

Manual GST3

Contents

1	Introduction	2
2	Architecture	2
3	Installation	3
3.1	Platforms	3
	Debian/Ubuntu	3
	RedHat/CentOS	3
	SLES	4
3.2	SSL Certificate	4
	Certificate creation	4
3.3	Database	7
	PostgreSQL	7
	Oracle	8
	SQLite	8
	Database change	9
3.4	Configuration	10
4	Starting the Server	15
4.1	Manual script	15
4.2	Systemd	15
5	Backup	16
5.1	Backup	16
5.2	Recovery	16
6	GST Server: Maintenance Mode	17
6.1	Maintenance Mode: TUI and CLI	17
7	Maintenance Mode functions	17
7.1	Reset Administrator Password	17
7.2	Vacuum Database	17
7.3	Check Database for Orphaned Tiles	18
7.4	Check Storage Directory for Orphaned Files	18
7.5	Delete Orphaned Database Tiles	18
7.6	Delete Orphaned Files from Storage Directory	18
7.7	Track GST Server Processes	18
8	Running GST Server Maintenance Mode	18
8.1	List of CLI Command Arguments	19

8.2 GST Server Maintenance Mode Error Codes 19

1 Introduction

GST3 is a major upgrade to GST2. It provides the necessary means not only to be faster but also to allow storage of more geometry types before especially including gridded geometries like Voxets.

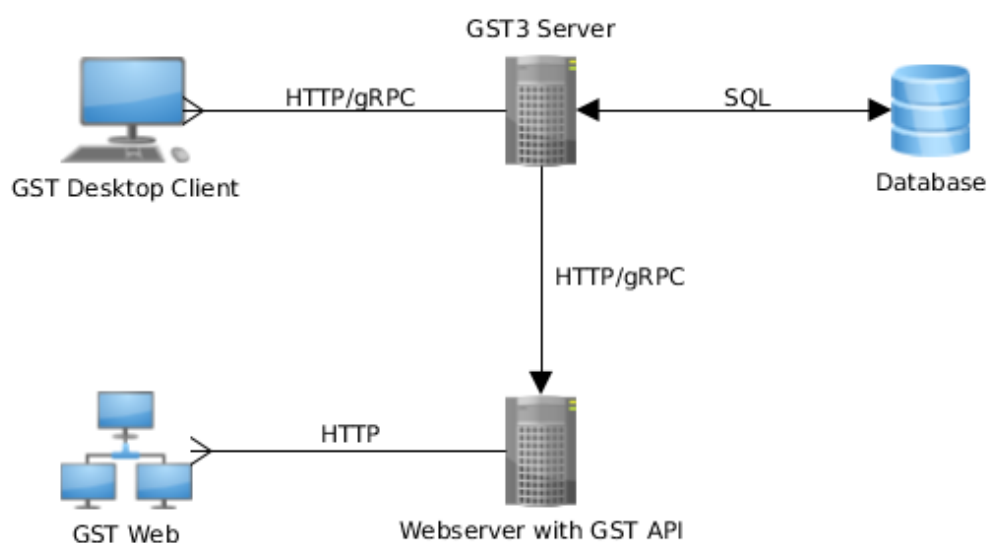
In order to be able to improve the existing GST2 in so many ways GST is not solely relying on a relational database management system[RDBMS] anymore. Nevertheless an RDBMS is being used in order to link other data coming from other databases and thus maintaining one of GST core features.

Additionally GST3 was written in Rust in order to use the features of Rust programming language over C++ which are for example: easy dependency management, a decent compiler, preventing memory leaks etc.

2 Architecture

The basic architecture of GST3 is displayed in figure.

Figure 2.1
Basic architecture of
GST3 full stack.



The core of the architecture of GST3 is the GST3 Server. It will store the geometries on the hard disk and keep the database up to date in order to maintain links. Furthermore the database is being used for basic user management. There also versions of GST3 which work with an internal SQLite database and thus do not need a special database server. The communication between GST3 and the database is based on the native protocols and SQL.

The actual interface to interact with application like GST Desktop or GST Web is using HTTP as protocol and on top of the gRPC. This allows GST3 to be used directly in many scenarios without the actual need of the GST C++ API.

GST Desktop itself is an application to be used by the end users. It provides a GUI in order to manage data within GST.

GST Web can be used in order to display data or models in 3d to people lacking access to sophisticated 3d modelling environments.

3 Installation

There are different ways for the installation of GST Storage since it can be deployed on several systems and comes in a variety of packages. Nevertheless some of the configuration after an installation is the same for all kinds of installation.

3.1 Platforms

Depending on your system please go on with the according section for your platform.

- * Debian/Ubuntu page 3
- * RedHat/CentOS page 3
- * SLESpage 4

Debian/Ubuntu

All of the afore mentioned components are coming in distinct packages and can be simply installed with by issuing a

```
# dpkg -i GST3-server-3.0.0.deb
```

command on the terminal.

RedHat/CentOS

All of the afore mentioned components are coming in distinct packages and can be simply installed with by issuing a

```
# rpm -ivh GST3-server-3.0.0.rpm
```

command on the terminal.

Note

Since with the RPM we cannot install the systemd script, this needs to be done manually. Please refer to [4.2](#) on how to install the systemd script.

SLES

All of the afore mentioned components are coming in distinct packages and can be simply installed with by issuing a

```
# rpm -ivh GST3-server-3.0.0.rpm
```

command on the terminal.

Note

Since with the RPM we cannot install the systemd script, this needs to be done manually. Please refer to [4.2](#) on how to install the systemd script.

3.2 SSL Certificate

In order to secure the communication via any client and the GST Server SSL encryption is being enforced. This means the GST Server needs to posses a valid SSL certificate and any client (web or desktop) needs the respective counterpart. The following steps describe how SSL certificates can be issued. However sometimes there are already certificates in place which can be used.

Note

We are going to use **openssl** for the creation of SSL certificates. We are not going to dive into all specifics and explain only the most important options. However, we recommend to consult the OpenSSL manual and read about the options.

Certificate creation

We recommend to create an **openssl.cnf** with the following content:

```
[ v3_api ]
basicConstraints = critical ,CA:false
keyUsage = nonRepudiation , digitalSignature
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always , issuer:always
subjectAltName = @alt_names

[ alt_names ]
DNS.1 = <hostname.domainname>
DNS.2 = localhost
```

This file allows to define several names which can be used to address the server. This can be handy if you use several hostnames to address the same server. If needed just add more **DNS.<number> = <hostname>** at the end of the file.

Note

Replace **<hostname.domainname>** with your FQDN.

Afterwards an unsigned SSL certificate can be easily produced on a server with the following command:

```
# openssl req \  
-nodes \  
-x509 \  
-days 365 \  
-newkey rsa:4096 \  
-keyout ca.key \  
-out ca.crt \  
-sha256 \  
-batch \  
-subj "/CN=<Your institute here>"
```

The above command issues a new key and saves this key in **ca.key**. The certificate will be saved in **ca.crt**. The certificate will be valid for 365 days.

Note

Replace **<Your institute here>** with your institution name.

Note

The above issued certificate is valid for 365 days only. This means you will need to create a certificate every year. However, you can change the validity period easily.

Do this only if you are aware of the security implications of this.

After issuing the **openssl** command you should see the following output and obtain two new files **ca.crt** and **ca.key**.

```
Generating a RSA private key  
.....++++  
writing new private key to 'ca.key'
```

Security Advice

Do not give anyone access to your **ca.key.**

This would allow others to compromise the communication between GST Server and other software.

3. Installation

In the following we are going to create a signing request with

```
# openssl req \  
-nodes \  
-newkey rsa:4096 \  
-keyout server.key \  
-out server.req \  
-sha256 \  
-batch \  
-subj "/CN=<Your domain>"
```

The output should look like

```
Generating a RSA private key  
.....+++++  
.....+++++  
writing new private key to 'server.key'
```

The resulting files are **server.key** and **server.req**.

Note

Replace **<Your domain>** with your domain.

Security Advice

Do not provide **server.key to anyone.**

This would allow others to compromise the communication between GST Server and other software.

In the final step we will use

```
# openssl x509 \  
-req \  
-in server.req \  
-out server.crt \  
-CA ca.crt \  
-CAkey ca.key \  
-sha256 \  
-days 3650 \  
-set_serial 01 \  
-extensions v3_api \  
-extfile openssl.cnf
```


3. Installation

The output should look like

```
Signature ok
subject=CN = <Your domain>
Getting CA Private Key
```

and you should obtain a new file **server.crt**.

After creating the certificates you can delete the **cert.key** and **server.req**. This means they can not be used to export more certificates with both.

Client Certificates

Please make sure that all your possible clients have access to the certificate file **ca.crt**. Especially for GST this means that there needs to be copy for GST Web and a copy for GST Desktop which needs to be installed on the Desktop computers.

3.3 Database

GST3 needs a database to store metadata. The database can be either:

- * PostgreSQL page 7
- * Oraclepage 8
- * SQLite page 8

It should be noted, that GST accesses the database with multiple connections but only with one single database user. Also while using GST there are no new database users being created.

Note

Whenever you make changes to database manually you may hurt GST's integrity!
Please only do so if instructed by our team or if stated here in the manual!

PostgreSQL

Everything up from Postgresql version 9.5 should work fine.

Note

It is advised to create a new database and also create a new user for **GST3**. However, you can also create a distinct schema on any existing database.

Note

Within the changes in Postgres 10.3 **GST3** should have also its own schema, since public schemata are now being prevented to be used or considered dangerous. [c.f. [PostgreSQL Wiki: CVE-2018-1058](#)]

Example:

Create a new database named gst3.

```
# psql -d postgres -c "CREATE DATABASE gst3"
```

Create a user gst3_user with password gst3_pwd

```
# psql -d gst3 -c "CREATE USER gst3_user PASSWORD 'gst3_pwd'"
```

Create a schema for gst3_user

```
# psql -d gst3 -c "CREATE SCHEMA IF NOT EXISTS gst3_user AUTHORIZATION  
gst3_user; "
```

The same database and user will be used in the following examples.

Oracle

It is advised to create a distinct user for GST. Within its on schema the database user should be able to create, alter, delete tables, indices, constraints and views.

The database user needs to be able to issue SELECT statements for the following Oracle system tables:

- * all_tables
- * all_tab_cols
- * all_constraints
- * all_cons_columns
- * all_views.

SQLite

If you do not want to use a large RDBMS you can also use the file based SQLite. Depending on the complexity and amount of data this can be as fast as any traditional RDBMS. Moreover, it can help to reduce network latency.

Note

Please note that SQLite has not been tested heavily in production yet. Additionally, SQLite lacks traditional checks and systems PostgreSQL and Oracle provide.

In order to use SQLite, you need a place on the filesystem where the SQLite database is located. This can also be the same data directory that is being used to the geometrical data. You need to make sure, that the user which runs the GST Server is allowed to read and write from that directory.

If there is no SQLite database in the directory present it will be created by GST Server on the first startup.

Note

Backing up an SQLite database is as easy as copying the single file that is being used and archive it. If you do not place the database in your data directory you should consider to backup the database regularly too.

Database change

There are several reasons why a change of a database can happen (the list is not concluding):

- * update to a newer version
- * retirement of old database servers
- * paradigm shift to use a different database system.

Whenever this is scheduled you would need to make sure to have a proper backup of your data and database. Backups and recovery of GST's data are described in [5](#). After a successful change to a new database server or system you need to adopt the configuration file of GST Server. Additionally, a manual change of the table **constrained_urls** is needed. The old connection string needs to be updated to reflect the change.

Note

Do not delete any records from the table **constrained_urls**. This will hurt the integrity of GST's internal structure!

To update the table **constrained_urls** properly you may use the following command:

```
update constrained_urls set url='<new connection string >'
where url='<old connection string >';
```

3. Installation

Of course you need to make sure to replace and adjust **<new connection string>** and **<old connection string>** accordingly.

If you also changed the underlying database management system, e.g. you moved from PostgreSQL to Oracle you also need to change the **database_type** field.

```
update constrained_urls set database_type=3
where url='<new connection string >';
```

The following **database_type** are supported:

Database System Type	database_type
PostgreSQL	1
SQLite	2
Oracle	3

3.4 Configuration

You need to edit a configuration file before the first start of **gst-server**. It should be in the same directory as **gst-server** or in e.g.

```
# /etc/GiGa
```

and might be named **config.toml**. There are four required parameters that need to be edited before the first start of **gst-server**. They are **url** in the **[database]** section and **data_path** in the **[storage]** section and **cert_path** and **key_path** in the **[server]** section.

url

Connection URI to the database system.

All connection strings must be preceded with the database system type:

- * For PostgreSQL the prefix is: **postgres://**
- * For Oracle the prefix is: **oracle://**
- * For SQLite the prefix is: **sqlite://**

For PostgreSQL:

See [PostgreSQL documentation](#) section **Connection URIs** for more info on the so called connection string.

For Oracle:

See [Oracle documentation](#) for more info on the so called **Easy Connect Naming Method**.

For SQLite:

You only put the path to the database file.

Examples:

- * **postgres://user:password@server/database**
- * **oracle://user:password@server/service**
- * **sqlite:///path/to/sqlite/file.db**

Example:

```
url = "postgres://gst3_user:gst3_pwd@localhost/gst3"
```

data_path

A path to where the geometry data files will be stored. Make sure **gst-server** has write access in that path.

Example:

```
data_path = /opt/gst3/data
```

cert_path

A path where the SSL certificate for the server can be found.

3. Installation

Example:

```
cert_path = "/etc/GiGa/ssl/server.crt"
```

key_path

The path for the certificate key.

Example:

```
key_path = "/etc/GiGa/ssl/server.key"
```

The other parameters can be left as they are. After editing those four values your config.toml should look like the following.

Example:

```
[server]
# ip address the server listens on.
# optional [default = "::"]
# value range: valid ip4 or ip6 addresss
ip = "::"
# port the server listens on.
# optional [default = 50051]
# value range: 0–65536
port = 50051
# The path to the the ssl certificates used by the server
cert_path = "server.crt"
key_path = "server.key"
# The path to the gst-server license
license_path = "/etc/GiGa/license/"

[database]
# databse url to the gst database
url = "postgresql://gst3_user:gst3_pwd@localhost/gst3"
# A list of non-standard oracle system users. NB:
# standard system users are:
# SYS, MDSYS, APEX_040200, CTXSYS, SYSTEM, XDB, WMSYS,
# ORDDATA, GSMADMIN_INTERNAL
oracle_system_users = ["SUPER_SQUIRREL"]

[storage]
# tile size
tile_size = 25000
# path where the binary geometry data will be stored
data_path = '/opt/gst3/data'
# number of cores to use.
# optional [default = all]
cpu_count = 4

[environment]
# Optional path to an alternative GDAL data directory.
gdal_data = "/usr/GiGa/share/gdal-3.0.4-share/"
# Optional path to additional nadgrid files.
grid_path = "/usr/GiGa/share/proj-6.3.0-share/"
```

The other parameters will be explained in the following.

Configuration file parameters

[server]

- **ip** the IP address gst-server listens on for incoming connections.
- **port** the port number gst-server listens on for incoming connections.
- **cert_path** explained above.
- **key_path** explained above.
- **license_path** path to the license file for **GST** Server.

[database]

- **url** explained above.
- **oracle_system_user** A list of non-standard oracle system users. Standard system users are:
SYS, MDSYS, APEX_040200, CTXSYS, SYSTEM, XDB, WMSYS, ORDDATA, GSMADMIN_INTERNAL
- **pool_wait_timeout** An optional timeout given in milliseconds that defines how long the server should wait for a connection to the database to become available again before aborting. Setting no value uses no timeout.
- **pool_create_timeout** An optional timeout given in milliseconds that defines how long the server should wait when creating a new connection to the database before aborting. Setting no value uses no timeout.
- **pool_recycle_timeout** An optional timeout given in milliseconds that defines how long the server should wait when recycling a connection to the database before aborting. Setting no value uses no timeout.

[storage]

- **tile_size** Geometry data is stored in tiles [one file per tile]. This parameter sets how many points / cells are stored per tile. The given default [25000] was determined by our benchmarks and should be good enough. Sensible values are in the range between 10000 and 1000000.
- **data_path** explained above.
- **cpu_count** numbers of CPUs/cores to be used for computation and or waiting for incoming connections.

[environment]

- **gdal_data** An optional path to additional gdal shared data. These files help to increase the precision of coordinate transformation.
- **grid_path** An optional path to additional grid shift files. These grid shift files help increase the precision of coordinate transformation.
- **internal_gdal_path** The path used to temporarily store gdal and proj configuration files bundled with GST-Borehole server. This setting is optional. If it's not set **\$TEMP_DIR/giga-infosystems-{randstr}** is used. The configured directory needs to be writeable for the gst-borehole-server application.

4 Starting the Server

4.1 Manual script

Run the following command from the install directory.

```
# /prog/GiGa/gst-server -config /prog/GiGa/config.toml > /var/log/gst.log 2>&1 &
```

4.2 Systemd

If you have installed **GST** Server with a package you may also have gotten a systemd startup script. These should be located at

```
# /lib/systemd/system
```

or on e.g. SLES

```
# /usr/lib/systemd/system
```

. With that you should be able to start **GST** Server via

```
# systemctl start gst.service
```

However if the file does not exist here is an example:

```
[Unit]
Description=GST3 Server Service

[Service]
Type=simple
ExecStart=/usr/GiGa/gst-server --config /etc/GiGa/config.toml
Restart=on-failure
User=gst
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=gst-server
LimitNOFILE=32768

[Install]
WantedBy=multi-user.target
```

This script assumes that:

- there is a distinguished user **gst** which has read access to the server certificate, the config.toml and write access to the data path
- the executable for **GST** Server has been installed at /usr/GiGa

Simply copy the content into a plain file at

```
# /lib/systemd/system/gst.service
```

or on e.g. SLES

```
# /usr/lib/systemd/system/gst.service
```

5. Backup

. Afterwards you can enable GST Server to be started whenever the system is being started up:

```
# systemctl enable gst.service
```

. You can now start GST with

```
# systemctl start gst.service
```

. After starting GST it is always advised to check the logs. With that you should get basic information if GST Server has been started successfully. You can obtain the logs either via

```
# systemctl status gst.service
```

or

```
# journalctl -xeu gst.service
```

.

If you see any error messages, you may need to fix your configuration file or adjust the rights accordingly.

5 Backup

5.1 Backup

We would advise you to do regularly backups. Most importantly you would need to backup the database and the data directory as configure in the configuration file of GST Server. It maybe handy to have a backup of the issued certificates and configuration too.

Note

There is currently no backup mechanism built-in in GST.

5.2 Recovery

To recover from a previous backup you first shutdown the GST server. Afterwards you would continue to restore the database. Please refer to you database manual to get to know how to do that. One of the last steps would be to copy the backed up data back to the data directory. Finally, you can start GST Server and all data should be restored.

Note

If there have been changes to the location of the data directory and especially to the database connection you need to make sure to update the configuration file accordingly. Moreover, a change of the database connection also needs a change on the database itself, c.f. [3.3](#).

6 GST Server: Maintenance Mode

GST sever maintenance mode is a feature of GST3 that allows administrators with access to the server to perform certain maintenance functions which are otherwise difficult, if not impossible, to perform safely. With the introduction of **Maintenance Mode** an administrator with access to the instance running GST Server can perform certain maintenance or recovery procedures, or set up a script to automate these tasks.

6.1 Maintenance Mode: TUI and CLI

The maintenance mode can be launched either as an interactive program in a terminal, or as a single command that runs to completion. Some operations are only available on one of the modes. For example checking the database for orphaned storage tiles can only be performed [on its own] in the CLI mode, while monitoring of server thread activity can only be carried out in the interactive TUI mode.

7 Maintenance Mode functions

- * Reset administrator password.
- * Vacuum database.
- * Check database for orphaned tile entries. [CLI mode only]
- * Check the storage directory for orphaned files. [CLI mode only]
- * Remove orphaned tile entries from database.
- * Remove orphaned files from storage directory.
- * Track GST Server processes. [TUI mode only]

7.1 Reset Administrator Password

Currently there is no easy way to reset the administrator password in GST. Here the administrator password can be reset to the default value.

7.2 Vacuum Database

Databases can become fragmented following multiple cycles of insertions and deletions, which affects performance. This allows the administrator to 'vacuum', or clean up a **Postgresql** based GST3 database.

Note

Cleanup of **Oracle** based database systems must be done manually using the tools that Oracle provides.

7.3 Check Database for Orphaned Tiles

GST3 keeps an index of stored geometries in the database, while the files are kept in storage. While this should not affect normal operations, it is possible to end up with the index entries remaining when the stored geometry tiles have been deleted. This operation checks for such 'orphaned' entries in the database. This can be useful when trying to improve performance of the database.

7.4 Check Storage Directory for Orphaned Files

GST keeps an index of stored geometries in the database, while the files are kept in storage. In some cases, if file uploads are interrupted, it is possible to end up with files stored in storage without creation of database entries. This can happen also in case of a server crash. Such orphaned files can be found with this operation. Deleting such files can be useful when freeing space in storage.

7.5 Delete Orphaned Database Tiles

If the database contains orphaned references to tiles stored in the storage directory, they can be found and removed with this operation. [See [Check Database for Orphaned Tiles](#)]

7.6 Delete Orphaned Files from Storage Directory

If the storage directory contains orphaned files, they can be removed with this operation. [See [Check Storage Directory for Orphaned Files](#)]

7.7 Track GST Server Processes

On occasion it may be desirable to monitor the processes launched by GST Server in real time. While this can be accomplished using a task manager, GST Server maintenance mode offers its own tool to accomplish this.

8 Running GST Server Maintenance Mode

GST Server maintenance mode can be launched in TUI mode by appending the 'maintenance' subcommand after the usual command to run GST Server.

Example:

```
# gst-server -config=/path/to/config.toml maintenance
```

Alternatively by appending a command argument after maintenance, it is possible to run a maintenance command non-interactively. For example one can append **--db-vacuum** to initiate the process of vacuuming the database.

Example:

Reset the admin password.

```
# gst-server -config=/path/to/config.toml maintenance --reset-admin-password
```

Note

Only **one** command argument may be used at a time.

8.1 List of CLI Command Arguments

- * Reset administrator password: **--reset-admin-password**
- * Vacuum database: **--db-vacuum**
- * Check database for orphaned tile entries: **--check-db-orphans**
- * Check the storage directory for orphaned files: **--check-storage-orphans**
- * Remove orphaned tile entries from database: **--delete-db-orphans**
- * Remove orphaned files from storage directory: **--delete-storage-orphans**

8.2 GST Server Maintenance Mode Error Codes

The maintenance mode of GST Server is designed specifically to allow automation of maintenance. Therefore, it comes with a number of custom exit codes. These can be useful when making automated maintenance scripts. Below is a table detailing these error codes. NB: Most of these codes are not likely to be seen in the current version of maintenance mode.

Code	Error type	Error Description
------	------------	-------------------

1	General error	Usually caused by incorrect arguments fed into the program, but other causes cannot be excluded.
2	Migration error	Migrations could not be run when starting.
3	Gdal error	Usually associated with transformation of geometries. Not likely to be seen in maintenance mode.
4	StdParse error	An input could not be converted into a number. Unlikely to be seen.
5	Stdin error	This means the program cannot take data from the standard input. Usually indicates a system problem.
6	Maintenance UI error	An error in the TUI. Should not be possible in maintenance CLI.
7	Diesel error	Indicates an underlying problem in database handling.
8	Storage error	Indicates a problem in the operation of GST Storage. Not possible in the current maintenance mode.
9	Writer error	An error writing a geometry. Not currently possible in maintenance mode.
10	Parser error	An error parsing a geometry. Not currently possible in maintenance mode.
11	Network error	An error in communication between server and client. Not currently possible in maintenance mode.
12	Geometry error	An error operating on a geometry. Currently not possible in maintenance mode.
13	Message	A custom error. Cause can only be elucidated from the human-readable text output.
51	.conf file parse error	An error parsing the configuration file
52	.conf file IO error	Configuration file could not be read on startup
53	.conf file net error	An error parsing the IP address in the config
54	.conf related error	Assorted errors related to the config file
98	Other	Unaccounted error types. This should not be triggered.
99	Other	Unaccounted error types. This should not be triggered.
101	Internal Server error	Assorted errors. See the error message for more details. Should not be possible in maintenance mode.
102	User not found	User not found in database.
103	Invalid login	An invalid login. Should not be possible in maintenance mode.
104	Geometry not found	A geometry is not found in the DB. Should not be possible in maintenance mode.
105	Geometry version not found	A geometry version is not found in the DB. Should not be possible in maintenance mode.

8. Running GST Server Maintenance Mode

106	Feature class not found	A feature class is not found. Should not be possible in maintenance mode.
107	Feature attribute alias not found	Not relevant in maintenance mode.
108	Feature property alias not found	Not relevant in maintenance mode.
109	SRS not found	Not relevant in maintenance mode.
110	Coordinate system not found	Not relevant in maintenance mode.
111	Level of detail not found	Not relevant in maintenance mode.
112	Property Not found	Not relevant in maintenance mode.
113	Invalid Uuid	Not relevant in maintenance mode.
114	Lock not found	Not relevant in maintenance mode.
115	Area locked	Not relevant in maintenance mode.
116	Feature locked in feature class	Not relevant in maintenance mode.
117	Same SRS Given	Not relevant in maintenance mode.
118	Cannot lock subgeometry	Not relevant in maintenance mode.
119	Property mismatch	Not relevant in maintenance mode.
120	Attribute not found	Not relevant in maintenance mode.
121	Group not found	Not relevant in maintenance mode.
122	Dynamic color scale not found	Not relevant in maintenance mode.
123	Colormap not found	Not relevant in maintenance mode.
124	Unknown srs format	Not relevant in maintenance mode.
125	Node link not found	Not relevant in maintenance mode.
131	Element not found	Not relevant in maintenance mode.
132	Element property not found	Not relevant in maintenance mode.
133	Element name not unique	Not relevant in maintenance mode.
134	No selector given	Not relevant in maintenance mode.
135	Invalid scale type	Not relevant in maintenance mode.
136	Moma Error	Not relevant in maintenance mode.
137	Moma Error	Not relevant in maintenance mode.
138	Moma Error	Not relevant in maintenance mode.
139	Feature class not empty	Not relevant in maintenance mode.
140	Column not found	Not relevant in maintenance mode.
141	Constrained attribute not found	Not relevant in maintenance mode.
142	No valid license	Not relevant in maintenance mode.
143	Commit not found	Not relevant in maintenance mode.
144	Safe area width missing box	Not relevant in maintenance mode.
145	Invalid safe are width	Not relevant in maintenance mode.
146	Access denied	Not relevant in maintenance mode.

8. Running GST Server Maintenance Mode

147	Db/Storage mismatch	High probability that the storage directory specified in the config file is incorrect [none of the files in the storage directory match up to the DB tiles.]
148	Admin not found	The default admin user ``admin" could not be found on the DB.
201	R2D2 Error	An error communicating with the database. Can occur when the address occurs.
202	IO error	Database related input/output error.
203	Database Error	An error within the database.
204	JSON error	Not relevant in maintenance mode.
205	Access denied	Access to the database is denied.
206	Attribute Type Mismatch	Not relevant in maintenance mode.
207	Feature class error	Not relevant in maintenance mode.
208	Id not found	Not relevant in maintenance mode.
209	Missing constrained value	Not relevant in maintenance mode.
210	Invalid constrained attribute key	Not relevant in maintenance mode.
211	Missing constrained database connection	Not relevant in maintenance mode.
212	Missing constrained database	Not relevant in maintenance mode.
213	Missmatched property value error	Not relevant in maintenance mode.
214	Poisoned error	Not relevant in maintenance mode.
215	Area already locked	Not relevant in maintenance mode.
216	Geometry fully locked	Not relevant in maintenance mode.
217	Cannot lock subgeometry	Not relevant in maintenance mode.
218	Cannot lock geometry fully	Not relevant in maintenance mode.
219	Geometry not found	Not relevant in maintenance mode.
220	Geometry not found	Not relevant in maintenance mode.
221	Version not found	Not relevant in maintenance mode.
222	Nan coordinate insert	Not relevant in maintenance mode.
223	Migration error	Migrations could not be run when starting up.
254	Assorted error	Miscellenious errors. The exact error can be determined from the message.