# GoA Reporting Database - GADATA\_RPT

## DatabaseSettings - Database creation

The three ‘DatabaseSettings’ files define the creation, configuration, and modification of the GADATA\_RPT database. These scripts together establish a robust foundation for the GADATA\_RPT database, tailored for specific operational needs. Below is a detail of what each file does:

1. **Database Creation:** Creates the database with defined data files, log files, and filegroups.

2. **Filegroup Modification:** Enables auto-growth for specific filegroups.

3. **Configuration Settings:** Applies various database settings for performance optimization, recovery options, and query monitoring.

### 01\_Database\_Create.sql

This script creates the GADATA\_RPT database with specific settings and file configurations:

* The database is created on the ‘master’ database with the name GADATA\_RPT.
* It uses a primary data file (GADATA\_RPT) with an initial size of 5GB, unlimited maximum size, and a growth increment of 10MB.
* A secondary filegroup (GADATA\_RPT\_FullTextIndex) is defined for full-text indexing, with an initial size of 10GB, a maximum size of 20GB, and a growth increment of 512MB.
* A log file (GADATA\_RPT\_log) is configured with an initial size of ~5.7MB, a maximum size of 20GB, and a growth increment of 10MB.

#### Additional Settings:

* CATALOG\_COLLATION is set to DATABASEDEFAULT.
* LEDGER feature is disabled.

#### SQL SCRIPT:

[01\_Database\_Create.sql](https://dev.azure.com/bki-tio/GoAReporting/_git/GoAReporting?version=GBmaster&path=/GoAReporting/iLab/SQL/Solutions/GoA_Reporting_v1.0/DatabaseSettings/01_Database_Create.sql)

### 02\_Filegroups\_Modify.sql

The script adjusts the configuration of the filegroups in the GADATA\_RPT database:

* The GADATA\_RPT\_FullTextIndex filegroup is modified to enable auto-growth for all files within this filegroup.

[02\_Filegroups\_Modify.sql](https://dev.azure.com/bki-tio/GoAReporting/_git/GoAReporting?version=GBmaster&path=/GoAReporting/iLab/SQL/Solutions/GoA_Reporting_v1.0/DatabaseSettings/02_Filegroups_Modify.sql)

### 03\_Configuration\_Settings.sql

This script configures various settings for the GADATA\_RPT database to optimize its behavior and performance:

* If full-text services are installed, a full-text catalog (GADATA\_RPT\_FTCatalog) is created with accent sensitivity enabled and set as default.

#### Database Options:

* ANSI-related options (e.g., ANSI\_NULL\_DEFAULT, ANSI\_NULLS, etc.) are turned off.
* Features like AUTO\_CLOSE, AUTO\_SHRINK, and AUTO\_UPDATE\_STATISTICS\_ASYNC are disabled.
* Recovery model is set to \*\*SIMPLE\*\*, which minimizes log space usage.
* Database integrity checks are ensured by enabling PAGEVERIFY CHECKSUM.

#### Query Store:

* The Query Store feature is enabled for performance monitoring and tuning. It is configured with settings such as:
* Operation mode: Read/Write
* Cleanup policy: Stale queries older than 30 days are cleaned up.
* Maximum storage size: 1GB
* Query capture mode: Automatic
* Wait statistics capture mode: Enabled

#### Miscellaneous Settings:

* Features like snapshot isolation, delayed durability, and accelerated database recovery are disabled.
* Multi-user access mode is enabled.

#### SQL SCRIPT:

[03\_Configuration\_Settings.sql](https://dev.azure.com/bki-tio/GoAReporting/_git/GoAReporting?version=GBmaster&path=/GoAReporting/iLab/SQL/Solutions/GoA_Reporting_v1.0/DatabaseSettings/03_Configuration_Settings.sql)

## FUNCTIONS

### fn\_ScriptPerformance.SQL

This SQL function ‘dbo.fnScriptPerformance’, which calculates script performance metrics, including runtime in milliseconds, minutes, seconds, and breakdowns of execution time.

#### Inputs

* StartTime: A string of type ‘NVARCHAR(20)’representing the start time of the script execution.

#### Outputs

Returns an integer (‘INT’) with a formatted message containing:

* Start time
* Total runtime in milliseconds.
* Runtime breakdown (minutes, seconds, milliseconds).
* End time.

#### SQL SCRIPT:

[fn\_ScriptPerformance.sql](https://dev.azure.com/bki-tio/GoAReporting/_git/GoAReporting?version=GBmaster&path=/GoAReporting/iLab/SQL/Solutions/GoA_Reporting_v1.0/Functions/fn_ScriptPerformance.sql)

#### Key Features

1. Runtime Calculation:
   1. Uses DATEDIFF(millisecond) to compute the elapsed time between StartTime and the current system date/time (GETDATE()).
2. Breakdown of Execution Time:
   1. Converts total elapsed milliseconds into hours, minutes, seconds, and remaining milliseconds for detailed analysis.
3. Formatted Output:
   1. Constructs a human-readable message summarizing:
   2. Start time.
   3. Total runtime in milliseconds.
   4. Breakdown of runtime components.
   5. End time.

#### Usage Example

To calculate script performance:

DECLARE @StartTime NVARCHAR(20) = FORMAT(GETDATE(), 'yyyy-MM-dd HH:mm:ss');

SELECT dbo.fnScriptPerformance(@StartTime);

This will return a formatted message detailing the script's performance metrics.

#### \*\*Notes

* Ensure ‘StartTime’is provided in a valid format (‘yyyy-MM-dd HH:mm:ss’) to avoid errors during execution.
* The function assumes success by default (‘ReturnCode = 1’) but does not explicitly handle exceptions or invalid inputs.

## Tables - addin

### addin\_DateDimension.sql

#### Description:

This script creates a table named ‘addin\_DateDimension’ in the ‘GADATA\_RPT’ database. The table is designed to store date-related information, including key details about specific date parts/combinations.

#### Purpose:

Provides dates in formats used by SProcs when updating daily metadata. Itegral part of a ‘rolling’ data model.

#### SQL Script:

[addin\_DateDimension.sql](https://dev.azure.com/bki-tio/GoAReporting/_git/GoAReporting?version=GBmaster&path=/GoAReporting/iLab/SQL/Solutions/GoA_Reporting_v1.0/Tables/addin_DateDimension.sql)

### \*\*Notes

* Run ‘sp\_PopulateDateDimension.sql’ to initiate this table.
  + Use Parameters to control DatDimention Tbl date range:
    - @StartDate DATE = '2020-01-01', <- *default*
    - @outYears INT = 30 <- *default*

### addin\_LogTblTriggerMaintHist.sql

#### Description:

This script creates a table named ‘addin\_LogTblTriggerMaintHist’ in the ‘GADATA\_RPT’ database. It is intended to log maintenance history for table trigger activity.

#### Purpose:

The table tracks actions performed on database tables, including the number of rows affected, action timestamps, and any errors encountered during trigger maintenance.

#### SQL Script:

[addin\_LogTblTriggerMaintHist.sql](https://dev.azure.com/bki-tio/GoAReporting/_git/GoAReporting?version=GBmaster&path=/GoAReporting/iLab/SQL/Solutions/GoA_Reporting_v1.0/Tables/addin_LogTblTriggerMaintHist.sql)

### addin\_Log\_SPRunHistory.sql

#### Description:

This script creates a table named ‘addin\_Log\_SPRunHistory’ in the ‘GADATA\_RPT’ database. It is intended to log maintenance history for stored procedure activity.

#### Purpose:

The table tracks actions performed on database tables, including the number of rows affected, action timestamps, and any errors encountered during stored procedure maintenance.

#### SQL Script:

[addin\_Log\_SPRunHistory.sql](https://dev.azure.com/bki-tio/GoAReporting/_git/GoAReporting?version=GBmaster&path=/GoAReporting/iLab/SQL/Solutions/GoA_Reporting_v1.0/Tables/addin_Log_SPRunHistory.sql)

### addin\_SvcLogTableStructures.sql

#### Description:

This script creates a table named ‘addin\_SvcLogTableStructures’ in the ‘GADATA\_RPT’ database. It is intended to store metadata for each table to be replicated.

#### Purpose:

The table contains metadata required to maintain replication for a GoA table. A row representing a desired TableDistroLvl can be duplicated, updating all table specific information for new tables to be replicated in GoA Reporting.

#### SQL Script:

[addin\_SvcLogTableStructures.sql](https://dev.azure.com/bki-tio/GoAReporting/_git/GoAReporting?version=GBmaster&path=/GoAReporting/iLab/SQL/Solutions/GoA_Reporting_v1.0/Tables/addin_SvcLogTableStructures.sql)

## Stored Procedures

### sp\_PopulateDateDimension.sql

#### Description:

This sproc populates a table named ‘addin\_DateDimension’ in the ‘GADATA\_RPT’ database; with a range of dates and their associated attributes. It is designed to support reporting and metadata requirements.

#### parameters:

* Use Parameters to control DatDimention Tbl date range:
  + @StartDate DATE = '2020-01-01', <- *default*
  + @outYears INT = 30 <- *default*

#### SQL Script:

[sp\_PopulateDateDimension.sql](https://dev.azure.com/bki-tio/GoAReporting/_git/GoAReporting?version=GBmaster&path=/GoAReporting/iLab/SQL/Solutions/GoA_Reporting_v1.0/StoredProcedures/sp_PopulateDateDimension.sql)

### sp\_fnLogSPRunHistory.sql

#### Description:

This procedure used by classes to log procedure activity to a table named ‘addin\_Log\_SPRunHistory’ in the ‘GADATA\_RPT’ database. Table records sproc execution results.

#### parameters:

* Use Parameters to log via sp\_fnLogSPRunHistory.sql
  + @Run\_id INT,
  + @TableAffected NVARCHAR(255) = 'NameMe', -- Default value
  + @RowsAffected INT = 0,
  + @Action NVARCHAR(255) = 'CommentMe', -- Default value
  + @ErrorCode INT = 0,
  + @ErrorMessage NVARCHAR(255) = NULL

#### SQL Script:

[sp\_fnLogSPRunHistory.sql](https://dev.azure.com/bki-tio/GoAReporting/_git/GoAReporting?version=GBmaster&path=/GoAReporting/iLab/SQL/Solutions/GoA_Reporting_v1.0/StoredProcedures/sp_fnLogSPRunHistory.sql)

### sp\_SvcLog\_0.MetaData.Master.sql

#### Description:

The stored procedure sp\_SvcLog\_0.MetaData.Master updates metadata tables, daily, in the GADATA\_RPT database. It facilitates the creation and maintenance of hard temporary tables (temp\_SvcLog\_1\_Create, temp\_SvcLog\_2\_Load, temp\_SvcLog\_3\_Purge).

#### parameters:

* Use Parameters to log via sp\_fnLogSPRunHistory.sql
  + @Run\_id INT = 0 OUTPUT

#### SQL Script:

[sp\_SvcLog\_0.MetaData.Master.sql](https://dev.azure.com/bki-tio/GoAReporting/_git/GoAReporting?version=GBmaster&path=/GoAReporting/iLab/SQL/Solutions/GoA_Reporting_v1.0/StoredProcedures/sp_SvcLog_0.MetaData.Master.sql)

### 1.Create Class Stored Procedures