A picture containing sky, water, person, outdoor

Description automatically generated



Version: 1.1

Date: 28 June 2024

Supplementary Implementation Guide

Global Risk Institute

**TABLE OF CONTENTS**

[DOCUMENT DETAILS 2](#_Toc170498184)

[1 INTRODUCTION 3](#_Toc170498185)

[1.1 Purpose 3](#_Toc170498186)

[1.2 Audience 3](#_Toc170498187)

[2 Artifacts 3](#_Toc170498188)

[2.1 Artifacts Provided 3](#_Toc170498189)

[3 IMPLEMENTATION: Migration GUIDE 4](#_Toc170498190)

[3.1 Migration Implementation 4](#_Toc170498191)

[3.2 Migration Implementation Status 4](#_Toc170498192)

[4 IMPLEMENTATION: USE CASES guide 5](#_Toc170498193)

[4.1 Use Case Implementation 5](#_Toc170498194)

[4.2 Use Case Implementation Status 6](#_Toc170498195)

# DOCUMENT DETAILS

**Version Management**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Nature of amendment |
| 1.1 | Jun 28, 2024 | Naftal Kerecha, Don Munasinghe | Supplementary Implementation Guide |

**Reviewer**

|  |  |  |
| --- | --- | --- |
| Name | Version | Date |
| Naftal, Kerecha |  |  |
| Don, Munasinghe |  |  |

# INTRODUCTION

## Purpose

This document contains details on the sections of the solution that BDO began implementing alongside the SMEs. It also provides guidance on how some of the remaining items should be implemented. It is meant to be a supplementary document to the published **GRI Database Migration Technical Design Document** and **GRI Database Migration Software Requirements Specification** documentswhich contain the full scope of the recommended implementation.

## Audience

The target audience is the implementation team and the SMEs that will be responsible for performing the rest of the migration.

# Artifacts

## Artifacts Provided

A zip file attachment that contains the files for the sections that have been implemented so far will be provided alongside this document. The following section details all the files provided and their location.

|  |  |
| --- | --- |
| File Name | Location |
| GRI Model DDL | DDL\GRI Model DDL.sql |
| GRI Migration Script | Migration\GRI Migration Script.sql |
| GRI Drop Tables Commands | Stored Procedures\Drop\GRI Drop Tables Commands.sql |
| GRI Log Trigger SP | Stored Procedures\GRI Log Trigger SP.sql |
| CreateAffiliation | Stored Procedures\Crud\Affiliation\CreateAffiliation.sql |
| CreateAttendance | Stored Procedures\Crud\Attendance\CreateAttendance.sql |
| CreateCompany | Stored Procedures\Crud\Company\CreateCompany.sql |
| CreateEvent | Stored Procedures\Crud\Events\CreateEvent.sql |
| CreateReporting | Stored Procedures\Crud\Reporting\CreateReporting.sql |
| CreateUser | Stored Procedures\Crud\Users\CreateUser.sql |
| UpdateUser | Stored Procedures\Crud\Users\UpdateUser.sql |
| Create Attendance Template | Stored Procedures\Crud\Attendance\CreateAttendanceTemplate.xlsx |

# IMPLEMENTATION: Migration GUIDE

## Migration Implementation

This section mainly references the migration part of the project focused on transitioning the data to utilizing the new schema. For the migration to be completed the following items are required:

* **Creation of the DDL script –** The DDL file is a SQL script that contains the definitions for the new schema model. This file contains all the table definitions, dependencies, constraints as well as relationships between tables. In the set of artifacts provided the DDL can be found at this location **DDL\GRI Model DDL.sql.**
* **Implementing the DDL script** – The default database schema name in SQL Server is **[dbo].** Running the DDL script should be done in the same database that has the current schema. The DDL script will create a new schema called **[mod].** The new set of tables that data will be migrated to as well as any sequences and stored procedures will then be created under this **[mod]** schema.
* **Creating the migration script** – All the SQL commands responsible for migrating the actual data are contained in the migration file. The file will migrate data from the static reference tables and then proceed with the migration of the remaining tables. In the set of artifacts provided the DDL can be found at this location **Migration\GRI Migration Script.sql.**
* **Validating the migration records and identifying data defects** – Once the migration script has been run. We need to go through a validation process to verify that all the required data has been migrated. This can be done through checks such as row counts. The migration script may fail in certain cases when in-built validations such as constraint checks fail which would imply the existence of defects within the source data.
* **Troubleshooting the defects and data quality issues** – Any defects identified will need to be resolved for the migration to be considered successful. So far, we have identified the following data quality issues:
  + Reference columns in certain tables do not have matching values in the reference tables. This requires that updates be made to the affected rows so that the mapping does not result in missing values during the migration. Affected tables are the following: e\_subtype, e\_theme, and e\_venue columns used in the Events Table and a\_group column used in the Affiliation Table.
  + The current user table has comma-separated tags for users who have multiple tags. The resolution for this is to create a separate table that will map all the tags assigned to a user.
  + Some company names in the a\_company column of the Attendance table do not have a corresponding entry in the Company table that we can use for mapping. This issue impacts seventy-eight records.
  + The Affiliation table attempts to reference users who are missing in the users’ table. This issue impacts two user records.

## Migration Implementation Status

This section below highlights the status of the various components required for migration to be complete:

|  |  |  |
| --- | --- | --- |
| Task | Status | Additional Details |
| DDL Script Creation | Complete | The DDL script that we created has been provided in the artifact zip file. |
| Initial DDL Script Run | Complete | The initial run was performed to verify that the script provided was functional. |
| Initial Migration script creation | Complete | An initial version of the migration script has been provided. |
| Migration script initial test run and output validation | Complete | The migration script provided was used in a test run to verify that it can execute. The migration failures with some of the tables were caused by the list of defects noted in the section above. |
| Resolve data defects and data quality issues | In Progress | The data issues identified above are in the process of being resolved by the SMEs. |
| Migration script update to track the previous primary keys | Pending | Due to pre-existing foreign key dependencies in the tables, we need to ensure that we temporarily keep track of the old primary keys as we migrate the data to the new set of tables. This will ensure that we can perform necessary joins to the old schema tables when loading data into the new tables. |
| Updated migration script test run and validation | Pending | Once data issues have been addressed and any required adjustments to the migration have been applied, we repeat the above processes until we can successfully validate the migrated data. |

# IMPLEMENTATION: USE CASES guide

## Use Case Implementation

The TDD and SRS listed several use cases that require independent processes created to support GRI operations. For each use case that needs to be developed, the following items will need to be completed:

* **Creation of the supporting Excel templates** – The use cases as defined in the TDD document will utilize Excel templates that are manually filled by the various teams. The implementation team will have to review the various use cases and decide if it is possible to utilize one Excel template for all use cases or if multiple Excel templates be a more reasonable approach.
* **Creation of the staging tables matching the Excel templates to be used** – For the Excel templates we will have to create matching staging tables to load template data. The data will be migrated from those staging tables to the relevant tables when the stored procedures for the use cases are executed. The tables will use the **[stg]** tag for their schema name. An example entry for a staging table can be found in the DDL script.
* **Creation of CRUD Stored procedures** – To support the functionality of the Use Cases, stored procedures that handle create, update, and delete operations for each table for multiple records at once will be required. These CRUD-stored procedures can then be referenced and executed if any use case requires it.
* **Execution of stored procedures for testing** – We recommend building out Use Case 1 as the base implementation. Once Use Case 1 is fully functional then the implementation team can proceed with working on the more complex use cases.

## Use Case Implementation Status

The section below highlights the status of the various components required for the Use Case scenarios to be functional:

|  |  |  |
| --- | --- | --- |
| Task | Status | Additional Details |
| Example Excel Support Template – Based on Use Case 1 | Complete | An example of what the Excel templates to be utilized for use cases has been included in the zipped artifacts. |
| Example of staging schema table - Based on Use Case 1 | Complete | An example of the staging table matching the Excel template has been included within the DDL and will be created when the DDL script has been run. |
| Supporting CRUD stored procedures | In Progress | The Create and Update functionalities for various tables have been implemented. This format can be followed by the implementation team for the remaining tables. |
| SSIS Data Profiling & Validation Set Up | Pending | The steps to be followed to perform this kind of setup were included in the published TDD file. |
| Execute and test Use Case 1 | Pending | Execute Use Case 1 Stored Procedure to verify that it is capable of successfully performing the required create or update actions. This is meant to be an iterative process until the expected functionality is achieved. |
| Implementing remaining use cases utilizing Use Case 1 as Base | Pending | The remaining use cases can be created once Use Case 1 is complete. Some of the use cases will depend heavily on the CRUD functionality and conditional checks to determine if operations to other tables are required. |