

Server Health and Performance Monitoring

One of the benefits of using CardioLog Analytics on-premise is a complete ownership of your own data, on your own servers, which allows you to control both privacy and security. This also means that server maintenance and database health is the customer's responsibility, including software and hardware upgrades. A well maintained server and database helps increase performance and supports smooth operation.

Contents

- [Optimal Performance Guidelines](#)
- [Monitoring the Operating System and Database](#)
- [Backup Policy](#)
 - [Database Backup](#)
 - [Application Backup](#)
- [Database Maintenance](#)
 - [Index Maintenance](#)
 - [Table Maintenance](#)
 - [Shrink Database/Data Files](#)
 - [Query Optimizer Statistics](#)

Optimal Performance Guidelines

To ensure optimal operation, we recommend following these guidelines:

1. Install the CardioLog application and database on dedicated servers that meet or exceed the recommended [hardware and software requirements](#).
2. Upgrade the product to the [latest version](#). To find out what your product version is, go to Administration > Product License.
3. [Monitor the operating system and database](#).
4. [Schedule all CardioLog service jobs](#) to execute during off-peak hours, while no other system jobs are running (eg., System backup, SharePoint search crawl, etc.). By default all CardioLog service jobs are scheduled to run every day at midnight, except Usage Data Processing which runs every hour.
5. Configure [email alerts for scheduled services and event collection thresholds](#) that will be sent to the system administrator.
6. Execute [this script](#) against the CardioLog database to learn about your current database size, the amount of usage data that you have, index fragmentation rates and more.
7. [Reorganize and rebuild database indexes](#) with high fragmentation rates.
8. [Archive history data](#) to free up disk space on the database.
9. Delete unused reports and [optimize existing reports](#). You can execute [this script](#) against the CardioLog database to retrieve a list of all reports in your system that includes the following details: report path, report name, schedule type, owner, number of widgets, website item filter and date range filter.

Monitoring the Operating System and Database

1. Monitor the IIS service and the CardioLog services ([CardioLog Scheduling Service](#) and [CardioLog Diagnostics Service](#)).
2. Monitor the server CPU performance and memory usage.
3. Monitor the logs directory size on the CardioLog server. Log file growth is restricted to 8MB each.
4. Monitor the database size a rough growth estimation is typically 5GB of storage per 1,000,000 page views. See the [database sizing form](#) ula in order to allocate enough storage.
5. Monitor the SQL database services.

Backup Policy

Database Backup

1. The CardioLog database is updated on a daily basis. The database recovery model should be set to "simple" due to its update behavior. We highly recommend not changing the CardioLog recovery model.
2. A full database backup should be taken before the nightly processes run (verify the [nightly processing schedule](#)). In order to backup the CardioLog database, schedule the following command:

```
Backup database cardiolog to disk = '<Path>\cardiolog.bak'
```

Note: It is the customer's responsibility to keep the CardioLog database backup in a safe location. For more information about the database backup procedure please consult your local database administrator (DBA) or [Intlock support](#).

Application Backup

Backup the CardioLog installation directory after every software update.

Database Maintenance

Index Maintenance

1. Creating and dropping Indexes:
Creating or dropping indexes from the CardioLog database is prohibited. Please consult [Intlock support](#) if a DROP or CREATE operation is needed.
2. Rebuilding Indexes:
Rebuild can be performed manually (consult your local DBA about index rebuild), with the following considerations taken into account:
 - Index rebuild may affect reporting performance.
 - Index rebuild should be done after the automated nightly processes have run.

To ensure optimal operation, we recommend:

- Rebuilding indexes that have an average fragmentation of over 30% (with ONLINE = ON), and reorganize indexes when they are between 5% to 30%.
- Reorganizing indexes once a day during off-peak hours.
- Rebuilding indexes once a week during off-peak hours. There is no need to reorganize indexes if they're rebuilt the same day.
- Rebuilding indexes only for the current month if using partitioned tables by month (for table tab_event_log).

Table Maintenance

CardioLog does not support manual table reorganization. In cases of major table fragmentation, or when a database table needs to be regionalized, please consult [Intlock support](#).

Shrink Database/Data Files

CardioLog shrinks the database log file automatically. To disable this option, run the following command against the CardioLog database:

```
update tab_global_settings  
set Value='false'  
where Category='OMAgent'and Name='ShrinkLogFile'
```

There is no need to consult Intlock support before shrinking the database data files.

Query Optimizer Statistics

We recommend using auto-statistics update. Disabling or changing the statistics gathering process can affect the application performance, therefore it is highly discouraged. In addition to the above recommendations, the following T/SQL procedures can be used to monitor and maintain the CardioLog database:

DBCC SHOWCONTIG	Displays data fragmentation information, and indexes of the specified table.
DBCC DBREINDEX	Rebuilds one or more indexes for a table in the specified database.
DBCC INDEXDEFRAG	Defragments clustered and secondary indexes within the specified table or view. DBCC INDEXDEFRAG is an online operation, so it does not hold long-term locks that can block running queries or updates. DBCC INDEXDEFRAG can be considerably faster than running DBCC DBREINDEX because a relatively unfragmented index can be defragmented much faster than a new index can be built.

DBCC SHOW_STATISTICS	Displays the current distribution statistics for the specified target on the specified table.
UPDATE STATISTICS	Updates information about the distribution of key values for one or more statistics groups in the specified table or indexed view.
sp_updatestats	Displays or changes the automatic UPDATE STATISTICS setting for a specific index and statistics, or for all indexes and statistics for a given table or indexed view in the current database.
sp_updatestats	Runs UPDATE STATISTICS against all user-defined tables in the current database.