

IDS Desktop Software Design Document

Miriam Rotella Alter

Andrew Hlynka

May 13, 2021

Section 1 – Project Description

1.1 Purpose of this Document

This document contains technical information on the IDS Transposer for converting data to the Intermediate Data Structure (IDS), specifically the modified Desktop GUI version. The original project was developed with a web-interface in mind; this version modifies the existing project to run with a Desktop GUI, to run locally on a machine with a graphic interface, or locally without a graphic interface through a command prompt.

Redundant information from the original project is not included in this document. Refer to <https://github.com/ICPSR/IDS> for more information on the original project.

1.2 Requirements to Run Program

The original IDS Transposer was web-based, and could be accessed through a web-browser, requiring an internet connection and an email address to receive output results. This Desktop variant can run locally on a computer without an internet connection.

Java 8 JRE (Java Runtime Environment) is required to run the pre-compiled .jar application.

Section 2- Application Design

2.1 Technologies Used

The technologies used in this Desktop variant are Java, the Java Swing library. The original source code was modified and compiled using Apache NetBeans IDE.

2.2 App.java

In addition to the primary, reused classes (IDSTransposer.java, IDSEException.java, and IDSTransposerController.java) that act as the backend for both the original web app and this Desktop variant, a new class named App.java is added to handle the Desktop GUI and user input.

The Swing-based GUI uses the Grid Bag Layout design to determine placement and size of components.

Functionally, this class replaces the JSPs component of the original web-based project.

The following is a list of important functions and a brief description of their purpose:

<code>public static void main(String[] args)</code>	The entry-point for the project. By default, it launches the Swing GUI (defined within App.java). If command-line arguments are also provided at launch, this function reads them to determine if the app should run with or without the GUI.
<code>private void initComponents()</code>	Defines UI components, position, and size, initializes GUI window.
<code>private void submitForm (java.awt.event.ActionEvent evt)</code>	Upon "Go" button being clicked, input parameters are sent to backend code to generate results.
<code>private void browseTargetFolder (java.awt.event.ActionEvent evt) , private void browseEntityFile (java.awt.event.ActionEvent evt) , private void browseRelationshipFile (java.awt.event.ActionEvent evt)</code>	Open file browser to allow user to select file or folder as an input parameter.
<code>private void addFile (java.awt.event.ActionEvent evt) , private void removeFile (java.awt.event.ActionEvent evt) {</code>	Open file browser to allow user to select or remove multiple optional input files as an input parameter.
<code>private void formComponentResized (java.awt.event.ComponentEvent evt)</code>	Resize font based on screen size, when user changes size of application window.

2.3 IDSTransposerController.java

The following is a list of important new functions and a brief description of their purpose:

<code>public void runInCmd (String inputFile)</code>	If App.java -> main function recognizes user specifies to not run with GUI, and provides an input file with a list of file parameters, this function is called to parse out files and directories from that input file, and relayed to the backend logic.
--	---

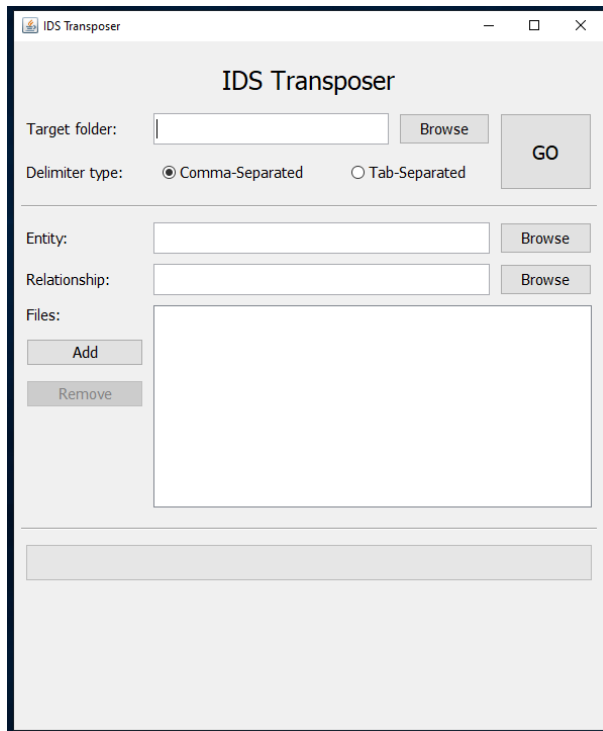
Section 3 – User Guide

3.1 Downloading App

The GitHub repository providing the source code for this Desktop variant also has the pre-compiled .jar executable file. This file alone is all a user needs to download to run a local instance of the program (both GUI and command-line versions).

The .jar file is available under */target/IDS-Transposer-App-jar-with-dependencies.jar* .

3.2 GUI version



The GUI version of the Desktop variant is the default version of the program. It can be launched by simply double-clicking on the .jar file, like you would with a standard .exe file.

Alternatively, you can launch this app with the following command, no input parameters:

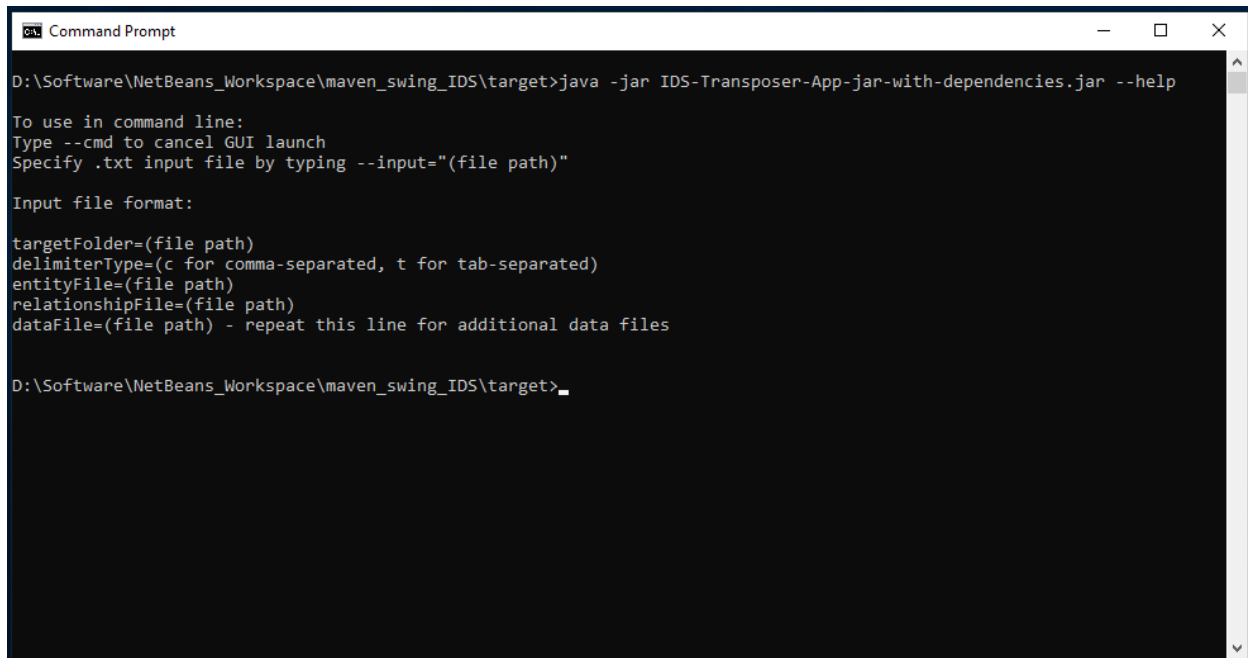
```
java -jar IDS-Transposer-App-jar-with-dependencies.jar
```

The user must fill the text boxes to provide input parameters. The 4 input properties are “*Target Folder*” (output directory for resulting files), “*Entity*,” “*Relationship*,” and “*Files*.” Additionally, “*Delimiter type*” can be specified to better configure the output files.

The 4 input properties are all files, with absolute file paths. The corresponding “Browse” buttons for “Target Folder,” “Entity” and “Relationship” opens a folder browser to easily find the folder or file you want to use. “Files” allows the user to specify multiple optional files for the configuration (unlike the 20-file limit of the Web application, there is no limit in this Desktop variant).

When all input properties are defined, the user hits the “GO” button to run the program and generate output files. A success or fail message will be displayed at the bottom of the GUI when the program finishes. Unlike the original Web application, the resulting files are not combined into a .zip file.

3.3 Command-line version



```
cmd Command Prompt
D:\Software\NetBeans_Workspace\maven_swing_IDS\target>java -jar IDS-Transposer-App-jar-with-dependencies.jar --help

To use in command line:
Type --cmd to cancel GUI launch
Specify .txt input file by typing --input="(file path)"

Input file format:
targetFolder=(file path)
delimiterType=(c for comma-separated, t for tab-separated)
entityFile=(file path)
relationshipFile=(file path)
dataFile=(file path) - repeat this line for additional data files

D:\Software\NetBeans_Workspace\maven_swing_IDS\target>
```

To launch the Desktop variant without the GUI, you can run the .jar file in a command prompt with input parameters.

Using the “—help” parameter provides some basic information on user options. You can run the —help command with the following:

```
java -jar IDS-Transposer-App-jar-with-dependencies.jar --help
```

To specifically run without the GUI, you can run the app using the “—cmd” parameter (run the “command line” version instead of the GUI version). For example:

```
java -jar IDS-Transposer-App-jar-with-dependencies.jar --cmd
```

However, this mode requires you specify the expected input files and directories. The user must provide this using the “—input=” parameter, specifying the file and file path of a .txt file that provides the expected parameters.

The format of an example .txt input file looks like the following (keep in mind the separation by lines, case-sensitive spelling, and lack of spacing):

```
targetFolder=C:\\...\\  
delimiterType=c  
entityFile=C:\\..\\entity.csv  
relationshipFile=C:\\..\\relationship.csv  
dataFile=C:\\..\\CHILDREN_INPUT.csv  
dataFile=C:\\..\\PARENTS_INPUT.csv  
dataFile=C:\\..\\PLACES_INPUT.csv
```

Notice that “targetFolder,” “delimiterType,” “entityFile,” “relationshipFile,” and “dataFile” are specified. “dataFile” is optional, and can be repeated for multiple files.

Suppose you have an input file named “inputFile.txt” stored at C:\. You would run the app with the following command:

```
java -jar IDS-Transposer-App-jar-with-dependencies.jar --cmd --input="C:\inputFile.txt"
```