# Overview

Conversion efforts are moving into large data cleanup as the application evolves. Work should be performed during a designated outage window to prevent records being updated or inserted during the conversion process. Requires a local install of [dbatools.](https://dbatools.io/" \t "_blank)

This method relies on building out a shell database to serve as the deploy target for the source database schema. Once created, the data is ported over, using special conditions on the blob table, and extra steps for the remaining tables.

**Note:** Always using a dacpac of the source database to avoid unintended consequences or collisions caused by using the most recent template.

# Process - Crime Document

## Create the Shell Database

For best results, restore the shell database to the same instance as the current production database and run each step from there.

1. CreateDatabase-EmptyTemplate.sql
   * Builds out an empty database with the suggested settings and properties.
   * Adjust the target data and log file sizes as needed to reflect the reduced size of the database post-cleanup.
2. Export-Publish-dbaDacPackage.ps1
   * Generates a dacpac of the source database, saves it to a local folder, and immediately publishes to the new database.
   * Specify the shell database as the destination.
   * Will automatically create the database if it does not already exist, but this is ill-advised.

## Prepare the Tables

1. ToggleForeignKeys.sql
   * Prevents foreign key conflict errors while the bulk inserts are performed.
   * Drop any foreign keys that either reference or are defined on the table being purged. Save both DROP and CREATE scripts.
   * Toggle off all other foreign keys. Run script and execute output for NOCHECK.
2. Drop non-clustered indexes on the target table in the new instance.
   * Improves performance of insert.

## Move the Data

Each step can be run simultaneously.

1. Copy-DbaDbTable-AdhocBlobCleanup.ps1
   * Copies data from a table in the source database to the destination database.
   * Conditions can be added in the script to selectively preserve blob data.
2. Copy-DbaDbTableData-ExcludingComputed.ps1
   * Copies data from all tables that do not have a computed column, which prevents seamless intelligent copy.
   * Pass in at least the target table to the ExcludeTables array and any others to ignore in the one-to-one copy process. Currently only supports the table name with no regards to schema.
3. InsertIntoComputedColumnTables.sql
   * Generates the INSERT INTO SELECT FROM statements, working around the computed columns, and dynamically accounting for identity insert.
   * Execute script and run output.

## Finalize the New Database

Once cleanup is complete, the database structure needs to be set back to its original state before it can be swapped in the application.

1. ToggleForeignKeys.sql
   * Recreate the foreign keys dropped on the cleaned table. Definitions should have been saved off when the drop statements were created.
   * Run the CHECK column output for the remaining foreign keys.
2. Re-add the non-clustered indexes to the cleaned table.
3. Swap the name of the old and new databases.
4. Proceed with the standard conversion process.