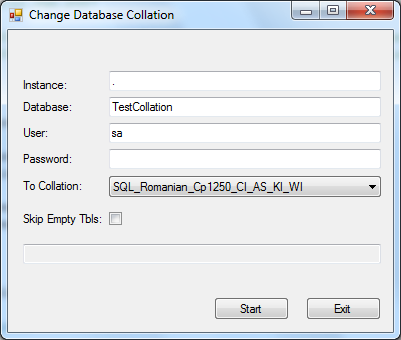
# Procedure

## Prepare for migration

You need two databases:

* Model (source) database, the database that will be used to generate migration scripts
* Target database, usually a copy of model database obtained by backup/restore

## Run MigrateDataFileScripts



Provide the connection data:

* Instance name (server name, using <Server>\<Instance> form if required
* Model database name
* User name
* Password

To Collation is the desired collation of the destination (target) database.

Starting execution will generate a sequence of scripts in Scripts folder. This might take a while (even 1 hour) for large databases.

After script creation finishes check ErrorsCreate.txt to identify the objects for which a create script was not generated due to raised errors. Remove the cause and rerun the tool.

**Important notices**:

* Please run the mentioned scripts in the exact order specified below. Apart from that scripts also other scripts/batches are generated for other purpose.
* Scripts must be run on target database from within MS SQL Management Studio

## Run Drop.sql

It might take several minutes. Be sure that no errors are reported at the end of run.

The scripts drops primary keys, foreign keys, computed columns, table/column level constraints as well as user defined functions that return a table

## Change manually database collation

You will have to switch the database to single user mode before this operation. Use Properties->Options and be sure you specify the same collation as specified when generating scripts

## Run AlterCollation.sql

## Run RefreshViews.sql

This script is provided with the tool, not generated. It will recompile views so errors might occur if some tables/columns were dropped since the view was compiled last time

## Run Create.sql

It will recreate all objects dropped previously so that altering the collation is possible. Some errors might occur for user defined functions if referred objects are missing from the database.

## Run RefreshInlineUDF.sql

Similar to RefreshViews.sql but applies to user defined functions.

# Final check

In order to check that all columns were correctly converted you can run the following (on target database):

select \* from sys.columns where collation\_name like 'SQL\_Latin1\_General\_CP1\_CS\_AS'

(Where 'SQL\_Latin1\_General\_CP1\_CS\_AS' must be replaced with the collection of the **source** database). No record must be returned if all columns were migrated to new collation

# Additional scripts

Beside the scripts mentioned above some other scripts/batches are generated in order to allow additional operations.

## Empty database

On a copy of the database:

* Run Drop.sql
* Run Truncate.sql
* Run Create.sql

## Migrate data

Do the following:

* On source database
  + Change Export.bat (provide actual server name, username, password, etc.)
  + Run Export.bat
* On target database:
  + Change Import.bat
  + Run Import.bat

One scenario where the above mentioned operations might be useful is changing the way tables are distributed on file groups (export data, empty the database, do the changes, import the data).