



GT.M

Release Notes

V7.0-000

Empowering
the Financial World

FIS

Contact Information

GT.M Group
Fidelity National Information Services, Inc.
200 Campus Drive
Collegeville, PA 19426
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
Website: <http://fis-gtm.com>

Legal Notice

Copyright ©2021, 2024 Fidelity National Information Services, Inc. and/or its subsidiaries. All Rights Reserved.

















Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts and no Back-Cover Texts.

GT.M™ is a trademark of Fidelity National Information Services, Inc. Other trademarks are the property of their respective owners.

This document contains a description of GT.M and the operating instructions pertaining to the various functions that comprise the system. This document does not contain any commitment of FIS. FIS believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. FIS is not responsible for any errors or defects.

Revision History		
Revision 1.1	05 July 2024	Added GTM-9356.
Revision 1.0	12 February 2021	V7.0-000

Table of Contents

V7.0-000	1
Overview	1
Conventions	1
Platforms	3
Platform support lifecycle	5
32- vs. 64-bit platforms	5
Call-ins and External Calls	5
Internationalization (Collation)	6
Environment Translation	6
Additional Installation Instructions	6
Recompile	7
Rebuild Shared Libraries or Images	7
Compiling the Reference Implementation Plugin	8
Upgrading to GT.M V7.0-000	9
Managing M mode and UTF-8 mode	10
Setting the environment variable TERM	11
Installing Compression Libraries	11
Change History	13
V7.0-000	13
Database	15
Language	17
System Administration	19
Other	21
Additional Information	23
Additional Information about GTM-9302	23
Error and Other Messages	27
BOVTMGTEOVTM 	27
DBADDRANGE 	27
DBADDRANGE8 	27
GTMCURUNSUPP 	27
HEX64ERR 	28
HEXERR 	28
INVSHUTDOWN 	28
NUM64ERR 	28
NUMERR 	28
REPL0BACKLOG 	29
REPLBACKLOG 	29
REPLNORESP 	29
SHUT2QUICK 	30
UNUM64ERR 	30
VIEWREGLIST 	30
WCSFLUFail 	30

This page is intentionally left blank.

V7.0-000

Overview

V7.0-000 supports database files with up to 16Gi blocks, For example, the maximum size of a V7 database file having 8KiB block size is 128 TiB (8KiB * 16Gi) which means, using multiple regions, it is easier to manage petabyte storage capacity. V7.0-000 includes support for operation with V6 database file format used by recent GT.M versions.

V7.0-000 also includes other fixes and enhancements including changes aimed a making management of replication more straightforward. 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

Messages are not part of the GT.M API whose stability we strive to maintain. Make sure that 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 on x86 (32- and 64-bits).

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.	-updnok
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 (the combination of operating system and hardware architecture is referred to as a platform). This set of specific versions is considered 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. This larger set of versions is considered 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. These are considered 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 plugin has its own additional requirements, should you opt to use it as included with GT.M.

Platform	Supported Versions	Notes
IBM Power Systems AIX	7.1 TL 5, 7.2 TL 3	<p>Only 64-bit versions of AIX with POWER7 as the minimum required CPU architecture level are Supported.</p> <p>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.</p> <p>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.</p> <p>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.</p>
x86_64 GNU/Linux	Red Hat Enterprise Linux 7.9, 8.3; Ubuntu	To run 64-bit GT.M processes requires both a 64-bit kernel as well as 64-bit hardware.

Platform	Supported Versions	Notes
	18.04 LTS, and 20.04 LTS	<p>GT.M should also run on recent releases of other major Linux distributions with a contemporary Linux kernel (2.6.32 or later), glibc (version 2.12 or later) and ncurses (version 5.7 or later).</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>Ubuntu 19.10 is Supportable.</p>



Note

Debian 10 (buster) x86 GNU/Linux is Supportable but not Supported. The 32-bit version of GT.M runs on either 32- or 64-bit x86 platforms; we expect the x86_64 GNU/Linux version of GT.M to be preferable on 64-bit hardware. Running a 32-bit

GT.M on a 64-bit GNU/Linux requires 32-bit libraries to be installed. The CPU must have an instruction set equivalent to 586 (Pentium) or better. If you are using the 32-bit GT.M, please also refer to the notes above on x86_64 GNU/Linux.

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 will attempt to provide support to customers in good standing for 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.

32- vs. 64-bit platforms

The same application code runs on both 32-bit and 64-bit platforms; however there are operational differences between them (for example, auto-relink and the ability to use GT.M object code from shared libraries exist only on 64-bit platforms). Please note that:

- You must compile the application code separately for each platform. Even though the M source code is the same, the generated object modules are different - the object code differs between x86 and x86_64.
- Parameter-types that interface GT.M with non-M code using C calling conventions must match the data-types on their target platforms. Mostly, these parameters are for call-ins, external calls, internationalization (collation) and environment translation, and are listed in the tables below. Note that most addresses on 64-bit platforms are 8 bytes long and require 8 byte alignment in structures whereas all addresses on 32-bit platforms are 4 bytes long and require 4-byte alignment in structures.

Call-ins and External Calls

Parameter type	32-Bit	64-bit	Remarks
gtm_long_t	4-byte (32-bit)	8-byte (64-bit)	gtm_long_t is much the same as the C language long type.
gtm_ulong_t	4-byte	8-byte	gtm_ulong_t is much the same as the C language unsigned long type.
gtm_int_t	4-byte	4-byte	gtm_int_t has 32-bit length on all platforms.
gtm_uint_t	4-byte	4-byte	gtm_uint_t has 32-bit length on all platforms



Caution

If your interface uses `gtm_long_t` or `gtm_ulong_t` types but your interface code uses `int` or signed `int` types, failure to revise the types so they match on a 64-bit platform will cause the code to fail in unpleasant, potentially dangerous, and hard to diagnose ways.

Internationalization (Collation)

Parameter type	32-Bit	64-bit	Remarks
<code>gtm_descriptor</code> in <code>gtm_descript.h</code>	4-byte	8-byte	Although it is only the address within these types that changes, the structures may grow by up to 8 bytes as a result of compiler padding to meet platform alignment requirements.



Important

Assuming other aspects of code are 64-bit capable, collation routines should require only recompilation.

Environment Translation

Parameter type	32-Bit	64-bit	Remarks
<code>gtm_string_t</code> type in <code>gtmxc_types.h</code>	4-byte	8-byte	Although it is only the address within these types that changes, the structures may grow by up to 8 bytes as a result of compiler padding to meet platform alignment requirements.



Important

Assuming other aspects of code are 64-bit capable, environment translation routines should require only recompilation.


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 V7.0-000 in a Filesystem Hierarchy Standard compliant location such as /usr/lib/fis-gtm/V7.0-000_arch (for example, /usr/lib/fis-gtm/V7.0-000_x86 on 32-bit Linux systems). A location such as /opt/fis-gtm/V7.0-000_arch would also be appropriate. Note that the **arch** suffix is especially important if you plan to install 32- and 64-bit versions of the same release of GT.M on the same system.
- 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 database encryption, TLS replication, or TLS sockets, you must compile the reference implementation plugin to match the shared library dependencies unique 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



Caution

These are separate steps to compile the encryption plugin for GT.M versions V5.3-004 through V6.3-000 when OpenSSL 1.1 is installed and OpenSSL 1.0.x libraries are still available.

- Download the most recent OpenSSL 1.0.x version
- Compile and install (default installs to /usr/local/ssl)

```
./config && make install
```
- Adjust the configuration : Move the newly installed libraries out of the way

```
mv /usr/local/ssl/lib /usr/local/ssl/lib.donotuse
```
- Adjust the configuration : Create another /usr/local/ssl/lib and symlink the existing 1.0.x library into it as the default. This ensures that the encryption plugin

is compiled using the compatible OpenSSL 1.0.x library. Adjust the path below as necessary.

```
mkdir /usr/local/ssl/lib && ln -s /path/to/existing/libssl.so.1.0.x /usr/local/ssl/libssl.so
```

- Recompile the encryption plugin following the above directions.
- Remove /usr/local/ssl/lib.donotuse to avoid future complications.

Upgrading to GT.M V7.0-000

There are two upgrade paths available when you upgrade to GT.M V7.0-000:

- **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.
- **V7 Upgrade Path 2:** Continue using your V6 databases with GT.M V7.0-000.

Choose V7 Upgrade Path 1 if you anticipate a database file to grow up to 16Gi blocks. 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. Note that the maximum size of a V6 database file having 8KiB block size is 7936GiB (8KiB*992Mi).

Other than 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. Note that on 64-bit versions of GT.M, the object code is in shared libraries, rather than 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_V7.0-000_i686`, then `gtmprofile` sets `$gtm_dist` to `/usr/lib/fis-gtm/gtm_V7.0-000_i686/utf8`).
 - On platforms where the object files have not been placed in a `libgtmutil.so` shared library, the last element of `$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(smkn), 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

V7.0-000

Fixes and enhancements specific to V7.0-000:

Id	Prior Id	Category	Summary
GTM-6952	-	Admin	Utilites accept deicmal or hexadecimal input in many places 🟢
GTM-8262	-	Admin	gtmprofile prints proper error messages if UTF-8 dependencies are not met
GTM-8263	-	Admin	UTF8 installation simplifications 🟢
GTM-8288	-	Admin	Install confiuration leave clean file authorizations 🟢
GTM-8517	-	Admin	GT.M installation script records ownership, permissions and openssh sha256 checksum values of files 🟢
GTM-8913	-	DB	🔴 Revised database format supports up to 16Gi blocks in a region 🟢
GTM-9268	-	Admin	JNLFLUSHLOG in the syslog if journal buffer flushing fails
GTM-9302	-	Admin	Acknowledged sequence number in -SOURCE -SHOWBACKLOG and available with ^%PEEKBYNAME 🟢
GTM-9340	-	Language	VIEW and \$VIEW() truncate region names at 31 characters; and \$VIEW() rejects multiple region names with a more appropriate error message
GTM-9343	-	Admin	MUPIP FTOK -JNLPOOL or -RECVPOOL show the entire path for the replication intance file
GTM-9344	-	Admin	MUPIP FREEZE (-ONLINE) with AIO freezes all regions at the same point in time
GTM-9346	-	Admin	Correct sleep period between soft tries connection attempts
GTM-9347	-	Language	Compiler optimization of out-of-range literals produces a NUMOLOW error

Id	Prior Id	Category	Summary
GTM-9349	-	Admin	GT.M installation script does not attempt to install semstat2 and ftok utilities.
GTM-9356	-	Admin	Remove dbcertify from GT.M distribution
GTM-9357	-	Other	🔴 ^%DSEWRAP removed from the distribution 🟢
GTM-9358	-	Admin	REPLICATE -SOURCE -ZEROBACKLOG - SHUTDOWN appropriately cleans up IPC resources
GTM-9361	-	Admin	🔴 ZEROBACKLOG considers acknowledged sequence number while determining the backlog 🟢
GTM-9362	-	Language	Fix for issue after failed compilation of indirection or XECUTE shown as a SIG-11 in UTF-8 mode \$TRANSLATE()
GTM-9368	-	Admin	A Source Server shutdown process exits with CTRL_C 🟢
GTM-9370	-	Language	All DIVZERO errors deferred to run-time
GTM-9371	-	Language	Ensure appropriate compilation of +<literal><non-relational-Boolean-operator><gvn @>

Database

- V7.0-000 creates database files that can hold up to 16Gi blocks; this requires a new database format using 64 rather than 32-bit pointers. The impact on storage efficiency is a function of the ratio of key size to pointers size. Prior versions used 32-bit block pointers and recent V6 versions supported a maximum of 992Mi Blocks. V7.000 supports databases created using V6, which appear the same within the global directory mapping. However, migrating existing data requires MUPIP EXTRACT and LOAD, which FIS recommends using a replicating instance to minimize down time. A future release will provide an upgrade path that does not require an extract and load. GT.M V7 does not support database files created with versions prior to V5. If upgrading from such a version, please use a V6.3-014 or other late V6 version as an intermediate step. (GTM-8913) 🍷🍏

This page is intentionally left blank.

Language

- VIEW and \$VIEW() processing truncates region names that exceed the maximum supported length - as of this writing 31; previously such an invalid name could cause the process to fail. This behavior was seen in in-house testing and was never reported by a customer. Also, \$VIEW() produces a VIEWREGLIST error when it detects an attempt to specify more than one region, as \$VIEW() does not accept region lists; previously it produced a NOREGION error for the second region even if that region existed. (GTM-9340)
- During compilation, GT.M reports NUMOFLOW errors for constants that exceed its numeric range. Such errors typically arise from use of the exponential ("E") notation. Starting in version V6.2-000, GT.M suppressed such errors, which could lead to incorrect results. (GTM-9347)
- \$TRANSLATE() appropriately handles UTF-8 mode processing; starting with V63.006 a recent prior compilation error could induce the process to fail with a segmentation violation (SIG-11). The root cause was a longstanding defect in GT.M management of memory in the transition from compilation back to execution and it is possible there were other symptoms. The workarounds are to avoid compilation errors in indirection and XECUTE or follow such an error with a successful compilation or a BREAK to direct mode. (GTM-9362)
- The GT.M compiler defers optimization of expressions containing modulo by zero (#0) or exponentiation of zero using a negative exponent (e.g. 0**-1) to run time, when, if executed, they produce DIVZERO errors. This is consistent with GT.M's treatment of divide by zero (/0) and integer divide by zero (\0). Note that except when it evaluates to match a recent and still cached object, XECUTE induces a mini-compilation with all detected errors deferred to run time in order to prevent double reporting. Starting in V6.3-001 the modulo divide by zero produced a segmentation violation (SIG11) when executed, and the negative exponentiation of zero typically failed in compilation with a GTMASSERT. (GTM-9370)
- GT.M appropriately compiles Boolean expressions of the form literal-operator-<2nd operand> where a plus-sign (+) precedes the literal, the operator is non-relational (AND/OR/NAND/NOR), and the second operand contains a global reference or indirection. The literal may not appear obvious because it may exist as an expression of literal elements, which the GT.M compiler reduces to a single literal; the global and/or indirection may appear anywhere in the right-hand operand or in later in an argument containing the Boolean. Since V6.3-001 such a construct tended to fail with a GTMASSERT, looping or out of memory during compilation; note that XECUTE might make the failure appear while executing at run time. (GTM-9371)

This page is intentionally left blank.

System Administration

- All command line arguments across GT.M utilities which previously accepted only decimal numbers as inputs, additionally accept hexadecimal numbers. The hexadecimal numbers must be prefixed with a 0x or 0X and the digits above nine are case insensitive. Furthermore, all the command line arguments across GT.M utilities which previously accepted only hexadecimal numbers as inputs continue to accept only hexadecimal numbers as inputs; however, FIS is deprecating the use of HEX values without a leading '0x' or '0X' and may cease to support them in the future. (GTM-6952) ✓
- The gtmprofile script checks dependencies if requested to run GT.M in UTF-8 mode, prints proper error messages and exits if any of the dependencies are not met. Previously it did not provide explicit error messages if dependencies were not met. (GTM-8262)
- The gtminstall script wrapper (gtminstall.sh) picks up the default ICU version of the system; previously it accepted ICU version specification with a --utf8 option. The manual gtminstall script (configure) no longer prompts for non-default ICU versions. When requesting UTF-8 support, if the dependencies are not met, the installation exits with an appropriate error. Previously, the installer accepted and prompted for non-default ICU versions and continued installation without UTF-8 support if the dependencies have not been met. Also, the gtmprofile script does not set gtm_icu_version on AIX platforms; previously, gtm_icu_version was unnecessarily set on AIX. (GTM-8263) ✓
- The gtminstall script (configure) explicitly removes setgid, setuid and sticky bits from the target installation directory if it exists with those bits set; previously the sub-directories created by the installation script inappropriately carried the setgid settings.(GTM-8288) ✓
- The gtminstall script (configure) records ownership, permissions and openssl sha256 checksum values of all installed files for future reference. (GTM-8517) ✓
- GT.M issues a JNLFLUSHLOG error message to the system log when the journal buffer flushing fails. Previously, GT.M issued only the BUFLUFAILED error message. (GTM-9268)
- The MUPIP REPLICATE -SOURCE -SHOWBACKLOG reports the sequence number acknowledged from the Receiver Server. The acknowledged sequence number from the replicating (secondary) instance can also be accessed from the originating (primary) instance using the "gtmsource_local_struct.heartbeat_jnl_seqno" field of the %PEEKBYNAME utility function. For example of the "gtmsource_local_struct.heartbeat_jnl_seqno" field, refer to the Additional Information section. Previously, MUPIP REPLICATE -SOURCE -SHOWBACKLOG reported information only from the Source Server which did not include an acknowledgement of updates that have reached the Receiver Server. (GTM-9302) ✓
- MUPIP FTOK -JNLPOOL or -RECVPOOL use the entire replication instance file path; previously it assumed the process current working directory. (GTM-9343)
- MUPIP FREEZE -ONLINE institutes the FREEZE on all regions in parallel to produce a consistent snapshot for a multiple-region database when ASYNCIO is ON. Previously, MUPIP FREEZE -ONLINE

with ASYNCIO froze each region serially which could result in an inconsistent snapshot with the final regions have sequence numbers higher than the first regions. (GTM-9344)

- During soft tries connection attempts, the Source Server waits for the specified soft tries period when it encounters a network error for the host name specified with the -SECONDARY qualifier. Previously, the Source Server skipped waiting which caused the Source Server log to record such attempts in quick succession. (GTM-9346)
- GT.M installation script does not attempt to install the deprecated semstat2 and flock utilities. When using existing directories the installation script force creates new soft links in utf8 directories if the destination files exist, previously creating such softlinks would throw a "File exists" error. When the installation fails, execute permissions of only the installed files are removed, previously execute permissions of even the directories were inappropriately removed. (GTM-9349)
- dbcertify, the tool necessary to upgrade a pre-V50000 database to the a V6 version, is no longer distributed. If you need to upgrade from a V4 major version to a current release, please upgrade to V63014 first and then to the current version. (GTM-9356)
- MUPIP REPLICATE -SOURCE -ZEROBACKLOG -SHUTDOWN clears all ipc resources associated with the replication instance file. Also, MUPIP REPLICATE -SOURCE -ZEROBACKLOG -SHUTDOWN -TIMEOUT displays "Initiating ZEROBACKLOG shutdown operation. Waiting for up to N seconds for backlog to clear" at the start of the command. Previously, MUPIP REPLICATE -SOURCE -ZEROBACKLOG -SHUTDOWN did not clear the ipc resources associated with the replication instance file and did not display the "Initiating ZEROBACKLOG shutdown operation" message. (GTM-9358)
- MUPIP REPLICATE -SOURCE -SHUTDOWN -ZEROBACKLOG accounts for the acknowledged sequence number received from the Source Server while determining whether the Source Server has backlog. It returns the REPL0BACKLOG message when there is no/zero backlog and the Receiver Server has acknowledged all sequence number updates or REPLBACKLOG when there is a backlog, unacknowledged updates, or the timeout expires; previously it did not include the sequence number acknowledgement from the Receiver Server to confirm that there is no backlog which may lead to in-flight updates and generate a lost transaction file during a switchover.

To reduce the possibility of a generating a lost transaction file during switchover, FIS recommends using -SOURCE -SHUTDOWN -ZEROBACKLOG and a check for the REPL0BACKLOG message during a planned switchover to help confirm that there is no backlog, all in-flight updates have reached the Receiver Pool.

The maximum -TIMEOUT specified with MUPIP REPLICATE -SOURCE -SHUTDOWN (with or without -ZEROBACKLOG) is 3600 seconds (1 hour). MUPIP produces the INVSHUTDOWN error if the -TIMEOUT is higher than the maximum shutdown timeout. If -TIMEOUT is not specified, the default timeout for MUPIP REPLICATE -SOURCE -SHUTDOWN is 120 seconds. Previously, the maximum timeout for both these cases was 30 seconds and MUPIP automatically adjusted the timeout downwards to 30 seconds when a higher timeout was specified. (GTM-9361) 🟢🟢

- A Source Server shutdown process (started with or without -zerobacklog) exits gracefully with CTRL_C; previously the Source Server did not respond to CTRL_C during the shutdown process. (GTM-9368) 🟢

Other

- ^%DSEWRAP, which was deprecated since V6.3-001, is removed from the release package. Please consider using MUPIP DUMPFHEAD, %PEEKBYNAME or DSE DUMP -FILEHEADER as alternatives to ^%DSEWRAP. (GTM-9357) 🍷 🍏

This page is intentionally left blank.

Additional Information

Additional Information about GTM-9302

Here are some examples of the new `gtmsource_local_struct.heartbeat_jnl_seqno` `^%PEEKBYNAME` field.

This routine returns the replication speed, that is the number of seqno updates per second acknowledged by the replicating instance during heartbeat intervals.

Example:

```
replspeed
; This routine returns the replication speed, that is the number of seqno updates per
second acknowledged by the replicating instance during the heartbeat intervals.
; Usage: $gtm_exe/mumps -r ^replspeed INSTANCE3 20
; The second parameter is the sampling size
set $etrap="write ""REPLSPEED-E-ACKSMPL : unable to fetch sampling data due to """,$zstatus
halt "
set $ztimeout="300:write ""Timeout occurred out after 5 minutes"",! zwrite halt"
new hrtbtperiod,instance,samplingsize,hrtbts,slot,i,hrtbtdiffs,diff,dump
set instance=$piece($zcmdline," ",1),slot=0
set samplingsize=$piece($zcmdline," ",2),hrtbtdiffs=0
set:$length(samplingsize)=0 samplingsize=10
if '($length(instance)) write "REPLSPEED-E-ARGS : ",$zcmdline," was specified. This routine
requires specifying an instance name.",! halt
for i=0:1:15 set instname=$$^
%PEEKBYNAME("gtmsource_local_struct.secondary_instname",i),instname=$piece(instname,
$char(0),1) set:(instname=instance) slot=i
; capture heartbeat_jnl_seqno samplingsize times. Wait for hrtbtperiod after every capture
of heartbeat_jnl_seqno
set hrtbtperiod=$piece($$^%PEEKBYNAME("gtmsource_local_struct.connect_parms",slot)," ",5)
for j=1:1:samplingsize do
. set hrtbts(j)=$$^%PEEKBYNAME("gtmsource_local_struct.heartbeat_jnl_seqno",slot,"I")
. do:(j>1)
. . set diff=hrtbts(j)-hrtbts(j-1)
. . do:(diff<0)
. . . write "REPLSPEED-E-ACKSEQNO : acknowledgement sequence number received is lower than
previous acknowledgement seqno."
. . . set dump="replspeed.dump" open dump use dump zwrite hrtbts close dump
. . . halt
. . set hrtbtdiffs=hrtbtdiffs+diff
. hang hrtbtperiod
set hrtbtdiffs=hrtbtdiffs/samplingsize
write $justify(hrtbtdiffs/hrtbtperiod,0,0)
quit
```

Use the following example confirm that an update corresponding to a Source Server sequence number has reached the Receiver Pool. This can be used as a tool to help confirm that there are no in-flight updates.

Example:

```
waitforhrtbt
; This routine returns 0 when the acknowledged seqno from the specified instance matches or
; exceeds the specified seqno
; If there is no confirmation (network issues etc) from the Receiver Server for 300
seconds, this routines returns 1.
; Usage: $gtm_exe/mumps -r ^waitforhrtbt <instname> <seqnocheckpoint>
set $etrap="write ""WAITFORHRTBT-E-ERROR, Error occurred while waiting for ackseqno
confirmation due to """,$zstatus halt "
new heartbeatseqno,hangduration,instance,checkseqno,i,instname,slot
; set $ztimeout to align with the REPLALERT threshold for the test system
set $ztimeout="900:write 1 halt"
set hangduration=1,slot=""
set instance=$piece($zcmdline," ",1)
set checkseqno=$piece($zcmdline," ",2)
if '($length(instance)&$length(checkseqno)) write "WAITFORHRTBT-E-ARGS : ",$zcmdline," was
specified. This routine requires specifying instance name and seqno.",! halt
for i=0:1:15 set instname=$$^
%PEEKBYNAME("gtmsource_local_struct.secondary_instname",i),instname=$piece(instname,
$char(0),1) set:(instname=instance) slot=i
if '($length(slot)) write "WAITFORHRTBT-E-INSTNOMATCH : No matching instance name for
",instance,! halt
for do
. set heartbeatseqno=$$^
%PEEKBYNAME("gtmsource_local_struct.heartbeat_jnl_seqno",slot,"I")-1
. if (heartbeatseqno>=checkseqno) write 0 halt
. hang hangduration
quit
```

Example:

```
waitforseqnosync
; This routine returns 0 when there the sequence numbers of the Source Server and the
; acknowledged sequence number from the Receiver Server is the same.
; It returns a non-zero value when there is no confirmation of the receipt of the latest
seqno from the secondary even when there is no backlog.
; If there is no confirmation (network issues etc) from the Receiver Server for 150
seconds, this routines returns 1.
; Usage: $gtm_exe/mumps -r ^waitforseqnosync <instname>
set $etrap="write ""WAITFORSEQNOSYNC-E-SRCBACKLOG : unable to get current Source Server
backlog and seqno updates status due to """,$zstatus halt "
new readseqno,heartbeatseqno,instance,i,instname,slot,hrtbtperiod
set $ztimeout="150:write 1 halt"
set slot=""
; hrtbtperiod: the heartbeat period (the fifth parameter of -CONNECTPARAMS)
set instance=$zcmdline
```

```
if '$length(instance) write "WAITFORSEQNOSYNC-E-ARGS : This routine requires specifying an
instance name.",! halt
for i=0:1:15 set instname=$$^
%PEEKBYNAME("gtmsource_local_struct.secondary_instname",i),instname=$piece(instname,
$char(0),1) set:(instname=instance) slot=i
if '($length(slot)) write "WAITFORSEQNOSYNC-E-INSTNOMATCH : No matching instance name for
",instance halt
set hrtbtperiod=$piece($$%PEEKBYNAME("gtmsource_local_struct.connect_parms",slot),"",5)
for do
. set seqno=$$%PEEKBYNAME("jnlpool_ctl_struct.jnl_seqno","", "I")
. set readseqno=$$%PEEKBYNAME("gtmsource_local_struct.read_jnl_seqno",i, "I")
. set heartbeatseqno=$$%PEEKBYNAME("gtmsource_local_struct.heartbeat_jnl_seqno",i, "I")
. if (seqno=readseqno=heartbeatseqno) write 0 halt
. hang hrtbtperiod
quit
```

This page is intentionally left blank.

Error and Other Messages

BOVTMGTEOVTM

BOVTMGTEOVTM, Journal file xxxx has beginning timestamp aaaa greater than end timestamp bbbb

MUPIP Error: This indicates that the beginning time stamp aaaa of the journal file xxxx is greater than the ending timestamp bbbb. This could be due to something that changed the system time, such as a daylight savings time change or a testing time reset, while GT.M was journaling. FIS recommends against changing system time during GT.M Run-time as a matter of course, as this disruption is not heavily tested.

Action: Confirm the reason for the time shift and decide whether to continue processing the affected journal files.

DBADDRANGE

DBADDRANGE, Database file rrrr element location aaaa: control vvvv was outside qqqq range bbbb to tttt

Run Time Information: This indicates a database control structure for database region rrrr at memory location aaaa contains a value vvvv outside range bbbb to tttt for quantity qqqq.

Action: This typically indicates a process terminated abnormally while updating the database. GT.M often fixes such an error unless there is a serious problem causing this error. If GT.M cannot correct the issue, the accompanying messages should expand on the situation; and you should report such database error to the group responsible for database integrity at your operation.

DBADDRANGE8

DBADDRANGE8, Database file rrrr element location aaaa: control vvvv was outside qqqq range bbbb to tttt

Run Time Error: This indicates a database control structure for database region rrrr at memory location aaaa contains a value vvvv outside range bbbb to tttt for quantity qqqq. This message is the same as a DBADDRANGE message except that vvvv, bbbb and tttt are 8-byte quantities (as opposed to 4-byte quantities in DBADDRANGE).

Action: This typically indicates a process terminated abnormally while updating the database. GT.M often fixes such an error unless there is a serious problem causing this error. If GT.M cannot correct the issue, the accompanying messages should expand on the situation; and you should report such database error to the group responsible for database integrity at your operation.

GTMCURUNSUPP

GTMCURUNSUPP, The requested operation is unsupported in this version of GT.M

All GT.M Components Error: GT.M tried to perform an operation that is unsupported in the current version. Currently this is only thrown by GT.M when trying to perform an upgrade/downgrade operation.

Action: GT.M does not currently support upgrade/downgrade between V6 and V7 databases. This feature will be added in a future release.

HEX64ERR

HEX64ERR, Error: cannot convert VVVV value to 64 bit hexadecimal number

All GT.M Components Error: The entered value does not correspond to a valid hexadecimal number representable in no more than 64 binary digits.

Action: Enter an appropriate hexadecimal value starting with 0X, using decimal integers 0-9 and ASCII letters A-F.

HEXERR

HEXERR, Error: cannot convert VVVV value to hexadecimal number

All GT.M Components Error: The entered value does not correspond to a valid hexadecimal number.

Action: Enter an appropriate hexadecimal value starting with 0X, using decimal integers 0-9 and ASCII letters A-F.

INVSHUTDOWN

INVSHUTDOWN, Shutdown timeout should be from 0 to 3600 seconds

MUPIP Error: This error appears when the -TIMEOUT specified with -SOURCE -SHUTDOWN exceeds 3600 seconds (1 hour).

Action: Specify -TIMEOUT between 0 to 3600 seconds

NUM64ERR

NUM64ERR, Error: cannot convert VVVV value to 64 bit decimal or hexadecimal number

All GT.M Components Error: The entered value does not correspond to a valid decimal number or hexadecimal number representable in no more than 64 binary digits.

Action: Enter an appropriate decimal value or hexadecimal value starting with 0X, and using decimal integers 0-9 and ASCII letters A-F.

NUMERR

NUMERR, Error: cannot convert VVVV value to 64 bit decimal or hexadecimal number

All GT.M Components Error: The entered value does not correspond to a valid decimal number or hexadecimal number.

Action: Enter an appropriate decimal value or hexadecimal value starting with 0X, and using decimal integers 0-9 and ASCII letters A-F.

REPL0BACKLOG +

REPL0BACKLOG, Total backlog for the specified replicating instance(s) is 0

MUIP Success: This message indicates a successful -ZEROBACKLOG -SHUTDOWN. It means that was no backlog for the specified replicating instance(s), no inflight updates, and all updates were successfully acknowledged by the Receiver Server.

Action: None.

REPLBACKLOG +

REPLBACKLOG, Timeout occurred while there was a backlog

MUIP Error: This error occurs when the -TIMEOUT specified with -SOURCE -ZEROBACKLOG -SHUTDOWN expires and there is either a backlog and/or there was a failure to receive an acknowledgment of the latest sequence number on the Source Server by the Receiver Server. If REPLNORESP also accompanies this error, it means that the Source Server did not receive a response from the Receiver Server for acknowledgement sequence number confirmation.

Action: This error means that the -ZEROBACKLOG checks did not pass. Restart the Source Server to clear any backlog. The presence of REPL0BACKLOG success message for -ZEROBACKLOG -SHUTDOWN confirms that there are no inflight updates and all updates are acknowledged by the Receiver Server.

REPLNORESP +

REPLNORESP, No sequence number confirmation from the replicating instance xxxx after waiting for nnnn second(s)

MUIP Warning: This message appears when the Source Server fails to receive a response from the Receiver Server during a -ZEROBACKLOG -SHUTDOWN. The presence of a REPLNORESP indicates that a -ZEROBACKLOG -SHUTDOWN check failed. This warning is accompanied by the REPLBACKLOG error message.

Action: This warning means that the -ZEROBACKLOG checks did not pass. Restart the Source Server to clear any backlog. The presence of REPL0BACKLOG success message for -ZEROBACKLOG -SHUTDOWN confirms that there are no inflight updates and all updates are acknowledged by the Receiver Server.

SHUT2QUICK +

SHUT2QUICK, Shutdown timeout ssss shorter than the heartbeat period SSSS; cannot confirm the backlog at the replicating instance iiiii

MUIP Warning: This warning appears when the -TIMEOUT=ssss specified with -ZEROBACKLOG is less than the heartbeat period SSSS (the fifth parameter of -CONNECTPARAMS). If -TIMEOUT is less than the heartbeat period, -ZEROBACKLOG cannot confirm that there is zero backlog as it cannot obtain the acknowledgement of the latest sequence number from the Receiver Server of instance iiiii in such a short time.

Action: Specify -TIMEOUT that is larger or equal to the heartbeat period.

UNUM64ERR +

UNUM64ERR, Error: cannot convert VVVV value to 64 bit unsigned decimal or hexadecimal number

All GT.M Components Error: The entered value does not correspond to a valid unsigned decimal number or hexadecimal number representable in no more than 64 binary digits.

Action: Enter an appropriate decimal value or hexadecimal value starting with 0X, and using decimal integers 0-9 and ASCII letters A-F.

VIEWREGLIST +

VIEWREGLIST, \$VIEW() only handles the first region subparameter

Run Time Warning: \$VIEW() with a region subparameter only operates on a single region. This differs from the VIEW command which has similar arguments and accepts region-lists for regions. This error is a warning and the function attempts to act on the first region.

Action: If the requirement is for multiple regions, use multiple \$VIEW() invocations, perhaps in a loop.

WCSFLUFAIL +

WCSFLUFAIL, Error flushing buffers -- called from module MMMM at line LLLL

All GT.M Components Error: This indicates that an attempt to flush a buffer to disk failed.

Action: If the error persists, report this database error to the group responsible for database integrity at your operation.