GT.M

Release Notes

V7.0-004

Contact Information

GT.M Group
Fidelity National Information Services, Inc.
347 Riverside Drive
Jacksonville, FL 13220
United States of America

GT.M Support for customers: gtmsupport@fisglobal.com
Automated attendant for 24 hour support: +1 (484) 302-3248
Switchboard: +1 (484) 302-3160

Legal Notice

Revision History		
Revision 1.0	20 September 2022	V7.0-004

Table of Contents

This page is intentionally left blank.

V7.0-004

Overview

V7.0-004 adds a feature to MUPIP CREATE that creates V6.3-014 database files. This makes it possible to seamlessly operate on V6 databases while using the latest V7 software. In addition, there are various enhancements including OpenSSL 3.0 and TLSv1.3 compatibility, additional $ZSOCKET() functionality Socket, restriction settings for LKE and audit logging for MUPIP, as well as various fixes. For more information, refer to the Change History section.

Items marked with ● document new or different capabilities.

Please pay special attention to the items marked with the symbols ● as those document items that have a possible impact on existing code, practice or process. Please be sure to recompile all objects to ensure all the updates are in place.

> **Note**
>
> While FIS keeps message IDs and mnemonics quite stable, messages texts change more frequently as we strive to improve them, especially in response to user feedback. Please ensure you review any automated scripting that parses GT.M messages.

Conventions

This document uses the following conventions:

Flag/Qualifiers	-
Program Names or Functions	upper case. For example, MUPIP BACKUP
Examples	lower case. For example: mupip backup -database ACN,HIST /backup
Reference Number	A reference number is used to track software enhancements and support requests. It is enclosed between parentheses ().
Platform Identifier	Where an item affects only specific platforms, the platforms are listed in square brackets, e.g., [AIX]

> **Note**
>
> The term UNIX refers to the general sense of all platforms on which GT.M uses a POSIX API. As of this date, this includes: AIX and GNU/Linux x86_64.

The following table summarizes the new and revised replication terminology and qualifiers.

Pre V5.5-000 terminology	Pre V5.5-000 qualifier	Current terminology	Current qualifiers
originating instance or primary instance	-rootprimary	originating instance or originating primary instance. Within the context of a replication connection between two instances, an originating instance is referred to as source instance or source side. For example, in an B<-A->C replication configuration, A is the source instance for B and C.	-updok (recommended) -rootprimary (still accepted)
replicating instance (or secondary instance) and propagating instance	N/A for replicating instance or secondary instance. -propagateprimary for propagating instance	replicating instance. Within the context of a replication connection between two instances, a replicating instance that receives updates from a source instance is referred to as receiving instance or receiver side. For example, in an B<-A->C replication configuration, both B and C can be referred to as a receiving instance.	-updnotok
N/A	N/A	supplementary instance. For example, in an A->P->Q replication configuration, P is the supplementary instance. Both A and P are originating instances.	-updok

Effective V6.0-000, GT.M documentation adopted IEC standard Prefixes for binary multiples. This document therefore uses prefixes Ki, Mi and Ti (e.g., 1MiB for 1,048,576 bytes). Over time, we'll update all GT.M documentation to this standard.

● denotes a new feature that requires updating the manuals.

● denotes a new feature or an enhancement that may not be upward compatible and may affect an existing application.

⛔ denotes deprecated messages.

△ denotes revised messages.

⊕ denotes added messages.

Platforms

Over time, computing platforms evolve. Vendors obsolete hardware architectures. New versions of operating systems replace old ones. We at FIS continually evaluate platforms and versions of platforms that should be Supported for GT.M. In the table below, we document not only the ones that are currently Supported for this release, but also alert you to our future plans given the evolution of computing platforms. If you are an FIS customer, and these plans would cause you hardship, please contact your FIS account executive promptly to discuss your needs.

Each GT.M release is extensively tested by FIS on a set of specific versions of operating systems on specific hardware architectures, we refer to the combination of operating system and hardware architecture as a platform. We deem this set of specific versions: Supported. There may be other versions of the same operating systems on which a GT.M release may not have been tested, but on which the FIS GT.M Group knows of no reason why GT.M would not work. We deem this larger set of versions: Supportable. There is an even larger set of platforms on which GT.M may well run satisfactorily, but where the FIS GT.M team lacks the knowledge to determine whether GT.M is Supportable and therefore deem them: Unsupported. Contact FIS GT.M Support with inquiries about your preferred platform.

As of the publication date, FIS supports this release on the hardware and operating system versions below. Contact FIS for a current list of Supported platforms. The reference implementation of the encryption reference plugin has its own additional requirements.

Platform	Supported Versions	Notes
IBM Power Systems AIX	7.1 TL 5, 7.2 TL 5	Only 64-bit versions of AIX with POWER7 as the minimum required CPU architecture level are Supported. While GT.M supports both UTF-8 mode and M mode on this platform, there are problems with the AIX ICU utilities that prevent FIS from testing 4-byte UTF-8 characters as comprehensively on this platform as we do on others. Running GT.M on AIX 7.1 requires APAR IZ87564, a fix for the POW() function, to be applied. To verify that this fix has been installed, execute **instfix -ik IZ87564.** Only the AIX jfs2 filesystem is Supported. Other filesystems, such as jfs1 are Supportable, but not Supported. FIS strongly recommends use of the jfs2 filesystem on AIX; use jfs1 only for existing databases not yet migrated to a jfs2 filesystem.
x86_64 GNU/Linux	Red Hat Enterprise Linux 7.9, 8.6; Ubuntu 18.04 LTS,	To run 64-bit GT.M processes requires both a 64-bit kernel as well as 64-bit hardware. GT.M should also run on recent releases of other major Linux distributions with a contemporary Linux kernel (2.6.32 or

Platform	Supported Versions	Notes
	20.04 LTS, and 22.04 LTS	later), glibc (version 2.12 or later) and ncurses (version 5.7 or later).
		Due to build optimization and library incompatibilities, GT.M versions older than V6.2-000 are incompatible with glibc 2.24 and up. This incompatibility has not been reported by a customer, but was observed on internal test systems that use the latest Linux software distributions from Fedora (26), Debian (unstable), and Ubuntu (17.10). In internal testing, processes either hung or encountered a segmentation violation (SIG-11) during operation. Customers upgrading to Linux distributions that utilize glibc 2.24+ must upgrade their GT.M version at the same time as or before the OS upgrade.
		GT.M requires a compatible version of the libtinfo library. On Red Hat, the ncurses-libs and ncurses-compat-libs packages contain the libtinfo library. On Debian/Ubuntu, libtinfo5 and libncurses5 packages contain the libtinfo library. If any of these packages is not already installed on your system, please install using an appropriate package manager.
		To support the optional WRITE /TLS fifth argument (the ability to provide / override options in the tlsid section of the encryption configuration file), the reference implementation of the encryption plugin requires libconfig 1.4.x.
		Only the ext4 and xfs filesystems are Supported. Other filesystems are Supportable, but not Supported. Furthermore, if you use the NODEFER_ALLOCATE feature, FIS strongly recommends that you use xfs. If you must use NODEFER_ALLOCATE with ext4, you **must** ensure that your kernel includes commit d2dc317d564a46dfc683978a2e5a4f91434e9711 (search for d2dc317d564a46dfc683978a2e5a4f91434e9711 at https://www.kernel.org/pub/linux/kernel/v4.x/ChangeLog-4.0.3). The Red Hat Bugzilla identifier for the bug is 1213487. With NODEFER_ALLOCATE, do not use any filesystem other than ext4 and a kernel with the fix, or xfs.
		Ubuntu 21.10 is Supportable.
		Note • To use TLSv1.3 with OpenSSL 1.1.1 and up, you must recompile the reference encryption plugins

Platform	Supported Versions	Notes
		• RHEL 8 includes compat-openssl10.x86_64 for binaries compiled against OpenSSL 1.0.2 on RHEL 7

⚠️ **Important**

> Effective V7.0-003, GT.M is no longer Supportable on the 32 bit x86 platform. Please contact your FIS account manager if you need ongoing support for GT.M on this platform.

Platform support lifecycle

FIS usually supports new operating system versions six months or so after stable releases are available and we usually support each version for a two year window. GT.M releases are also normally supported for two years after release. While FIS attempts to provide support for customers in good standing on any GT.M release and operating system version, our ability to provide support diminishes after the two year window.

GT.M cannot be patched, and bugs are only fixed in new releases of software.

Additional Installation Instructions

To install GT.M, see the "Installing GT.M" section in the GT.M Administration and Operations Guide. For minimal down time, upgrade a current replicating instance and restart replication. Once that replicating instance is current, switch it to become the originating instance. Upgrade the prior originating instance to become a replicating instance, and perform a switchover when you want it to resume an originating primary role.

⚠️ **Caution**

> Never replace the binary image on disk of any executable file while it is in use by an active process. It may lead to unpredictable results. Depending on the operating system, these results include but are not limited to denial of service (that is, system lockup) and damage to files that these processes have open (that is, database structural damage).

• FIS strongly recommends installing each version of GT.M in a separate (new) directory, rather than overwriting a previously installed version. If you have a legitimate need to overwrite an existing GT.M installation with a new version, you must first shut down all processes using the old version. FIS suggests installing GT.M V7.0-004 in a Filesystem Hierarchy Standard compliant location such

as /usr/lib/fis-gtm/V7.0-004_arch (for example, /usr/lib/fis-gtm/V7.0-004_x86_64 on Linux systems). A location such as /opt/fis-gtm/V7.0-004_arch would also be appropriate.

- Use the appropriate MUPIP command (e.g. ROLLBACK, RECOVER, RUNDOWN) of the old GT.M version to ensure all database files are cleanly closed.

- Make sure gtmsecshr is not running. If gtmsecshr is running, first stop all GT.M processes including the DSE, LKE and MUPIP utilities and then perform a **MUPIP STOP *pid_of_gtmsecshr***.

- Starting with V6.2-000, GT.M no longer supports the use of the deprecated $gtm_dbkeys and the master key file it points to for database encryption. To convert master files to the libconfig format, please click ▼ to download the CONVDBKEYS.m program and follow instructions in the comments near the top of the program file. You can also download CONVDBKEYS.m from http://tinco.pair.com/ bhaskar/gtm/doc/articles/downloadables/CONVDBKEYS.m. If you are using $gtm_dbkeys for database encryption, please convert master key files to libconfig format immediately after upgrading to V6.2-000 or later. Also, modify your environment scripts to include the use of gtmcrypt_config environment variable.

Recompile

- Recompile all M and C source files.

Rebuild Shared Libraries or Images

- Rebuild all Shared Libraries after recompiling all M and C source files.

- If your application is not using object code shared using GT.M's auto-relink functionality, please consider using it.

Compiling the Reference Implementation Plugin

If you plan to use the example / reference implementation plugin in support of database encryption, TLS replication, or TLS sockets, you must compile the reference plugin in order to match the shared library dependencies specific to your platform. The instructions for compiling the Reference Implementation plugin are as follows:

1. Install the development headers and libraries for libgcrypt, libgpgme, libconfig, and libssl. On Linux, the package names of development libraries usually have a suffix such as -dev or -devel and are available through the package manager. For example, on Ubuntu_x86_64 a command like the following installs the required development libraries:

   ```
   sudo apt-get install libgcrypt11-dev libgpgme11-dev libconfig-dev libssl-dev
   ```

   Note that the package names may vary by distribution / version.

2. Unpack $gtm_dist/plugin/gtmcrypt/source.tar to a temporary directory.

```
mkdir /tmp/plugin-build
cd /tmp/plugin-build
cp $gtm_dist/plugin/gtmcrypt/source.tar .
tar -xvf source.tar
```

3. Follow the instructions in the README.

- Open Makefile with your editor; review and edit the common header (IFLAGS) and library paths (LIBFLAGS) in the Makefile to reflect those on your system.

- Define the gtm_dist environment variable to point to the absolute path for the directory where you have GT.M installed

- Copy and paste the commands from the README to compile and install the encryption plugin with the permissions defined at install time

> The encryption plugin currently uses functionality that is deprecated in OpenSSL 3.0. This will be fixed in a future release.

Re-evaluate TLS configuration options

The GT.M TLS reference encryption plugin implements a subset of options as documented in the OpenSSL 1.0.2 man page for SSL_set_options which modify the default behavior of OpenSSL. Future versions of the plugin will enable new options as and when the OpenSSL library adds them. To enable options not supported by the GT.M TLS reference plugin, it is possible to create an OpenSSL configuration for GT.M processes. See the OpenSSL man page for "config".

Upgrading to GT.M V7.0-004

There are two upgrade paths available when you upgrade to GT.M V7.0-003 from a V6 version:

- **V7 Upgrade Path 1**: Perform a MUPIP EXTRACT -FREEZE on an existing V6 database and then perform MUPIP LOAD on an empty V7 database using replication to minimize downtime. You can also use the MERGE command to move data from V6 to the new V7 database; as with MUPIP EXTRACT, you need to keep the source data in a stable state. If you are using triggers, extract the triggers from the V6 database and load them in the new V7 database. We recommend you use replication to further stage the conversion and minimize down time.

- **V7 Upgrade Path 2**: Continue using your V6 databases with GT.M V7.0-003. You may wish to keep your latest V6 version available to create new database file, in case you do not wish to operate with files of differing format, as V7 releases always create V7 format databases.

Choose V7 Upgrade Path 1 if you anticipate a database file to grow up to 16Gi blocks or require trees of up to 11 levels. Note that the maximum size of a V7 database file having 8KiB block size is 114688GiB (8KiB*16Gi).

Choose V7 Upgrade Path 2 if you do not anticipate a database file to grow beyond the V6 database limit of 994Mi blocks or a tree depth limit of 7 levels. Note that the maximum size of a V6 database file having 8KiB block size is 7936GiB (8KiB*992Mi).

Other that the new maximum database file size that comes with V7 Upgrade Path 1, there is no difference between V7 Upgrade Path 1 and V7 Upgrade Path 2. You can choose V7 Upgrade Path 2 first and then later choose V7 Upgrade Path 1 if a need arises.

Note

> To upgrade your database from V5 to V7, you need to first upgrade your database from V5 to a V6 database and then choose an appropriate V7 upgrade path. Refer to the appropriate V6 release notes for the database upgrade instructions or consult your GT.M support channel.

Important

> In an upcoming release, the FIS GT.M team plans to provide in-place database upgrade with minimal downtime.

Managing M mode and UTF-8 mode

With International Components for Unicode® (ICU) version 3.6 or later installed, GT.M's UTF-8 mode provides support for Unicode® (ISO/IEC-10646) character strings. On a system that does not have ICU 3.6 or later installed, GT.M only supports M mode.

On a system that has ICU installed, GT.M optionally installs support for both M mode and UTF-8 mode, including a utf8 subdirectory of the directory where GT.M is installed. From the same source file, depending upon the value of the environment variable gtm_chset, the GT.M compiler generates an object file either for M mode or UTF-8 mode. GT.M generates a new object file when it finds both a source and an object file, and the object predates the source file and was generated with the same setting of $gtm_chset/$ZCHset. A GT.M process generates an error if it encounters an object file generated with a different setting of $gtm_chset/$ZCHset than that processes' current value.

Always generate an M object module with a value of $gtm_chset/$ZCHset matching the value processes executing that module will have. As the GT.M installation itself contains utility programs written in M, their object files also conform to this rule. In order to use utility programs in both M mode and UTF-8 mode, the GT.M installation ensures that both M and UTF-8 versions of object modules exist, the latter in the utf8 subdirectory. This technique of segregating the object modules by their compilation mode prevents both frequent recompiles and errors in installations where both modes are in use. If your installation uses both modes, consider a similar pattern for structuring application object code repositories.

GT.M is installed in a parent directory and a utf8 subdirectory as follows:

- Actual files for GT.M executable programs (mumps, mupip, dse, lke, and so on) are in the parent directory, that is, the location specified for installation.

● Object files for programs written in M (GDE, utilities) have two versions - one compiled with support for UTF-8 mode in the utf8 subdirectory, and one compiled without support for UTF-8 mode in the parent directory. Installing GT.M generates both versions of object files, as long as ICU 3.6 or greater is installed and visible to GT.M when GT.M is installed, and you choose the option to install UTF-8 mode support. During installation, GT.M provides an option that allows placing the object code in shared libraries in addition to individual files in the directory.

● The utf8 subdirectory has files called mumps, mupip, dse, lke, and so on, which are relative symbolic links to the executables in the parent directory (for example, mumps is the symbolic link ../mumps).

● When a shell process sources the file gtmprofile, the behavior is as follows:

  - If $gtm_chset is "m", "M" or undefined, there is no change from the previous GT.M versions to the value of the environment variable $gtmroutines.

  - If $gtm_chset is "UTF-8" (the check is case-insensitive),

    - $gtm_dist is set to the utf8 subdirectory (that is, if GT.M is installed in /usr/lib/fis-gtm/gtm_V7.0-004_i686, then gtmprofile sets $gtm_dist to /usr/lib/fis-gtm/gtm_V7.0-004_i686/utf8).

    - On platforms where the object files have not been placed in a libgtmutil.so shared library, the last element of $gtmroutines is $gtm_dist($gtm_dist/..) so that the source files in the parent directory for utility programs are matched with object files in the utf8 subdirectory. On platforms where the object files are in libgtmutil.so, that shared library is the one with the object files compiled in the mode for the process.

For more information on gtmprofile, refer to the Basic Operations chapter of GT.M Administration and Operations Guide.

Although GT.M uses ICU for UTF-8 operation, ICU is not FIS software and FIS does not support ICU.

Setting the environment variable TERM

The environment variable TERM must specify a terminfo entry that accurately matches the terminal (or terminal emulator) settings. Refer to the terminfo man pages for more information on the terminal settings of the platform where GT.M needs to run.

● Some terminfo entries may seem to work properly but fail to recognize function key sequences or fail to position the cursor properly in response to escape sequences from GT.M. GT.M itself does not have any knowledge of specific terminal control characteristics. Therefore, it is important to specify the right terminfo entry to let GT.M communicate correctly with the terminal. You may need to add new terminfo entries depending on your specific platform and implementation. The terminal (emulator) vendor may also be able to help.

● GT.M uses the following terminfo capabilities. The full variable name is followed by the capname in parenthesis:

```
auto_right_margin(am), clr_eos(ed), clr_eol(el), columns(cols), cursor_address(cup),
 cursor_down(cud1), cursor_left(cub1), cursor_right(cuf1), cursor_up(cuu1),
```

```
eat_newline_glitch(xenl), key_backspace(kbs), key_dc(kdch1),key_down(kcud1),
key_left(kcub1), key_right(kcuf1), key_up(kcuu1), key_insert(kich1),
keypad_local(rmkx),keypad_xmit(smkx), lines(lines).
```

GT.M sends keypad_xmit before terminal reads for direct mode and READs (other than READ *) if EDITING is enabled. GT.M sends keypad_local after these terminal reads.

Installing Compression Libraries

If you plan to use the optional compression facility for replication, you must provide the compression library. The GT.M interface for compression libraries accepts the zlib compression libraries without any need for adaptation. These libraries are included in many UNIX distributions and are downloadable from the zlib home page. If you prefer to use other compression libraries, you need to configure or adapt them to provide the same API as that provided by zlib.

If a package for zlib is available with your operating system, FIS suggests that you use it rather than building your own.

By default, GT.M searches for the libz.so shared library in the standard system library directories (for example, /usr/lib, /usr/local/lib, /usr/local/lib64). If the shared library is installed in a non-standard location, before starting replication, you must ensure that the environment variable LIBPATH (AIX) or LD_LIBRARY_PATH (GNU/Linux) includes the directory containing the library. The Source and Receiver Server link the shared library at runtime. If this fails for any reason (such as file not found, or insufficient authorization), the replication logic logs a DLLNOOPEN error and continues with no compression.

Although GT.M uses a library such as zlib for compression, such libraries are not FIS software and FIS does not support any compression libraries.

Change History

V7.0-004

Fixes and enhancements specific to V7.0-004:

Id	Prior Id	Category	Summary
GTM-DE201295	GTM-9265	Language	⊘ Honor timeout value on initial socket connect
GTM-DE270421		DB	Correct navigation when subscript-level mapping "hides" data
GTM-DE297205		Language	Prevent assert from stopping a process that is in the midst of OPEN'ng a SOCKET device
GTM-DE304273	GTM-9480	Language	Appropriate result for SET $ZTRAP under certain conditions
GTM-DE305529		Language	Protect correct value of $ZTSLATE
GTM-DE307442		Language	Handle error with USE LISTEN or CONNECT and IOERROR="T"
GTM-DE308470		Language	Preserve appropriate left-to-right evaluation within $SELECT() when using FULL_BOOLEAN compilation
GTM-F135380	GTM-9374	Admin	Restrictions for LKE and LKE CLEAR ✔
GTM-F135381	GTM-9375	Admin	Audit MUPIP facility and improved handling of network errors by the audit facility ✔
GTM-F158404		Other	Multiple Enhancements for OpenSSL 3.0 and TLSv1.3 compatibility ✔
GTM-F166755		Admin	MUPIP BACKUP retry normalization ✔
GTM-F167559		Admin	GT.M V7 versions support the ability to create database files compatible with GT.M V6.3-014 ✔

This page is intentionally left blank.

_____
GTM V7.0-004

Page 12, September 20, 2022   FIS

Database

- GT.M handles $ORDER(), $ZPREVIOUS(), $QUERY(), $NEXT(), and MERGE operations which traverse subscript-level mappings correctly. Previously, these operations could return gvns which should have been excluded by the global directory. In the case of MERGE, the target could contain data from excluded subtrees. (GTM-DE270421)

This page is intentionally left blank.

Language

- SOCKET device OPEN better honors timeouts; similarly, USE for a SOCKET honors an implicit zero timeout. Note that very short timeouts, particularly zero (0), may be unsuitable for dealing with the timing of some network events, and thus may need adjustment to avoid timing out after this correction. Previously, these commands did not start their own timers to bound the operation, causing the CONNECT to only check for a timeout condition when some other process, or operating system, event interrupted the process. (GTM-DE201295)

- GT.M processes shut down gracefully when they receive a MUPIP STOP while preforming an OPEN of a SOCKET device. Previously, these conditions could cause a GT.M process to issue a GTMASSERT2 error. (GTM-DE297205)

- SET $ZTRAP, for example $ZTRAP=$ETRAP, works correctly during XECUTE frame or from a non-base Direct Mode prompt, for example due to a BREAK command, when gtm_ztrap_new is TRUE. Previously, such a circumstance tended to produce a corrupted value for $ZTRAP. (GTM-DE304273)

- GT.M maintains the correct value of $ZTSLATE. Previously, garbage collection could mishandle the ISV resulting in values not consistent with proper application execution (GTM-DE305529)

- SOCKET device USE handles errors associated with device parameters CONNECT or LISTEN and IOERROR="T" appropriately. Previously, such errors could cause odd behavior including segmentation violation (SIG-11). This was only seen in development and not reported by a customer. (GTM-DE307442)

- Expratoms in a boolean expression are evaluated in left-to-right order in FULL_BOOLEAN compilation mode. Previously, side effects in a $SELECT() could affect earlier expratoms in a boolean expression. (GTM-DE308470)

This page is intentionally left blank.

System Administration

- The LKECLEAR keyword in the GT.M restrictions file prevents the use of the LKE CLEAR command by unauthorized users, while the LKE keyword prevents any unauthorized use of the LKE utility. Previously the only way to restrict the use of LKE was by setting the authorizations on the executable; there was no way to allow the use of LKE while blocking the use of LKE CLEAR. (GTM-F135380) ⊘

- When the Audit MUPIP facility (AM_ENABLE) is enabled, MUPIP establishes a connection (via a UNIX/TCP/TLS socket) to a logger/listener process, and sends any MUPIP shell command through the socket to the listener for logging. If sending is successful, MUPIP executes the command. If the connection is not successful or sending of the command fails, then MUPIP produces an error and does not execute the command. When this facility is enabled, all commands typed at the MUPIP prompt (MUPIP>) produce the RESTRICTEDOP error. When this facility is disabled, which it is by default, MUPIP commands execute as usual.In addition, the APD_ENABLE facility displays the appropriate network error messages and exits the process gracefully. Previously, certain network errors could result in a segmentation fault without reporting the reason. (GTM-F135381) ⊘

- By default, the MUPIP BACKUP -DATABASE retries file copy operation once if the first attempt fails for a retriable reason; this is equivalent to -RETRY=1. -RETRY=0 prevents any retries, so if the copy fails MUPIP immediately reports an error. Additionally, any copy failure error messages include region information when displaying OS service error codes. Previously, while the default was equivalent to -RETRY=1 that incorrectly meant one BACKUP attempt, -RETRY=0 did not perform a backup, and error reports did not identify the region associated with the failure. (GTM-F166755) ⊘

- MUPIP CREATE creates V6 (V6.3-014) database files when the environment variable gtm_db_create_ver is defined as [V]6, or when the command line specifies: -V6. The -NOV6 command line option allowing overrides a gtm_db_create_ver and creates a V7 database file. This means this V7 release can operate seamlessly with V6 databases. Prior V7 versions did not have this support. (GTM-F167559) ⊘

This page is intentionally left blank.

Other

- When using OpenSSL 3.0+, the GT.M TLS example plugin uses automatic Diffie-Hellman parameters. This action is the recommended behavior from the OpenSSL project. Previously, the plugin allowed Diffie-Hellman parameter configuration files which required the user understand how those parameter configuration options affected the Diffie-Hellman key exchange.

  The GT.M example TLS encryption library applies the configuration directed cipher suites, cipher-list, to TLSv1.3 sessions. Previously, the CipherSuites configuration only applied to pre-TLSv1.3 sessions.

  Replication encrypted using the reference TLS example plugin may perform TLSv1.2 re negotiations when the Receiver Server's plugin was compiled with OpenSSL 3.0+. Previously, these would fail with the error message "no renegotiation". (GTM-F158404)

This page is intentionally left blank.

Error and Other Messages

AUDCONNFAIL ⊕

AUDCONNFAIL, Audit XXXX facility failed to connect to audit logger

Run Time Error: The facility for logging activity generated by the use of XXXX is enabled, but is unable to form a connection with its configured logging program. This prevents a process from taking actions configured for logging when using XXXX.

Action: Check to make sure logger program is running and listening/accepting connections. If using a TCP or TLS-enabled logger, make sure the port number the logger is listening/accepting on matches the port number provided in the restriction file. Ensure the provided information (logger's connection info) in the restriction file is correct. Also make sure the line in restriction file is in correct format. If running a TLS-enabled logger, make sure the logger's TLS certificate is signed by a root Certificate Authority specified in the GT.M TLS configuration file. Check syslog or additional messages for more information. After addressing identified issues, restart the XXXX process.

AUDINITFAIL ⊕

AUDINITFAIL, Audit XXXX facility failed to initialize audit information

Run Time Error: GT.M was unable to process or initialize the provided information (e.g. IP, hostname, port number, UNIX domain socket file path, or TLS ID) from the restriction file. This prevents a process from taking actions configured for logging when using XXXX.

Action: Check the restriction file to make sure information is in proper format. After addressing identified issues, restart processes using XXXX.

AUDLOGFAIL ⊕

AUDLOGFAIL, Audit XXXX facility failed to log activity

Run Time Error: GT.M was unable to send the to-be-logged activitiy by XXXX to logger. This prevents a process from taking the action when using XXXX.

Action: Check to make sure that GT.M is able to successfully connect to the logger program. Check syslog or additional messages for more information.

This page is intentionally left blank.