



Database Information Guide

GWGuardian Database Information Guide

Getting Started

About GWGuardian

GWGuardian is the perfect solution for enterprise security needs: it provides superior spam and virus protection while maximizing return on investment. Layered with intelligence, it pre-authenticates all legitimate users before scanning for spam. It auto-updates virus definitions 24/7 and provides 24/7 updated spam protection. Also included is a multi-layered approach to prevent spammers from abusing and exploiting your network assets.

Note: Virus protection is available in GWGuardian Enterprise Edition and GWGuardian Academic Edition.

Intended Audience

This manual is intended for the GWGuardian administrator. It explains how to configure, operate, and maintain the GWGuardian Web Administration interface. This manual assumes the user is fairly knowledgeable about common GroupWise and NetWare terms, concepts, and techniques.

Other Resources

Messaging Architects is committed to providing the best support available for GWGuardian. If your question is not answered in this manual, our website (www.messagingarchitects.com) includes additional information.

Feedback

If you have a general feature question or comment, please submit it at www.messagingarchitects.com/contactus.

Technical Support

If you have a technical support question, please consult the Technical Support section of our website at <http://www.messagingarchitects.com/support/> or call (514) 392-1303, and a member of our technical support team will contact you as soon as possible during regular business hours.

Sales

To contact a Messaging Architects sales team member, send an email to info@messagingarchitects.com, call **1-866-497-0101**, or complete the form at www.messagingarchitects.com/contactus and we will get in touch with you.

Other Issues

If you have any other questions, send email to info@messagingarchitects.com and we will contact you.

Corporate Headquarters

Messaging Architects
180 Peel, Suite 333
Montreal, QC Canada H3C 2G7
Tel: 514-392-9220
Fax: 514-392-9120

How to Use This Guide

This guide is intended to complement the GWGuardian Administration Console, and introduces concepts in the same order as the layout of the console. If you are looking for a particular function, take a look at the index found at the end of the guide.

Advice boxes

Throughout the guide, you will sometimes see special advice boxes. These advice boxes are intended to supplement the information presented in the section where they are found. These advice boxes serve different functions based on the icon used to represent them. The types of advice boxes are:



More information: This advice box tells you where to find more information pertaining to the current subject. Look for this advice box if you want to find out where else in the document a certain subject is being discussed.



Warning: This advice box lets you know when something requires caution. The goal of this advice box lets you know about the potential errors into which you might run when using the function in question.



Helpful tip: This advice box will inform you of advanced configuration tricks.

Table of Contents

1 Introduction	7
1.1 Database Formats	7
1.2 Creating a New Database	7
1.3 Running a Database Script	8
2 Creating an ODBC Connection.....	9
3 Creating and Configuring the Quarantine Database	11
4 Creating and Configuring the Monitoring Database.....	13
4.1 SQL Server	13
4.2 PostgreSQL.....	13
4.3 MS Access	14
5 Creating and Configuring the Sieve Database	15
5.1 SQL Server / SQL Express	15
5.2 PostgreSQL.....	15
5.3 MS Access	15
6 Creating and Configuring the Audit Database	17
6.1 SQL Server / SQL Express	17
6.2 PostgreSQL.....	17
7 Upgrading the Databases	19
7.1 Quarantine	19
7.2 Audit.....	19
8 Security and Maintenance	21
8.1 SQL Server Security Enhancements.....	21

1 Introduction

This document provides information necessary to create and configure the various databases required for use by GWGuardian. It also provides the scripts for upgrading GWGuardian. This document supplements the information in the *GWGuardian Installation and Configuration Guide*.

1.1 Database Formats

The following table provides the recommended database formats for GWGuardian:

Database	MS-SQL2000+	MS-SQL Express	PostgreSQL	MS Access and MSDE
Quarantine	√	√	√	√
Monitoring	√	√	√	√
Audit	√	√	√	x
Sieve	√	√	√	√

- MS Access and MSDE have a size limitation of 2GB
 - These are not recommended for large environments but are acceptable in small environments
 - Clients should consider using another database program as these have limitations and will not support future features
- SQL Express has a size limitation of 4GB
 - It can be used in medium-sized environments
 - This only applies to the Audit DB as it can grow fairly large
- MySQL can be used for the Quarantine database but is not recommended

1.2 Creating a New Database

The following section provides information for creating a new database:

SQL Server

1. Open the **SQL Enterprise Manager**.
2. Expand the tree and choose **Databases**.
3. Select **Action > New Database**.
4. In **Database Properties**, name the database. Under **Collation Name**, use the drop-down menu to select **Latin1_General_CI-AS**.



This ensures that the database remains case-insensitive.

5. Click **OK**.

PostgreSQL

1. Open the **pgAdmin III**.
2. Expand the tree and choose **Databases**.
3. Right-click on **Databases**, and select **New Database**.
4. When creating a Quarantine database, select **SQL_ASCII** for the encoding.
5. When creating an Audit database, select **UNICODE** for the encoding.

6. Click **OK**.

MS SQL Express

1. Open the **SQL Server Management Studio**.
2. Right-click **Databases**, and select **New Database**.
3. On the **General** panel, enter the name of the new database.
4. Click **Options**. Under **Collation Name**, use the drop-down menu to select **Latin1_General_CI-AS**.
5. Click **OK**.

1.3 Running a Database Script

The following provides basic information for running database scripts:

SQL Server

1. Start the **SQL Query Analyzer**.
2. Select the appropriate database.
3. Choose **File > Open** and go to the location of the scripts (C:\Program Files\Messaging Architects\GWGuardian\DBStructures\SQL Server).
4. Click on ► to run the scripts.

PostgreSQL

1. Open the **pgAdmin III**.
2. Expand the tree and select the appropriate database.
3. Click on **Tools > Query Tool**.
4. Choose **File > Open** and go to the location of the scripts (C:\Program Files\Messaging Architects\GWGuardian\DBStructures)
5. Click on ► to run the scripts.

MS SQL Express

1. Open the **SQL Server Management Studio**.
2. Choose **File > Open** and go to the location of the scripts (C:\Program Files\Messaging Architects\GWGuardian\SQL Server).
3. Select the database.
4. Click on **!Execute** to run the scripts.

2 Creating an ODBC Connection

The following provides information for creating an ODBC connection:

1. On the server, go to **Administrative Tools > Data Sources (ODBC)**.
2. Click on **System DSN**.
3. Click on **Add**.
4. Select the appropriate driver (SQL Server, PostgreSQL or MS Access) and click **Finish**.

SQL Server / SQL Express

1. Enter a name for the data source name.
2. Optionally, enter a description of the data source.
3. Select the SQL server from the drop-down menu and click **Next**.
4. Select the SQL Server authentication method and enter the login ID and password.
5. Click **Next**.
6. Change the default database to whichever database is being configured (e.g. Quarantine, Sieve, etc.).
7. Click **Next** and then **Finish**.

PostgreSQL

1. Enter a name for the data source name and a description of the data source.
2. Enter the name of the database being configured (e.g. Quarantine, Sieve, etc.).
 - Enter the name or IP address of the PostgreSQL server.
 - Enter the user name and password.
 - Click **Save**.

MS Access

1. • Enter the data source name and description of the data source.
2. • Click **Select** to choose the database being configured and click **OK**
3. • Click **OK**.
4. • Close the **ODBC Data Source Administrator panel**.

3 Creating and Configuring the Quarantine Database

The following provides information for creating the Quarantine database and configuring it in GWGuardian:

SQL Server / SQL Express

1. Create a new database
2. Run the following script, located in C:\Program Files\Messaging Architects\GWGuardian\DBStructures\SQL Server\Quarantine.
 - mssql_quarantine_without_imap.sql
3. Create the ODBC connection.
4. In the GWGuardian Administration Console, choose **System > Properties > Quarantine Database**.
5. Enter your ODBC information and click **Apply**.
6. Stop and start all Modus services in the GWGuardian Administration Console.

PostgreSQL

1. Create a new database.
2. Remember to select **SQL_ASCII** for the encoding .
3. Run the following script, located in C:\Program Files\Messaging Architects\GWGuardian\DBStructures\SQL Server\Quarantine.
 - pg_quarantine_without_imap.sql
4. Create the ODBC connection.
5. In the GWGuardian Administration Console, choose **System > Properties > Quarantine Database**.
6. Enter your ODBC information and click **Apply**.
7. Stop and start all Modus services in the GWGuardian Administration Console.

MS Access

1. The database is created automatically when GWGuardian is installed.
2. In the GWGuardian Administration Console, choose **System > Properties > Quarantine Database**.
3. Select **Use Native MDB Database**.
4. Click **Apply**.
5. Stop and start all Modus services in the GWGuardian Administration Console.

4 Creating and Configuring the Monitoring Database

The following provides information for creating the Monitoring database and configuring it in GWGuardian:

4.1 SQL Server

1. Create a database.
2. Open a Command Prompt on the SQL server.
3. Go to C:\Program Files\Messaging Architects\DBStructures\SQL Server\Monitoring.
4. Run **CreateMonitoringSQL.exe** with the following parameters:
 - createmonitoringsql <server> <database> <username> <password>
5. Create the ODBC connection.
6. In the GWGuardian Administration Console, choose **System > Properties > Monitoring Database**.
7. Select **Use ODBC Database** and enter your ODBC information
8. Click **Apply**.
9. Stop and start all Modus services in the GWGuardian Administration Console.

4.2 PostgreSQL

If you answered **yes** to install PostgreSQL during the initial GWGuardian installation, the Monitoring database was automatically created:

1. In the GWGuardian Administration Console, choose **System > Properties > Monitoring Database**.
2. Select **Use Native Postgres Database** and enter the appropriate information.
3. Click **Apply**.
4. Stop and start all Modus services in the GWGuardian Administration Console.

If you answered **no** to install PostgreSQL during the initial GWGuardian installation, ensure that version 8.0 or higher is installed:

1. The necessary files can be found in the C:\Program Files\Messaging Architects\GWGuardian directory:
 - moduspgmon.dll
 - Monitor_pg.sql
2. Create the GWGuardian tables in your PostgreSQL database and link GWGuardian to it:
 - Extract the moduspgmon.dll file to ... \PostgreSQL\8.0\lib
 - From a Command Prompt, go to C:\Program Files\Messaging Architects\GWGuardian and type the following commands:

```
SET PGPASSWORD=<<your superuser password>> <enter>
```

```
...\Postgresql\8.0\bin\createdb -h localhost -p 5432 -U postgres -q -E SQL_ASCII -T template1 -O postgres Modus <enter>
```

```
...\Postgresql\8.0\bin\psql -h localhost -p 5432 -U postgres -q -d Modus -f Monitor_pg.sql <enter>
```

```
SET PGPASSWORD= <enter>
```

5. In the GWGuardian Administration Console, choose **System > Properties > Monitoring Database**.
6. Select **Use Native Postgres Database** and enter the appropriate information.
7. Click **Apply**.
8. Stop and start all Modus services in the GWGuardian Administration Console.

4.3 MS Access

1. This database is created automatically during the GWGuardian installation
2. In the GWGuardian Administration Console, choose **System > Properties > Monitoring Database**.
3. Select **Use Legacy Access Database**.
4. Click **Apply**.
5. Stop and start all Modus services in the GWGuardian Administration Console.

5 Creating and Configuring the Sieve Database

The following provides information for creating the Sieve database and configuring it in GWGuardian:

5.1 SQL Server / SQL Express

1. Create a new database.
2. Run the following script, located in C:\Program Files\Messaging Architects\GWGuardian\DBStructures\SQL Server\SieveStore.
 - mssql_sievestore_new.sql
3. Create the ODBC connection
4. In the GWGuardian Administration Console, choose **Spam > Properties > Sieve Database**.
5. Enter your ODBC information and click **Apply**
6. Stop and start all Modus services in the GWGuardian Administration Console.

5.2 PostgreSQL

1. Create a new database.
2. Run the following script, located in C:\Program Files\Messaging Architects\GWGuardian\DBStructures\PostgreSQL\SieveStore.
 - pg_sievestore_new.sql
3. Create the ODBC connection.
4. In the GWGuardian Administration Console, choose **Spam > Properties > Sieve Database**.
5. Enter your ODBC information and click **Apply**
6. Stop and start all Modus services in the GWGuardian Administration Console.

5.3 MS Access

1. The database is created automatically when GWGuardian is installed
2. In the GWGuardian Administration Console, choose **Spam > Properties > Sieve Database**.
3. Select **Use Native MDB Database**.
4. Click **Apply**.
5. Stop and start all Modus services in the GWGuardian Administration Console.

6 Creating and Configuring the Audit Database

The following provides information for creating the Message Audit database and configuring it in GWGuardian:

6.1 SQL Server / SQL Express

1. Create a new database
2. Run the following scripts, in this order, located in C:\Program Files\Messaging Architects\GWGuardian\DBStructures\SQL Server\Audit.
 - Audit_tables.sql
 - Audit_procs.sql
3. In the GWGuardian Administration Console, choose **System > Properties > Message Audit Database**.
4. Select **Microsoft SQL Server**.
 - Enter the required information.
5. Click **Apply**.
6. In the GWGuardian Administration Console, choose **Logs > Properties > Message Audit Database**.
 - Enable **Audit Logging** and **System-Wide Logging**.
7. Click **Apply**.
8. Stop and start all Modus services in the GWGuardian Administration Console.
9. From a Command Prompt, type **iisreset** <enter> to restart the IIS service.

6.2 PostgreSQL

1. Create a new database
2. Run the following scripts, in this order, located in C:\Program Files\Messaging Architects\GWGuardian\DBStructures\PostgreSQL\Audit.
 - Audit_tables.sql
 - Audit_procs.sql
3. In the GWGuardian Administration Console, choose **System > Properties > Message Audit Database**.
4. Select **PostgreSQL**.
 - Enter the required information.
5. Click **Apply**.
6. In the GWGuardian Administration Console, choose **Logs > Properties > Message Audit Database**.
 - Enable **Audit Logging** and **System-Wide Logging**.
7. Click **Apply**.
8. Stop and start all Modus services in the GWGuardian Administration Console.
9. From a Command Prompt, type **iisreset** <enter> to restart the IIS service.

7 Upgrading the Databases

The upgrade scripts can be run for whichever version of GWGuardian is being upgraded. Messaging Architects provides scripts that are run automatically with every upgrade. These scripts are run manually because they take very long to complete. It is, therefore, recommended that you run these scripts during off peak times and for every upgrade.

7.1 Quarantine

Before running the manual script, stop the MODUSADM service.

SQL Server

Location: C:\Program Files\Messaging Architects\GWGuardian\DBStructures\SQL Server\Quarantine

- Run `mssql_quarantine_manual_upgrade.sql`

PostgreSQL

Location: C:\Program Files\Messaging Architects\GWGuardian\DBStructures\PostgreSQL\Quarantine

- Run `pg_quarantine_manual_upgrade.sql`

7.2 Audit

SQL Server

Location: C:\Program Files\Messaging Architects\GWGuardian\DBStructures\SQL Server\Audit

- Stop all Modus services
- Run `audit_upgrade.sql`
- Run `audit_procs.sql`
- Start all Modus services
- Run `audit_upgrade_data.sql`

8 Security and Maintenance

The following provides basic information about the security and maintenance of databases. These measures are optional but Messaging Architects recommends that you maintain database standards.

8.1 SQL Server Security Enhancements

To improve security, the database user should have **minimum** privileges:

- No server administrator role.

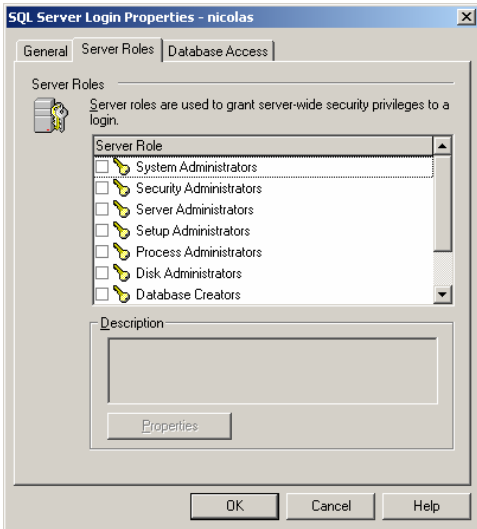


Figure 1: SQL Login Server Roles

- Access granted to the new database **only**.

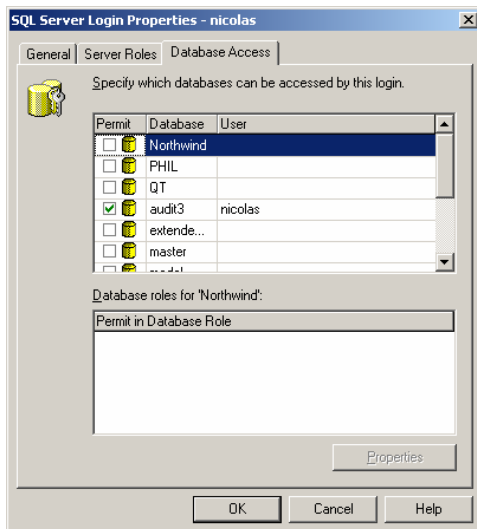


Figure 2: SQL Login Database Access

- Only db_datareader and db_datawriter permissions for the new database

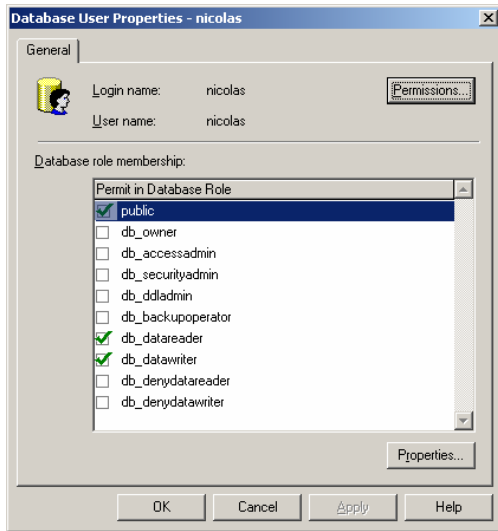


Figure 3: SQL Database User Role

- No permission to create, drop or alter tables.
- Permission to execute all DB stored procedures.

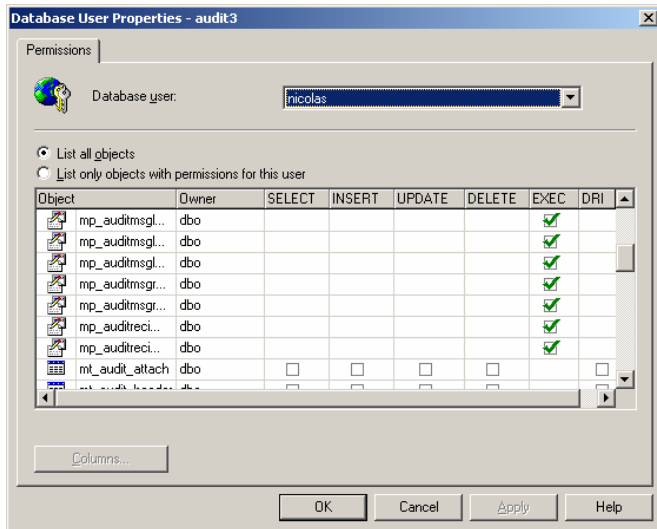


Figure 4: SQL Database User Permissions

Maintaining a SQL Server Database

The following provides basic information for maintaining SQL databases.

SQL Server allows for the recovery of databases. The recovery models available for SQL Server databases each establish the type of backup and restore task that is permitted as well as the acceptable limit for data loss. There are three recovery models: Full, Simple and Bulk logged.

Full Recovery Model

- Uses database and transaction log backups which provide the option for full or differential restores.
- Allows you to precisely recover the database to a specific time which greatly reduces the loss of data.
- All operations are logged and can easily be recovered.
- Provides the most flexibility and is the most commonly used option.
- Use this method if:
 - your data is critical and you cannot afford to lose any of it.
 - you require recovery to a point in time.
 - you use replication and require the option to synchronize all of your databases to a particular point in time.
 - your standard transaction incorporates bulk logged activities.

Simple Recovery Model

- Only restores data from the most recent full or differential backup.
- There are no transaction log backups.
 - The contents for the log are truncated whenever a checkpoint is issued for the database.
- Use this method if:
 - your data is not critical and does not change often.
 - data is collected from various sources and can be easily reproduced.
 - disk space is limited for logging transaction (the issue with disk space should be addressed with a possible hardware upgrade).

Bulk-Logged Recovery Model

- Provides protection without affecting system performance.
- Some operations are minimally logged and are not fully recoverable.
- Damaged file recovery requires you to manually resolve the issue for operations that are not completely logged.
- Only allows for database restores to the end of a transaction log backup (where it contains bulk changes).
- Use this method if:
 - data is critical but so is system performance.
 - bulk operations are performed during off-hours and do not impede regular processing.
 - recovery to a point in time is required.

Changing Recovery Models

1. In SQL Server, open **Enterprise Manager**
2. Right-click on the database name and select **Properties**
3. Select the **Options** tab.
4. Under **Recovery Model**, use the drop-down menu to select the model.
5. Click **OK**.

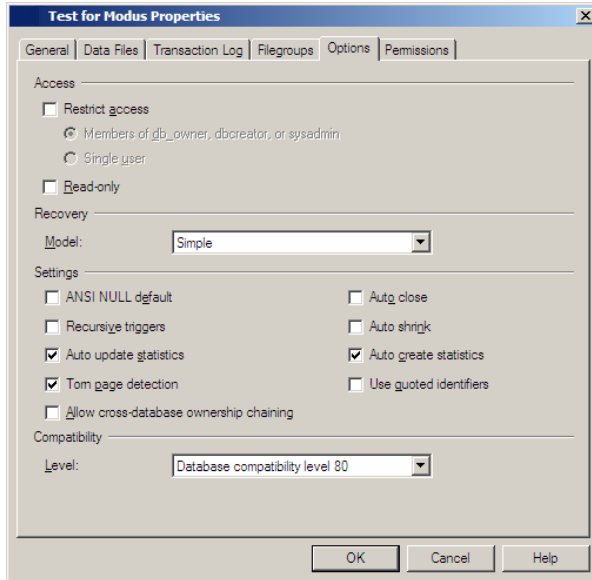


Figure 5: Recovery Model Settings

Optimizing PostgreSQL for GWGuardian

Checkpoint Segments

GWGuardian data written to the PostgreSQL database is cached in a log file until enough data is accumulated to warrant a disk-write. Ideally, the log size should be larger than the default setting as this allows GWGuardian to perform faster (if PostgreSQL writes to the disk too often, GWGuardian performance slows).

The **checkpoint_segments** configuration setting determines when PostgreSQL can write data to the disk. PostgreSQL has two settings for the checkpoints. The first, **checkpoint_segments**, is based on the amount of data modified. The second, **checkpoint_timeout**, is a timeout in seconds. If there are no checkpoints performed in the past X seconds, PostgreSQL will perform one.

A segment represents a certain amount of data modified. During the PostgreSQL installation, this value defaults to **3** so that when 3 segments of data are accumulated, PostgreSQL writes to the disk.

Messaging Architects recommends changing the **checkpoint_segments** value to **10** for optimum GWGuardian performance. Keep the default value for **checkpoint_timeout**.

To change the **checkpoint_segments** setting:

1. Choose **PostgreSQL > Configuration files** and select **edit postgresql.conf**.
2. Find the line **checkpoint_segments** and change the value to 10.

3. Remove the **#** at the start of the line to **uncomment** the line
4. Save the file.
5. Stop and start the POSTGRESQL service on the server.
6. Stop and start **all** Modus services in the GWGuardian Administration Console.

Max_fsm_pages

Whenever a database is modified, PostgreSQL allocates new space in pages of 8K. The old page is still there and cannot be re-used until there is a **vacuum** command called.

Once every hour, GWGuardian issues the command to be able to re-use the old pages. Re-using an old page will not increase the size of the file on the disk. The setting **max_fsm_pages** indicates the maximum number of pages to mark as re-usable when the vacuum command is used.

Problems occur when there are more than 20,000 pages that are modified per hour.

Setting the configuration to 1,000,000 pages should re-use all of the old pages even if the vacuum operation takes many hours to complete. The downside is that it uses more memory. PostgreSQL will use about 5MG of additional memory when the setting is changed from 20,000 pages.

1. From the **Start** menu, go to **PostgreSQL > Configuration Files > Edit postgresql.conf**
2. Locate the line **#max_fsm_pages = 20000** and modify it to **max_fsm_pages = 1000000**
 - Remove the **#** so that the line is no longer a comment
3. Stop and start the POSTGRESQL service on the server.
4. Stop and start **all** Modus services in the GWGuardian Administration Console.