

Database Terminator

Explanatory Note

Annotation

A WPF application for displaying the provided database in Microsoft SQL Server format.

Central repository address

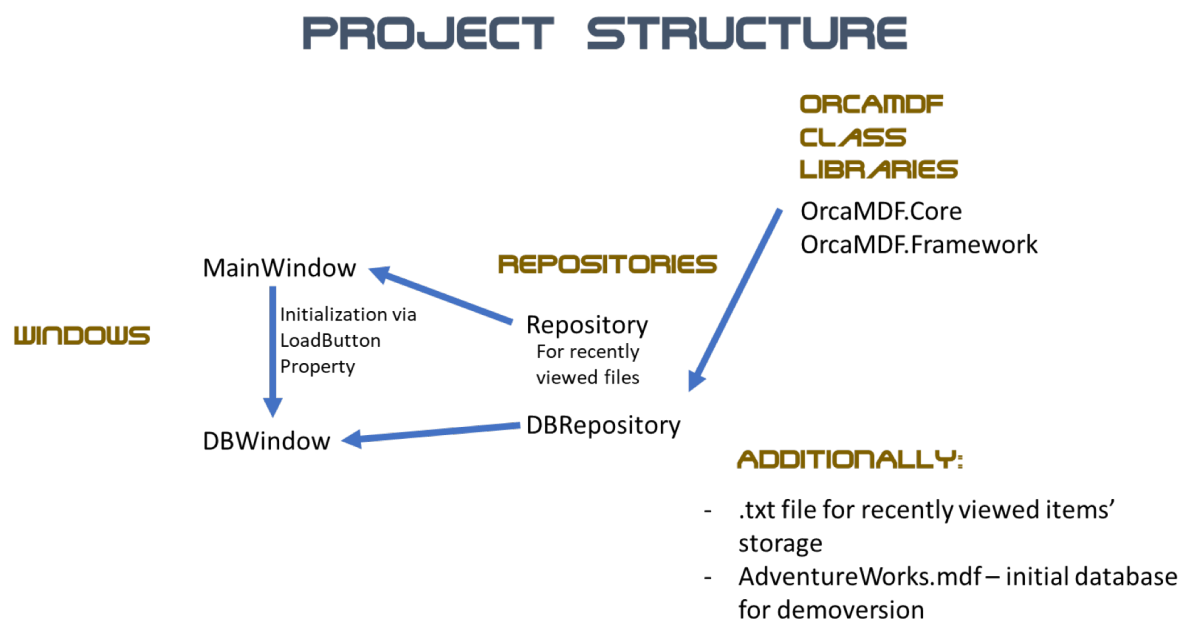
<https://github.com/boris453/DbTerminator>

Team members

Boris Belyakov 164 — UI, Main application logic

Ivan Reshetsov 164 — Main application logic, testing

Classes description



1. OrcaMDF represents the class library of our project for reading the database and actually makes it work. Creating an appropriate parser, converting mdf-files for C#-readable format was undoubtedly the most complicated part of our work. Despite making maximum efforts and using all our skills and creativity, we have decided that we are not as much experienced as needed to make a solid and effective parser. Therefore, we have imported OrcaMDF into our project. It comprises two libraries, containing classes with overridden SQL data types,

clustered indexes, keys, tables, columns, etc. Apparently, they are utilized to parse four crucial system tables: sysallocunits, syssschobjs, sysrowsets, sysrowsetcolumns. It is crucial to mention OrcaMDF.Core.Engine.Database class that uses a string filename to instantiate a database from a single data file. The parsing of SQL data types is performed with the help of abstract programming, specifically abstract SqlTypeBase class implementing ISqlType interface.

2. DbRepository, as it can be noticed from its name, is a repository class of the ModelTerminator library, containing two repository classes. DbRepository provides a coupling between class library and WPF elements. It contains essential methods for generating TreeView representation of the database structure (Tables, Views, Stored Procedures) in DbWindow.
3. Repository is a class designed for coupling between UI and logic, implementing saving recently viewed files into a text file and loading it into the MainWindow's RecentlyViewed listbox
4. MainWindow provides an opportunity to download the mdf-file for converting it to a readable view. It initializes a DbWindow, designed directly for database representation and contains a listbox with recently viewed files as a part of user-friendly interface. Each file downloaded by a user is immediately saved to a .txt collection which forms the source of RecentlyViewed listbox. Additionally, the listbox contains AdventureWorks.mdf file intended for a demo-version of the project's demonstration.
5. DbWindow is designed directly for representing a database in WPF through a TreeView mode and a DataGrid. This window is loosely coupled with a ModelTerminator library via DbRepository.