

User Manual: Core Practice Data Migration Tool

Introduction

The Core Practice Data Migration Tool is a C# WinForms application designed to simulate the Extract, Validate, Transform, and Load (EVTL) process for migrating legacy dental practice data (Patient and Treatment records) into the new Core Practice database schema.

The tool ensures **data integrity** by performing mandatory field checks and validating **Foreign Key (FK) relationships** before generating the final, structured SQL INSERT statements.

1. The Migration Workflow

The migration process is organized into a clear, four-step workflow, controlled by the buttons in the action bar:

Step	Button	Purpose
1	Load CSVs	Loads the source <code>patients.csv</code> and <code>treatments.csv</code> files into the application's memory.
2	Run Validation	Executes data quality checks and Foreign Key integrity checks, filtering out invalid records.
3	Ingest Data (Simulate DB)	Performs the Transform stage: groups treatments into invoices, assigns new database IDs, and generates the final SQL INSERT statements.
4	Copy SQL to Clipboard	Copies the generated SQL from the log, preparing it for external database import.

2. Detailed Step-by-Step Guide

Step 1: Load CSVs

- Click the 1. Load CSVs button.
 - Action:** The tool attempts to read the contents of `patients.csv` and `treatments.csv` from the simulated environment.
 - Result:** The application will populate the "1. Patient Data Review" and "2. Treatment Data Review" tabs with the loaded raw data. The **Validation** button will become enabled.

Step 2: Run Validation

- Click the 2. Run Validation button.
 - Action:** The tool performs two primary checks:
 - Mandatory Field/Format Check:** Ensures required fields (e.g., Patient Name, DOB, Treatment Item Code, Fee, Date) are present and in the correct format (e.g., date is `YYYY-MM-DD`).
 - Foreign Key (FK) Check:** For treatments, it verifies that the `PatientID` referenced in the `treatments.csv` file exists in the list of **valid** patient records.
 - Result:**
 - The data grids in Tab 1 and Tab 2 are updated to show **only invalid rows**, highlighted in pink.
 - The **summary labels** above the grids indicate how many valid records are ready for the next step.
 - The tool automatically switches to the "3. Validation Summary & Errors" tab, providing a detailed log of all identified errors, including the affected row index and the reason for failure.
 - The **Ingest Data** button becomes enabled (if valid records exist).

Step 3: Ingest Data (Simulate DB)

1. Click the 3. Ingest Data (Simulate DB) button.
- Action: This is the core transformation and ingestion simulation phase. The tool:

Creates a mapping between the old CSV IDs and new, sequential Internal Database IDs (PatientId, TreatmentId, InvoiceId).

Groups Treatment Records: Treatments for the same Patient on the same Date are grouped together to create a single Invoice (tblInvoice).

Generates SQL INSERT statements for four target tables: tblPatient , tblInvoice , tblTreatment , and tblInvoiceLineItem .

Result:

The tool switches to the "4. Data Ingestion Log (SQL)" tab, displaying the full, executable SQL script.

The Copy SQL to Clipboard button becomes enabled.

Step 4: Export to Database Browser

This final step moves the processed data out of the tool and into your target database environment (e.g., DB Browser for SQLite).

1. Click the 4. Copy SQL to Clipboard button.
- Action: The entire contents of the Data Ingestion Log are copied to your system's clipboard.
2. External Action:
- Open DB Browser for SQLite (or your preferred SQL tool).

Ensure you are on the "Execute SQL" tab.

Paste the copied SQL statements into the editor.

Click the "Execute all/selected SQL" button to run the script and populate your database tables.

3. Data Mapping and Schema

The tool maps the CSV data to the following target tables in the Core Practice schema, generating sequential IDs as it goes:

Target Table	Source Data	Transformation Logic
tblPatient	patients.csv	Primary Key: PatientId (new, sequential INT). The CsvId is stored in the PatientIdentifier field for traceability.
tblInvoice	Grouped treatments.csv	Primary Key: InvoiceId (new, sequential INT). Created once per Patient per day of treatment. FK: PatientId is linked.
tblTreatment	treatments.csv	Primary Key: TreatmentId (new, sequential INT). FKs: Linked to the new PatientId and the new InvoiceId .
tblInvoiceLineItem	treatments.csv	Created one line item for every treatment record. FKs: Linked to the new PatientId , TreatmentId , and InvoiceId .