# tagline

***RMA Migration Project***

***Database Migration Installation***

***& Configuration***

***February 2021***

## Purpose of the Document

This document provides a description of the zipped file, along with instructions of installing and using the Data Transfer Utility application for database migration.

This document is intended for anyone involved with the delivery of the RMA Migration Project.

## Revisions

| **Revision** | **Date** | **Description** |
| --- | --- | --- |
| 1.00 | May 30, 2017 | Initial Document |
| 1.10 | June 16, 2017 | Revised instructions in the “Installation and First Time Use” section. |
| 1.20 | July 9, 2017 | Revised the first time use in the “Installation and First Time Use” section. |
| 1.30 | July 15, 2017 | Added connection2 and updated use. |
| 1.40 | August 8, 2018 | Added the “Scripts Location” field to execute the script to create the views. |

Table of Contents

[i](#_Toc8300233)

[Purpose of the Document ii](#_Toc8300234)

[Revisions ii](#_Toc8300235)

[Introduction 1](#_Toc8300236)

[SQL Script Files 2](#_Toc8300237)

[Installation and First Time Use 4](#_Toc8300238)

[Reusing the Data Transfer Utility 7](#_Toc8300239)

## Introduction

The Data Transfer Utility consists of 1 zipped file containing the application’s executable, a “Scripts” folder containing various SQL scripts, and a “Data” folder containing the data to be used as the baseline during the migration process.

This application will allow RMA to perform a number of database migrations during the course of migration project in order to periodically clean up the data, if needed. This will result in no or minimal data fixes when RMA goes live with the migrated application.

## SQL Script Files

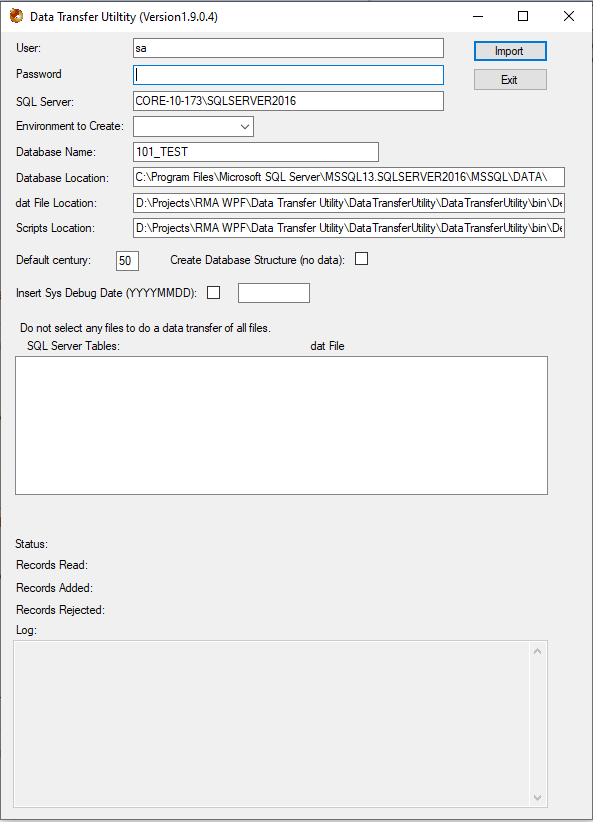
The SQL scripts included in this package are used to create the RMA101C database. Below is a brief description of each SQL file.

The SQL scripts are designed to work with the application, as there are placeholders in them that are replaced with values entered in from the application.

|  |  |
| --- | --- |
| SQL Script File | Description |
| Connection.sql | Connection string used to connect to SQL Server with sa user |
| Connection2.sq1 | Connection string used to connect to SQL Server with windows authentication user |
| CreateDatabase.sql | Create the database specified by the user |
| CreateDirectTables.sql | Create the tables for the DIRECT schema |
| CreateIndexedTables.sql | Create the tables for the INDEX schema |
| CreateLoggingTables.sql | Create the Logging table |
| CreateSchema.sql | Create the DIRECT, INDEXED, SEQUENTIAL, and TEMPORARYDATA schemas |
| CreateSecurityTables.sql | Create the Security tables |
| CreateSequentialTable.sql | Create the tables for the SEQUENTIAL schema |
| CreateTemporaryTables.sql | Create the tables for the TEMPORARYDATA schema |
| CreateView\_DIFF\_AMTS\_SEL.sql | Creates the view DIFF\_AMTS\_SEL which is used in the RMA application |
| CreateView\_DIFF\_SV\_DATE\_SEL.sql | Creates the view DIFF\_SV\_DATE\_SEL which is used in the RMA application |
| CreateView\_EXTF002HDR.sql | Creates the view EXTF002HDR which is used in the RMA application |
| CreateView\_F002\_ORIG\_DTL.sql | Creates the view F002\_ORIG\_DTL which is used in the RMA application |
| CreateView\_OUTSTANDING\_CLAIMS.sql | Creates the view OUTSTANDING\_CLAIMS which is used in the RMA application |
| CreateViews.sql | Used to execute the CreateView sql scripts |
| InsertCoreDebugSysdate.sql | Inserts the system date into the table CoreDebugSysdate in the 101C database |
| InsertSecurityRecords.sql | Insert records into the Renaissance Security tables |
| KillProcesses.sql | Kills any processes to the existing migrated database before reloading the data |
| UpdateF010\_Pat\_Mstr.sql | Update records in the F010\_PAT\_MSTR table |
| UpdateF020Doctor\_Extra.sql | Update records in the F020\_DOCTOR\_MSTR table |

## Installation and First Time Use

The following steps will guide you through installation of the Data Transfer Utility application. It is already assumed the data files have already been downloaded from UNIX and put on the computer where the data transfer is going to occur.

1. Copy the zip file (Data Transfer Utility.zip) to the computer where the data transfer will be executed. Note: SQL Server needs to be on the same computer. Unzip the contents of the file to a folder on the computer.
2. Create a folder under the Drop 1 subfolder. Copy the Unix data files to this new folder. Add the “.dat” extension to any of the data files that do not have the “.dat” extension.
3. Go to the folder where the Data Transfer Utility application was unzipped to. Depending on the environment the Unix data files came from, copy one of the “Import\_schema\_XXX.core” files to the folder where the data files have been copied to.  
     
   Import\_schema\_101C.core for the Unix files from the 101C environment  
   Import\_schema\_MP.core for the Unix files from the MP environment  
   Import\_schema\_SOLO.core for the Unix files from the SOLO environment
4. Double click on “Data Transfer Utility.exe”. The following screen will appear.  
     
   
5. Enter in values in the following fields:

|  |  |
| --- | --- |
| User | The user name to log into SQL Server. If the user is a “Windows Authenticated” user, this field can be left blank. |
| Password | The password of the SQL Server user name used. If the user is a “Windows Authenticated” user, this field can be left blank. |
| SQL Server | Name of the SQL Server instance the data will be loaded into |
| Environment to Create | Select the “Security” environment from the dropdown. |
| Database Name | Name of the database that will be created in SQL Server (i.e. RMA101C)  Note: The Security and Logging databases are defaulted and cannot be change |
| Database Location | Physical location where the SQL Server database files will be stored |
| dat File Location | Physical location where the dat files are located, if different from the default displayed in the field. |
| Scripts Location | Physical location where the SQL scripts are located. |
| Default century | Number used to determine the century to be placed in front of 2 digit years (i.e. if the value is 50, any 2 digit year that is 50 or higher will have a 19 in from of it, and a 20 in front of the year that is 49 and less). The default is 50. |
| Create Database Structure | Click the checkbox to only create the database structure. For the “Security” environment, the table will be created and the data will be inserted. |
| Insert Sys Debug Date | Check the checkbox if a system debug date should be inserted. If it is checked, enter the date in a “YYYYMMDD” format. |
| List of files to Import | Select the file(s) that should be imported into the migrated database, or select none to import all the files.  Note: If you checked the “Create Database Structure” checkbox, the data will not be migrated.  Note: If you are migrating all the data files, the database will be recreated. |

1. Press the “Import” button.
2. Select “Logging” from the Environment to Create dropdown.
3. Press the “Import” button.
4. Select “101C” or “MP”, or “SOLO” from the Environment to Create dropdown. Enter values in the following fields:

|  |  |
| --- | --- |
| Create Database Structure | Click the checkbox to only create the database structure. For the “Security” environment, the table will be created and the data will be inserted. |
| Insert Sys Debug Date | Check the checkbox if a system debug date should be inserted. If it is checked, enter the date in an “YYYYMMDD” format. |
| List of files to Import | Select the file(s) that should be imported into the migrated database, or select none to import all the files.  Note: If you checked the “Create Database Structure” checkbox, the data will not be migrated.  Note: If you are migrating all the data files, the database will be recreated. |

1. Press the “Import” button.  
     
   If the “Create Database Structure” checkbox is checked, a warning message will appear stating only the data structure will be created and no data will be loaded. Click the “Yes” button to create the data structure.  
     
   If the “Create Database Structure” checkbox is not checked, the utility will import the data into the migrated database.

When the “Import” button is pressed, the application saves some of the values entered before starting the data transfer process. This is so the next time the application is used, the last values entered will be populated in the fields.

The first step in the data transfer process, when 101C, MP, or SOLO is selected as the environment to create, is executing the SQL scripts to create the database, schemas, views, as well as the tables, indices, constraints and triggers for each schema. Once the SQL scripts are completed and the database and views are created, the data transfer of the data from the dat files is imported to the migrated database

If there are errors loading records into a table during the data transfer process, a log file is created in the “Log” folder, inserting into it the erroneous records, along with information about the error. A log file will be created for each dat file where errors occurred. At the end of the data transfer, a text file (Output.txt) is created in the folder where the Data Transfer Utility executable is located. This file will contain the values displayed in the log section of the application.

## Reusing the Data Transfer Utility

The Data Transfer Utility can be used periodically during the migration to check for errors in the legacy data and correct it, so when the migrated application goes live, the final database migration will have minimal or no errors in the data.

1. Export the data from the legacy application.
2. Copy the dat files to a folder on the computer where the Data Transfer Utility is installed on. Ensure the “.dat” file extension is on all dat files that will be imported.
3. Copy the import schema file to the folder where the dat files were copied to in step 2.
4. Go to the folder where the Data Transfer Utility application is located. Double click on “Data Transfer Utility.exe”.
5. If Windows authentication is not used to connect to SQL Server, enter in the password of the SQL Server user.
6. Select which environment to migrate from the “Environment to Create” dropdown list. If the import schema for the legacy data selected is not in the folder in the “data File Location” field, a message will be displayed.
7. If necessary, change the database name and the path where the dat files to be used are located.
8. To do a data transfer of one or more tables, but not all tables, click on the table name in the list box displaying the SQL Server tables. The row will be highlighted if selected. If a data transfer all the dat files is to be done, do not select any tables in the list.
9. Press the “Import” button.

There are certain tables that when selected, data for other tables will also have its data refreshed. This is due to the legacy data being split into different tables in the migrated database. Below is the list of tables that will load data into other tables.

Only one of the tables in the various groups below needs to be selected in the application to refresh the data for all the other related tables.

|  |  |  |
| --- | --- | --- |
| F020\_DOCTOR\_MSTR  F020\_DOCTER\_EXTRA | F020\_CLAIMS\_MSTR\_HDR  F020\_CLAIMS\_MSTR\_DTL  F020\_CLAIMS\_MSTR\_DTL\_DESC | CONSTANTS\_MSTR\_REC\_1  CONSTANTS\_MSTR\_REC\_2  CONSTANTS\_MSTR\_REC\_3  CONSTANTS\_MSTR\_REC\_4  CONSTANTS\_MSTR\_REC\_5  CONSTANTS\_MSTR\_REC\_6  CONSTANTS\_MSTR\_REC\_7 |