GemStone/S 64 Bit™
Release Notes

Version 3.6.3

December 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, **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 3.6.3 release. Read these release notes carefully before you begin installation, upgrade, or development with this release.

No separate Installation Guide is provided with this release. For instructions on installing GemStone/S 64 Bit version 3.6.3, or upgrading or converting from previous products or versions, see the Installation Guide for version 3.6.2.

Terminology Conventions

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

Technical Support

Support Website

gemtalksystems.com

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

- **Documentation** for the current and for previous released versions of all GemTalk products, in PDF form.
- **Product download** for the current and selected recent versions of GemTalk software.

- **Bugnotes**, identifying performance issues or error conditions that you may encounter when using a GemTalk product.
- **Supplemental Documentation** and **TechTips**, providing information and instructions that are not in the regular documentation.
- **Compatibility matrices**, listing supported platforms for GemTalk product versions.

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

Help Requests

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

Website: techsupport.gemtalksystems.com

Email: techsupport@gemtalksystems.com

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

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

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

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

24x7 Emergency Technical Support

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

Training and Consulting

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

Table of Contents

Chapter 1. Release Notes for 3.6.3

Overview 7

Supported Platforms 7

Platforms for Version 3.6.3 7

GemBuilder for Smalltalk (GBS) Versions 8

VSD Version. 8

Changes in this version 9

Distribution Changes 9

Data Encryption using AWS key management 9

Locking 10

Encryption/decryption 10

Key rotation 11

Making a copy of a key. 11

Key equality 11

System clock. 11

Restore from Backup specifying SystemUser password 11

Security Implications of this change 12

New Restore Methods 12

Post-restore login as SystemUser 12

GsTestResult adjustments 13

GsTestResult did not distinguish errors and failures 13

GsTestResult stored printStrings rather than instances 13

Bugs Fixed 14

Strings with µ/181 and ÿ/255 do not convert to uppercase correctly . . . 14

Handling of remote caches that failed startup. 14

Chapter

1

Release Notes for 3.6.3

Overview

GemStone/S 64 Bit™ 3.6.3 is a new version of the GemStone/S 64 Bit object server. Version 3.6.3 includes AWS key management, a new restore feature, and fixes a number of bugs. We recommend everyone using or planning to use GemStone/S 64 Bit upgrade to this new version.

These Release Notes include changes between the previous version of GemStone/S 64 Bit, v3.6.2, and v3.6.3. If you are upgrading from a version prior to 3.6.2, review the release notes for each intermediate release to see the full set of changes.

The Installation Guide has not been updated for this release. For installation, upgrade and conversion instructions, use the Installation Guide for version 3.6.2.

Supported Platforms

Platforms for Version 3.6.3

GemStone/S 64 Bit version 3.6.3 is supported on the following platforms:

- Red Hat Enterprise Linux Server and CentOS Linux 7.9 and 8.4; and
  Ubuntu 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 on x86
- AIX 7.1 and 7.2
- OSX 11.1 (Big Sur) with Darwin 20.2.0 kernel on x86, and OSX 10.15.6 (Catalina) with Darwin 19.6.0 kernel; and
  OSX 11.6 (Big Sur) with Darwin 20.6.0 kernel on Apple M1
  (Mac is supported for development only)

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

GemBuilder for Smalltalk (GBS) Versions

GemStone/S 64 Bit version 3.6.3 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 3.6.3:

GBS/VW version 8.5

VisualWorks **9.0** **32-bit and 64-bit**	**VisualWorks** **8.3.2** **32-bit and 64-bit**	**VisualWorks** **8.2.1** **32-bit and 64-bit**
▸ Windows 10 ▸ RedHat ES 7.9 and 8.4; Ubuntu 18.04 and 20.04	▸ Windows 10 ▸ RedHat ES 7.9 and 8.4; Ubuntu 18.04 and 20.04	▸ Windows 10

GBS/VA version 5.4.6

VAST Platform **10.0.2**	**VA Smalltalk** **9.2.2**	**VA Smalltalk** **8.6.3**
▸ Windows Server 2016 and Windows 10	▸ Windows Server 2016 and Windows 10	▸ Windows Server 2016 and Windows 10

For more details on 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 v3.6.3 distribution includes VSD version 5.5.2; this is the same version that was included in the previous version of GemStone/S 64 Bit, v3.6.2.

Note that in GemStone/S 64 Bit v3.6 and later, **statmonitor** writes additional information to the statmonitor file. As a result, statmonitor files from v3.6 and later cannot be read by versions of VSD earlier than v5.5. VSD 5.5.2 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 5.5.2 is included with the GemStone distribution, and can also be downloaded as a separate product from https://gemtalksystems.com/vsd/

Changes in this version

Distribution Changes

To support the AWS key management feature, $GEMSTONE/lib now includes the shared libraries:

```
libaws-3.6.3-64.so
libgsaws-3.6.3-64.so
```

and the license file, $GEMSTONE/licenses/aws-sdk-cpp.txt.

Data Encryption using AWS key management

GemStone now supports key management using Amazon Web Services (AWS), on Linux and MacOS on Intel.

Amazon Web Services (AWS) provides tools and features that support applications running in the Amazon Cloud. AWS's Key Management Service (KMS) provides a centralized way to securely manage cryptographic keys. New classes in GemStone allow you to use KMS for data encryption.

The AWS SDK for C++ (see https://github.com/aws/aws-sdk-cpp#aws-sdk-cpp) is an open-source project that provides a production-quality C++ interface to AWS. The GemStone now includes shared libraries based on the AWS toolkit. These libraries are included on Linux and Mac OS, which are the platforms that are supported by both the GemStone server and the AWS toolkit.

This version of GemStone includes AWS SDK for C++ v1.9.140.

The following classes have been added:

AbstractCloudCredentials - Abstract class, provided for future use

AwsCredentials - encapsulates the private and public keys needed for authentication to Amazon Web Services (AWS).

AbstractCloudKey - Abstract class, provided for future use

AwsDataKey - holds a local encryption key that can be used to encrypt and decrypt Strings and ByteArrays. This requires a CMK (Amazon Key Management Service's Customer Master Key), and the AwsCredentials that are needed to authentication the CMK.

AwsError - signalled when errors occur in AWS API classes.

To use this feature, you must have an account with Amazon Web Services, have created an AWS Identity and Access Management (IAM) user, and obtained the credentials for that user. These will include an access key (aws_access_key_id) and a secret key (aws_secret_access_key). Access to KMS requires authentication using these AWS keys. Using KMS, you acquire a Customer Master Key (CMK), which is used for the encryption. Consult the AWS documentation, for more information.

GemStone's API allows you to authenticate a CMK using the IAM credentials, and use the resulting key to encrypt a String or ByteArray, which can be decrypted using that key.

A key is created using code such as:

```
credentials := AwsCredentials
   newWithAccessKeyId: 'AKIAIOSFODNN7EXAMPLE'
   secretKeyId: 'wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY'.
cmk := 'arn:aws:kms:us-east-2:1234567890:key/12345-67890'.
key := AwsDataKey
   createKeyUsingAwsCredentials: validCreds
   cmsKeyId: cmk
   keySizeBytes: 16
```

This instance of AwsDataKey should be persisted in the database for ongoing use for encryption and decryption. If the key instance is lost, any data that was encrypted with that key cannot be decrypted, and is permanently inaccessible, even if the credentials and CMK are available.

AwsCredentials can be recreated as needed. Since a persisted instance of AwsDataKey is locked in newly logged in sessions, the same or a recreated instance of AwsCredentials will be needed to unlock a persisted instance of AwsDataKey.

Locking

An AwsDataKey may be in a locked or unlocked state. A newly created key is unlocked. The key can be explicitly locked, and is automatically locked when the creating session logs out. If it is saved to disk, it is saved in the locked state.

If a key is locked, it cannot be used to encrypt or decrypt data. Before it can be used, it must be unlocked using an instance of AwsCredentials that can authenticate the CMK that is contained in the AwsDataKey.

The following methods are available:

```
AwsDataKey >> lock
AwsDataKey >> unlockWithAwsCredentials: anAwsCredentials
AwsDataKey >> isLocked
AwsDataKey >> isUnlocked
```

Encryption/decryption

Encryption and decryption done by sending a request to the (unlocked) AwsDataKey.

Encryption is performed using the AES-OCB authenticated encryption, which ensures data that has been successfully decrypted has not been modified in any way.

`AwsDataKey >> encrypt:` *srcByteObj* `into:` *destByteObj*

Uses the receiver to encrypt *srcByteObj* and stores the resulting encrypted bytes into *destByteObj* as a Base64 string. *srcByteObj* must be a non-empty byte object. *destByteObj* must be a mutable byte object; any contents will be overwritten.

`AwsDataKey >> encryptAndErase:` *srcByteObj* `into:` *destByteObj*

Encrypt *srcByteObj*, erasing all data in *srcByteObj*, and store the resulting encrypted bytes into *destByteObj* as a Base64 string. *srcByteObj* must be a non-empty byte object. *destByteObj* must be a mutable byte object; any contents will be overwritten.

`AwsDataKey >> decrypt:` *srcByteObj* `into:` *destByteObj*

Uses the receiver to decrypt *srcByteObj* and stores the resulting decrypted bytes into *destByteObj*. *srcByteObj* must be a non-empty byte object containing Base64 encrypted text. *destByteObj* must be an mutable byte object.; any contents will be overwritten.

Key rotation

AWS KMS provides the ability to do key rotation for improved security. You may update an instance of AwsDataKey with a new CMS using the following method. To ensure that the change will be committed if successful, this method write locks the receiver and commits the session, so this requires that there are no uncommitted changes.

`AwsDataKey >> changeCmsKeyIdTo:` *newCmsKeyId* `usingNewCredentials:` *newCreds*

Atomically update the receiver to use *newCmsKeyId*, which is accessed with *newCreds* in a single operation, and commit. The receiver must be unlocked, the session must not have uncommitted changes, and the session must be able to write lock the receiver.

Making a copy of a key

An AwsDataKey can be copied. The copy of an AwsDataKey has the same locked state as the original, and can be unlocked using the same credentials.

`AwsDataKey >> copy`

Answer a new object which is a deep copy of the receiver. The lock state of the receiver is preserved when copied, i.e.: if the receiver is unlocked the resulting copy will also be unlocked.

Key equality

AwsDataKey objects are considered equal (=) if they have the same CMK (Customer Managed Key) string and the same encrypted key string. Lock state does not affect equality.

System clock

AWS request signing uses the system clock to include a timestamp in the signature. For this reason, AWS methods may fail if the system clock time is incorrect.

Restore from Backup specifying SystemUser password

GemStone backups include all objects in the repository, which include UserProfiles such as SystemUser and DataCurator, and the passwords for these accounts at the time the backup was made. These passwords, of course, must be known to GemStone Administrators, but should be kept private and changed regularly for system security.

In previous releases, this added a requirement that along with GemStone backups, the administrative passwords that were valid at the time of that particular backup had to be recorded. Otherwise, while the backup could be restored if necessary, it was possible that administrative accounts could not log in.

Now, you may restore secure or unsecured backups and at the same time, specify a new password for SystemUser, using Repository methods with the new `newSystemUserPassword:` keyword. This allows you to restore a backup so the restored repository will have SystemUser’s password set to the argument password.

SystemUser can login, and update other passwords as needed to allow complete use of the restored repository.

Security Implications of this change

If your application contains sensitive data that should not be visible to unauthorized individuals, there are several administrative requirements that become more immediately visible with this feature. These requirements are not new; in previous releases plain text data may be visible within the bytes of a non-secured backups.

- All backups should be secured, using `Repository>>secureFullBackup*` methods; or access to/file permissions of backup files must be strictly controlled. Note that, while not officially supported, a backup from a previous release of GemStone can be restored into v3.6.3.
- Only GemStone userProfiles for users with full authorization to see all data in the repository should be granted FileControl privilege. This privilege allows you to make programmatic backups as well as restore and perform transaction log operations.

New Restore Methods

The following instance methods have been added to Repository, which are variants of existing methods with the additional keyword `newSystemUserPassword:`.

```
restoreFromBackup:newSystemUserPassword:
restoreFromBackups:newSystemUserPassword:
restoreFromSecureBackup:privateDecryptionKey:
   passphrase:newSystemUserPassword:
restoreFromSecureBackup:privateDecryptionKey:passphraseFile:
   newSystemUserPassword:
restoreFromSecureBackups:privateDecryptionKey:passphrase:
   newSystemUserPassword:
restoreFromSecureBackups:scavengePagesWithPercentFree:
   privateDecryptionKey:passphrase:newSystemUserPassword:
```

Previously, the simple secure backup method accepting a passphrase file did not exist; this method, and the variation with the additional keyword, have also been added:

```
restoreFromSecureBackups:privateDecryptionKey:passphraseFile:
restoreFromSecureBackups:privateDecryptionKey:passphraseFile:
   newSystemUserPassword:
```

The argument to the `newSystemUserPassword:` keyword is a String meeting the same restrictions as to the `password:` method. Following the restore, login as SystemUser requires this password string.

Post-restore login as SystemUser

When restoring using the methods that allow you to specify a new SystemUser password, you must be logged in as SystemUser. This restores the objects in the backup; but the new SystemUser password is saved in encrypted form in the root page state of the repository. The restore itself does not update SystemUser’s UserProfile object.

It is strongly recommended to log in again as SystemUser after the `commitRestore` (if you are in full transaction logging mode), or after the restore is complete (in partial logging mode). This login will cause the SystemUser password to be updated in SystemUser's UserProfile.

- If you make a programmatic backup before the SystemUser login (post-commitRestore), the backup will **not** contain the password defined by `newSystemUserPassword:`; since backup backs up the UserProfile data. If this second backup is restored (without using `newSystemUserPassword:`) the SystemUser password will be the one in the original backup, not the one specified by the first restore using `newSystemUserPassword:`.
- It is disallowed to explicitly set SystemUser's password (using e.g. `password:`), when logged in as a user other than SystemUser, after a restore specifying a new SystemUser password but without any later (post-commitRestore) logins as SystemUser.

Shutting down the stone does not affect the SystemUser password state in the root page; after restart, the SystemUser password remains as set by `newSystemUserPassword:`.

GsTestResult adjustments

In v3.6.2, the former example SUnit classes customized for GemStone testing were moved to the base image. There are some API changes in this release.

GsTestResult did not distinguish errors and failures

GsTestResult treated errors and failures as the same, which led to miscounting of errors and of failures. SUnit supports defects, which includes both errors and failures.

GsTestResult now tracks defects, as well as supporting the superclass errors and failures. The instance variable defects has been added to GsTestResult.

Note that the `TestCase >> defects` returns results sorted with all errors followed by all failures, while `GsTestCase >> defects` returns both errors and failures in order of occurrence.

The methods for logging failures has been corrected; the methods:

`GsTestResult >> logFailure:` *aTestCase*

`GsTestResult >> failureLogFile`

have been removed, and are replaced by new methods:

`GsTestResult >> logDefect:` *aTestCase*

`GsTestResult >> defectLogFile`

And these methods now write to a log file named SUnitDefects.log, rather than SUnitFailures.log

GsTestResult stored printStrings rather than instances

GsTestResult stored printStrings of the GsTestCase instance run, rather than the instances themselves. This is more space efficient, but sending methods such as `isError:` to these objects returned incorrect results (always false).

The following methods, inherited from TestResult, are now disallowed for GsTestResult:

```
isError:
isFailure:
isPassed:
```

Bugs Fixed

The following bugs in v3.6.2 are fixed in v3.6.3.

Strings with µ/181 and ÿ/255 do not convert to uppercase correctly

The characters µ/code point 181 and ÿ/code point 255 have uppercase forms that are outside of the range of a Single-byte String. Sending `asUppercase` to a single-byte String containing either of these characters did not convert to a DoubleByteString, and resulted in a single-byte String containing the wrong upper case form. (#49741)

Handling of remote caches that failed startup

When a remote cache fails to start, the messages were misleading, and the retry logic incorrectly attempted retrying the fork. (#48718)