Migrator Design Doc

The main purpose of Migrator is database migrations similar to how Ruby on Rails. You write small migrations in separate files and then Migrator applies them to a database taking care not apply the same migration twice.

The other feature of Migrator is creating a Schema of an existing database. You point Migrator at an existing database and it creates a complete Schema of the database including tables, view, and stored procedures.

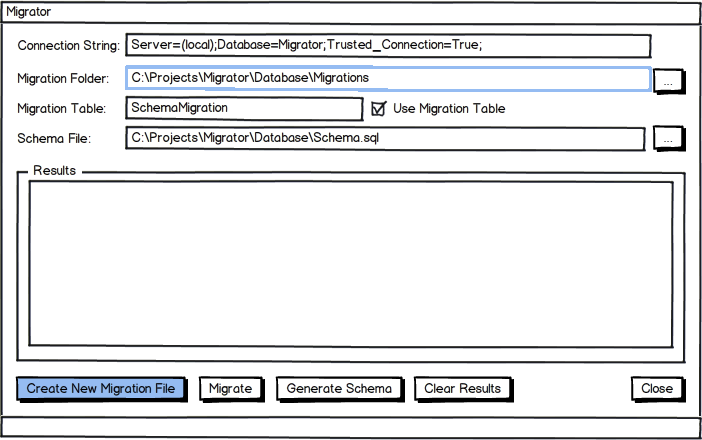
# The User Interface

There is only one main form for doing everything.

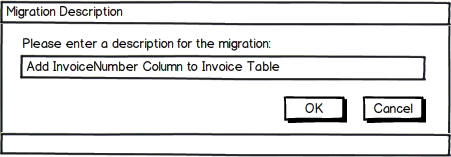
<It really sucks and could use some TLC>

## Creating and Running a Migration

Before we can run a migration we need some migration files. Create a folder to hold all these cute little migration files. Usually this folder is called “Migrations” but you can call it whatever you want. Once the folder is created run the Migrator application and make sure the Migration Folder points to the folder you just created. Then click the “Create New Migration File” button.



Then enter a description of the migration. This will become the file name so don’t make the description too long.



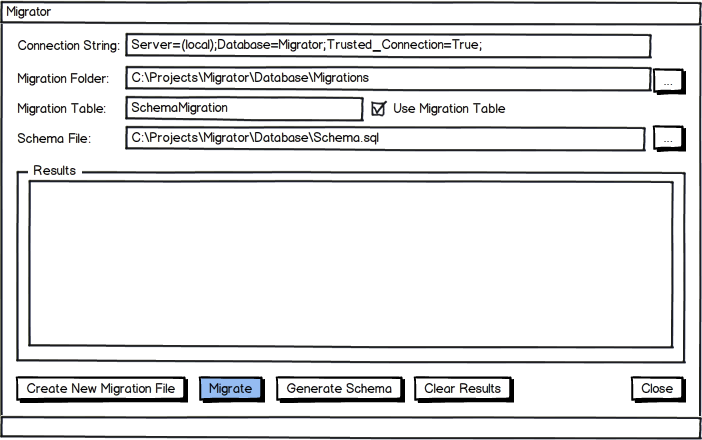
This will create a file in the Migrations file called:

20120602153289\_Add\_InvoiceNumber\_Column\_to\_Invoice\_Table.sql

The number at the front of the file is a unique number for the migration in question. Open up the migration file in notepad or a SQL editor and it will be blank. Enter the SQL for the migration, in this case adding a new column.

-- Missed the invoice number column  
-- when the table was first created.  
Alter Table Invoices  
Add InvoiceNumber Varchar(25) NULL  
GO

It’s just standard SQL, nothing fancy. Now to actually apply this migration to your database go back to Migrator and check the connection string is pointing to the correct database. Then



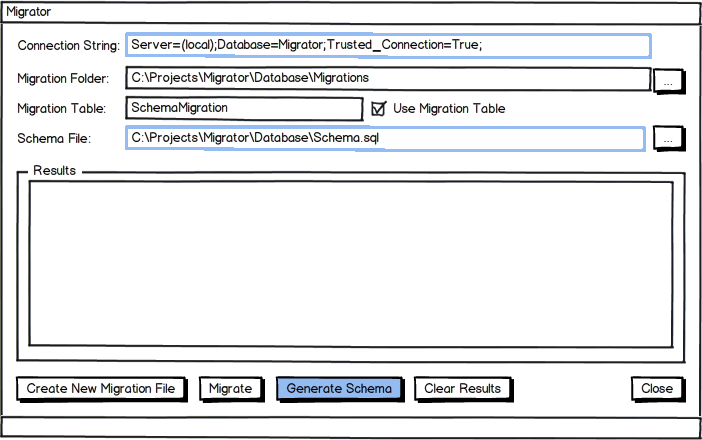
When you click the migrate button each migration file is checked. If the migration has already been ran against the database then nothing happens, the SQL inside the file is ignored. If the migration has not been applied to the database yet then the SQL inside the migration file is executed against the database.

To check if a migration has been applied to the database Migrator looks in the migration table for the unique number of the file. If the number exists then migration has already been applied.

The results of the migration, including any errors, are written to the results windows.

## Generating the Schema

Migrator can create a schema of an existing database including the tables, views, stored procedures, etc. To create the schema update the connection string to point to the database you want the schema of and then set the file where you want the schema saved. Once you are done with the settings then click the Generate Schema button.



Be patient while the schema is generated. Once done you can view the results in the schema file.

# Command Line

You can either call command line with no arguments, in which case it will use the settings in the app config file, or you can pass in the arguments. The command line looks like:

migrator –cs <connectionstring> -mf <migrationfolder> -vt <versiontable>

migrator –cs <connectionstring> -sf <schemafile>

The first command is for running a migration and second one is for generating a database schema.

<Note: The command line is only partially implemented>.

# Internals

The application will have the following projects: