print usage information.

Functionally, the script is unchanged.

topaz DUMPOBJ command added

DUMPOBJ *anObjectSpecification*

This command does a low-level dump of an object, the node of a large object, or the node of an IdentityBag. For a byte format object, it displays the integer values of the bytes. This command cannot be abbreviated.

Configuration Parameter changes

The following configuration parameters have been added:

STN_TRAN_INCREMENTAL_LOGGING

The Stone configuration parameter STN_TRAN_INCREMENTAL_LOGGING has been added. This allows you to disable incremental transaction logging, avoiding one of the required conditions for bug #49382. For details on how and why to use this, see page 6.

GemFailSafeNscEnumerate

Runtime-only configuration parameter for the Gem, which enables fail-safe enumeration logic in primitives for IdentityBag. This parameter should only be enabled after getting a corrupt object error (error 2261) during an enumeration.

To enable, execute:

```
System gemConfigurationAt: #GemFailSafeNscEnumerate put: true
```

Statmonitor Changes

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 creates new files at a specified interval, 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)

Added statmonitor option

The **-a** option has been added; this is similar to **-A** except that it does not collect CPU statistics. This avoids clutter for systems with very large numbers of cores.

-a Collect all available system statistics except per-core CPU statistics.

Foreign Function Interface (FFI) related changes

Pre-parsing of C header files

Previously, the FFI parsing of C library headers could fail with unexpected token errors, depending on definitions within the header file and its include files. (#47979).

Now, before the FFI performs parsing, the native C preprocessor is automatically run on the header file.

Note that header file names/paths that contain characters outside of the ASCII range are not supported.

C Preprocessor must be available

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

Comment style requirements change

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

Comments no longer included in output

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

GciLibrary instance methods no longer include comments

GciLibrary instance methods are generated using the FFI, and as a result of the preprocessing changes, these methods no longer include any comments from the source in `gci.hf`. Refer to the *GemBuilder for C* manual, or the text in `$GEMSTONE/include/gci.hf`, for function details.

CHeader specification of a library search path

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

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

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

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

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

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

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

Subclassing FFI classes

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

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

FFI-related bug fixes

CHeader macros do ## concatenation before argument substitution

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

The enum type was not handled correctly

If a C function signature declared an argument as enum, the wrapper for C calls was generated incorrectly as ptr. (#48867, 48859)

Return type of void* was generated as void

Return type of void* was generated as void. (#48748)

Other added and removed methods

The following method have been added by the code changes.

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

The following methods have been removed by code changes:

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

Bug Fixes

The following bugs that were present in v3.4.5 are fixed in this version.

Abort of a nested transaction may not roll back all changes

Sending #objectSecurityPolicy: to an object changes the GsObjectSecurityPolicy associated with an object. These changes were not rolled back correctly when an abort was done in a nested transaction. (#49104)

Some changes to a kind of IdentityBag within a nested transaction were not rolled back correctly by an abort, and if the outer transaction committed, it was possible for the change that was meant to be aborted to become committed. (#49106)

See page 5 for more on issues related to IdentityBags within nested transactions.

Repository scans do not scan possibleDead objects

Repository scan operations, such as `objectAudit`, `listInstances`, etc., do not scan or report on objects that are in the possibleDead set (as read on the preceding abort). After the possibleDead Objects are voted on by other sessions, these objects may either be garbage collected, or restored as valid objects in the repository.

Following the changes in this release, it is possible for a repository scan operation to return objects that will end up not existing or that are a different actual object (if the OOP is reused for another object); that is, if the results of the scan are retained through the garbage collection cycle, or persisted. However, possibly-valid search results will no longer be missing. (#48580)

It is recommended to do a `markForCollection` and `reclaimAll` before scanning, to get the most reliable results for repository scans.

objectAudit did not run to completion for some cases of corruption

When `Repository >> objectAudit` encountered a corrupted data page that could not be read, it stopped, rather than completing the audit and reporting that as an error. (#48518)

Indexes with optionalPathTerms error on removing elements without the term

GemStone indexing allows you to specify the `optionalPathTerms` option, which allows you to add objects to an indexed collection that do not have an instance variable matching the path term that the index is on. Attempting to remove an object that did not have that instance variable from a collection with optional path term index, however, encountered an error. (#48965)

Statmonitor could fail to add correct file extension

When specifying a file name for a statistics file using the **-f** or **-F** command, if the file pattern included `.lz4` or `.out` in a non-terminal location, the correct extension was not appended, resulting in a misleading file name. (#48531)

logreceiver may need restart after some logsender connection failures

When the logsender shuts down uncleanly and is restarted, the logreceiver reconnects automatically. However, it may not correctly reinitialize its data queue and attempt to write data received just prior to the disconnect that was re-transmitted from the logsender. (#49100).

copydbf with compression to directory did not compress

When the destination for a **copydbf** is a directory rather than a file name, a compression argument (**-Z**, **-z**, **-C**) also included was not applied. (#48540)