

Project “Comparator”

Class Master (Master.cs)

Object of class Master is created at application startup and controls the operation of the application in one of the modes - window or batch.

This object:

- manages application settings - creates / reads / saves an object of the Options class (see Options.cs in project “Common”);
- manages profile settings - creates / reads / checks / saves an object of the Profile class;
- in window mode:
 - subscribes to events of objects that implements IView and IViewSource interfaces (see Views.cs in project “Common”). In this application winform-class FormView implements IView (see FormView.cs) and user-control SourcePanel implements IViewSource (see project “Sources”);
 - defines objects for transferring Profile data to IView and IviewSource;
 - controls the process of select fields pairs;
- starts processes of preparing data from sources, invoking methods for receiving content data (see descendants of abstract SourceContent class in project “Sources”);
- starts comparison of the received data by calling method of DSComparer static class (project “DSComparer”) and receiving comparison result as object of CompareResult class:
- controls the progress of comparison process and its interruption;
- controls the output of comparison result - to a form (window mode) or to a file (batch mode);
- controls the sending of comparison result (in batch mode) via the static Mailer class (see SendMail.cs in project “Common”)
- supplies the necessary messages using an object that implements ILogger interface (see Common.cs in project “Common”). In batch mode, this is an object of Logger class that uses methods of static class Log to write to the log file. In window mode the output of messages is made through FormView class that implements ILogger interface.

Class FormView (FormView.cs)

Represents the application's user interface.

It implements IView interface for creating, modifying, verifying and saving Profile data, for starting comparison process and for viewing the latest comparison results.

Contains two UserControls that implement the IViewSource interface.

To select pairs of fields for comparison, opens FormSelectPair form.

When the comparison starts, it opens FormCompare form to display the progress of data receiving and comparison with the possibility of process interruption.

Project “Sources”

Class Source (Source.cs)

Describes the data source for comparison. Objects of this class are created with object of Profile class (in project Comparator) for each of the two sources.

Object of Source class contains an object property of one of the descendant classes of the abstract SourceContent class. It is DbContent, ExcelContent, CsvContent and XmlContent classes that provide an implementation of methods for receiving data from a source for a particular type.

Features:

- class ExcelContent - its methods use methods of static class ExcelProc (see ExcelProc.cs in project “Common”) for access to opened Excel books
- class DbContent - implements the interface ISqlModel, which describes the connection and query,

uses SqlController class to receive data (see SqlController.cs in project "SqlSource")

- class XmlContent – uses XmlController static class to work with XML data - reading, xsl-transformation and conversion to a table.

Class SourcePanel (SourcePanel.cs)

This is UserControl, which displays the settings of the selected source.

For DB and XML sources it opens forms with detailed settings and the ability to work with source data -

SqlView (see SqlView.cs in project "SqlSource") and FormXml.

For Excel and CSV sources, it opens a form for quick data viewing - FormFlatData (see FormFlatData.cs in project "Common")

Project "SqlSource"

Class SqlController (SqlController.cs)

Creates or receives at creation an object that implements ISqlModel - connection and query parameters.

Creates or receives at creation an object that implements ISqlView (see Views.cs in project "Common") - an interface for configuring the connection and executing query. In this application winform-class SqlView implements ISqlView .

Sends data from ISqlModel to ISqlView, subscribes to ISqlView events.

Controls the progress of the data receiving process and its interruption.

Project "DSComparer"

Class DSComparer (DSComparer.cs)

Contains the Compare method, which compares two DataTables according to the specified settings and returns a CompareResult object with comparison results.

Class CompareResult (CompareResult.cs)

Contains the comparison results and methods for their output in html, excel and form FormResult.

Project "Common"

Contains general procedures and definitions for the projects described above.

Note: Some classes and projects have no dependencies of the other parts of the application and can be used separately. For example – project "SqlSource", project "DSComparer", FormXml with XmlController, static classes from project "Common"