KNOLLEDGE CONSULTING AND SOLUTIONS LLC
STACI KNOLL, CEBS, RPA
Founder and CEO
1417 Broadway, Penthouse
Denison, IA 51442
StaciKnoll@KnollEdgeHub.com
#: 402-659-8016
www.KnollEdgeHub.com
www.KnollEdgeConsultingandSolutions.com

MERGER MARJORIE©

JOIN FILES. ALIGN DATA. TRUST THE RESULTS.

OBJECTIVE

Merger Marjorie was developed to solve a common and frustrating problem faced by analysts, HR teams, finance departments, and operations professionals: ***merging reports from multiple systems into a single reliable dataset****.*

In many organizations, critical business information is spread across different platforms such as payroll systems, HR databases, accounting software, benefits systems, and operational reporting tools. When teams need to perform analysis, produce reports, or prepare compliance documentation, these datasets must be combined into a single spreadsheet.

Traditionally, this work is performed manually using spreadsheets.

Analysts export reports from each system and attempt to combine them using spreadsheet formulas, sorting, copying columns, and lookup functions. While workable, this process can be time-consuming and error-prone. Small formatting differences, inconsistent identifiers, duplicate records, or incorrect formulas can lead to inaccurate results and require repeated troubleshooting.

Over time, what begins as a simple task often evolves into a fragile and repetitive process that must be recreated each time new reports are generated.

Merger Marjorie was designed to replace this manual workflow with a structured and reliable data integration process.

Instead of manually manipulating spreadsheets, users can load two datasets, define how the records should align, and generate a ***validated, ordered, analysis-ready output dataset****.*

The objective of Merger Marjorie is to reduce the time, risk, and complexity associated with manual spreadsheet merging while improving data integrity and reproducibility.

Merger Marjorie provides organizations with a structured and auditable data integration process that:

- *eliminates repetitive manual merging work*
- *reduces spreadsheet formula errors*
- *improves auditability and data lineage*
- *enables consistent and repeatable integrations*
- *produces standardized analysis-ready datasets*

By transforming a fragile manual process into a repeatable workflow, Merger Marjorie helps organizations prepare data more quickly for the next steps in ***analysis, marketing, operational decision-making, and regulatory reporting****.*

SCOPE

*Merger Marjorie is designed to support **structured data integration involving two independent datasets**. The application enables users to combine and validate information from separate systems to produce a single, ordered dataset ready for downstream use.*

Merger Marjorie is particularly useful when organizations must combine reports exported from different operational systems—such as payroll platforms, HR systems, accounting software, benefits administration tools, customer databases, or marketing platforms—into a single validated spreadsheet.

*The application focuses on solving the common challenge of **aligning structured reports between systems that were not designed to communicate directly**, allowing users to perform controlled data merges without complex spreadsheet formulas or custom scripts.*

Merger Marjorie can be used across a wide range of operational, analytical, and compliance workflows, including:

- ***health insurance census files and implementation accumulation files** for carriers and brokers*
- ***ACA employer reporting datasets** used to generate IRS Form 1095-C records*
- ***payroll integration datasets** used for financial reconciliation and system migration*
- ***marketing analytics datasets** combining CRM and campaign performance data*
- ***financial reconciliation datasets** aligning operational transactions with accounting records*

Supported File Formats

Merger Marjorie currently supports the following structured data formats:
- *Microsoft Excel (.xlsx)*
- *Microsoft Excel (.xls)*
- *Comma-Separated Values (.csv)*

These formats represent the most common export formats used by payroll systems, HR platforms, accounting systems, and operational reporting tools.

Core Processing Capabilities

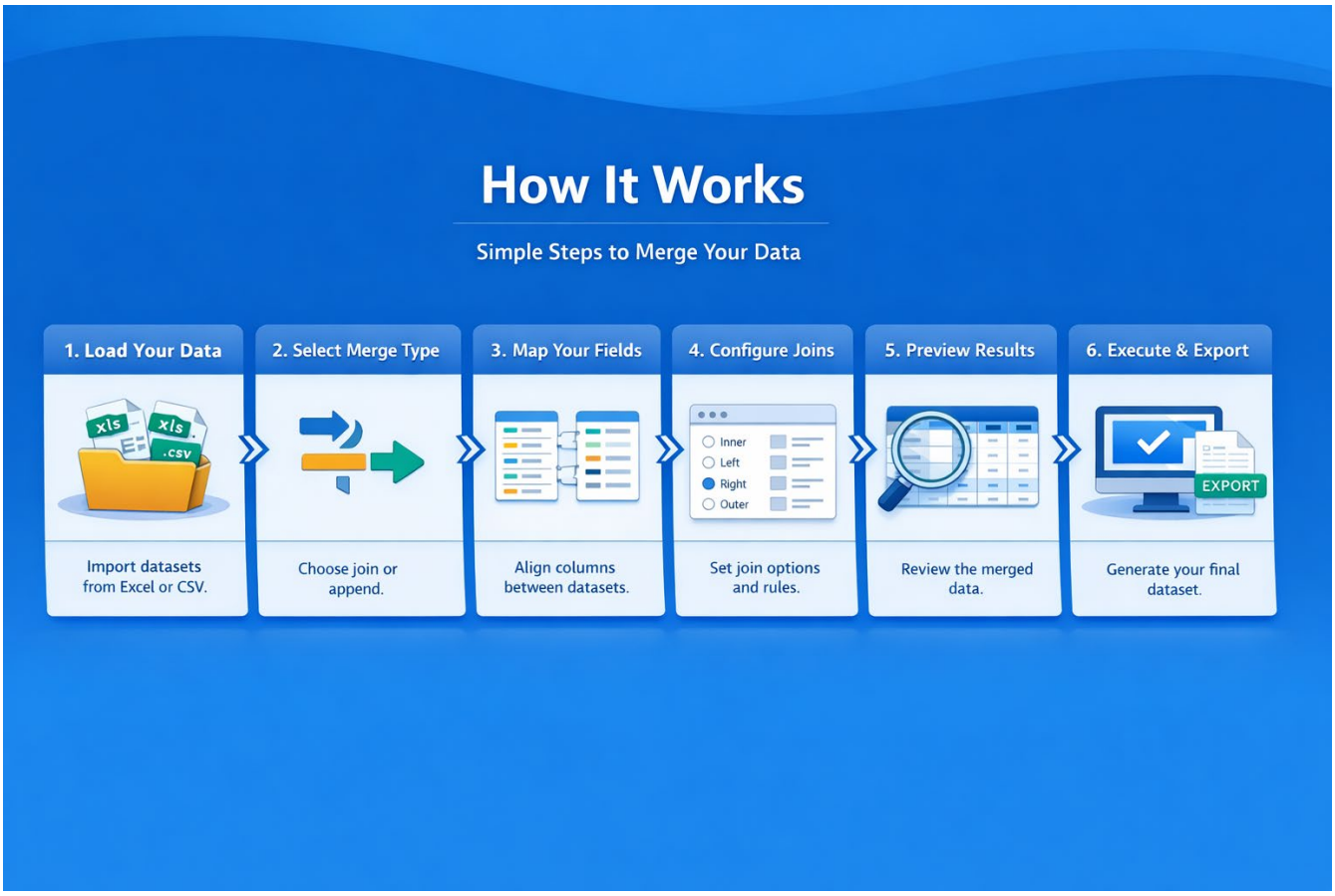
During the merge workflow, Merger Marjorie performs several automated processing steps to ensure reliable results. These include:

- *dataset ingestion and file validation*
- *column and schema inspection*

- *data cleaning and normalization*
- *primary key matching and record alignment*
- *type compatibility validation*
- *duplicate detection and handling*
- *configurable join operations*
- *ordered output dataset generation*

*The final output is a **clean, validated dataset suitable for analysis, reporting, reconciliation, compliance preparation, or system import**.*

Throughout the workflow, Merger Marjorie maintains validation checks and structured processing controls to improve data integrity and reduce the risks commonly associated with manual spreadsheet merging.



PRIVACY AND SENSITIVE DATA HANDLING

Merger Marjorie is designed for organizations that regularly work with sensitive business data such as employee records, payroll information, financial transactions, and customer data. Protecting this information is a core design principle of the application.

Local Processing Architecture

*Merger Marjorie operates as a **desktop application installed on a user's local Windows workstation**.*

All data processing occurs locally within the user's computing environment. The application reads input files, performs the merge operations, and generates output files entirely on the user's machine.

*Merger Marjorie does **not require cloud connectivity or external servers** to perform data merges.*

No Cloud Data Transmission

*Merger Marjorie does **not transmit, upload, or store customer data in any cloud service**.*

All input files remain within the organization's controlled environment. Data is processed only in memory during execution and written to the output file location selected by the user.

This design ensures that sensitive data—such as payroll records, Social Security numbers, benefits enrollment data, or financial transactions—remains under the direct control of the organization using the software.

Customer Data Ownership

Users and their organizations retain full ownership and control of all datasets processed by Merger Marjorie.

The software does not claim ownership of customer data and does not collect, store, or access user datasets outside the local environment.

If customer data is voluntarily shared with the software provider for troubleshooting or support purposes, that data is used solely for the purpose of resolving the support request.

Recommended Data Security Practices

Organizations using Merger Marjorie should follow standard data governance practices when working with sensitive datasets.

Recommended practices include:

- *storing datasets in secure internal locations*
- *restricting access to authorized personnel*
- *protecting workstations using organizational security policies*
- *encrypting storage devices when required by policy*
- *validating output files before sharing or distributing reports*

Appropriate Use of Sensitive Data

Merger Marjorie is commonly used to process datasets containing confidential information, including:

- *employee records*
- *payroll and compensation data*
- *benefits enrollment information*
- *Social Security numbers or employee identifiers*
- *financial transaction records*
- *customer or vendor lists*

Users remain responsible for ensuring that all data processed within the application complies with their organization's privacy policies and applicable regulatory requirements.

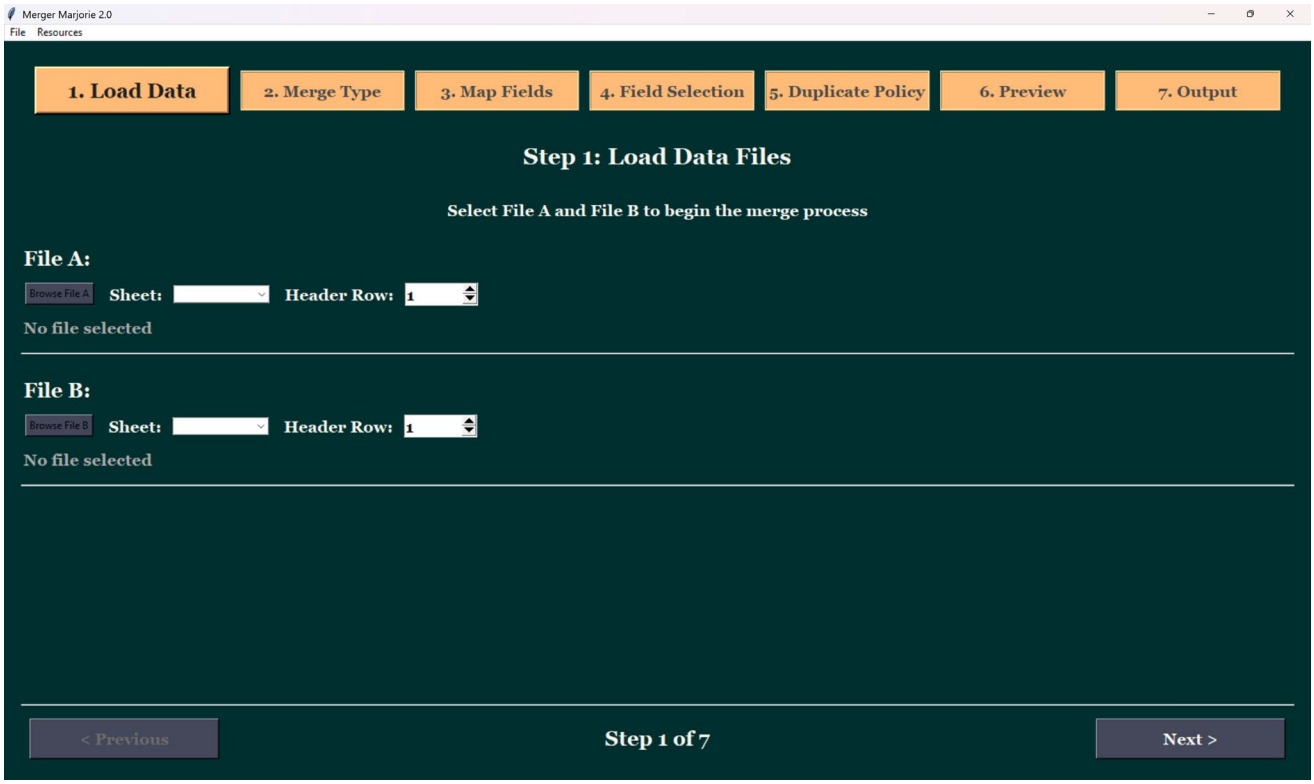
Summary

Merger Marjorie is designed to provide a secure and controlled data integration workflow by operating entirely within the user's local environment. By eliminating the need for cloud data processing, the application helps organizations maintain full control over sensitive datasets while reducing the risks associated with manual spreadsheet merging.

MERGER MARJORIE USER GUIDE

Application Workflow

Merger Marjorie guides users through a structured workflow designed to ensure data integrity, repeatability, and auditability. Each step in the workflow performs a specific function within the data integration pipeline.



STEP 1 — LOAD DATA FILES

What This Step Does

This step loads the two datasets that will be merged. The system reads the files and performs initial ingestion and metadata extraction, including:

- *file format detection*
- *row count and column count identification*
- *header detection*
- *worksheet selection for Excel files*

Supported file formats include CSV and Excel files.

Why It Matters

Correctly loading the datasets ensures that Merger Marjorie can properly analyze the structure of each file.

During ingestion the system prepares the datasets for validation and transformation by:

- *standardizing column names*
- *cleaning string formatting*
- *removing invalid or null rows*

These processes reduce common data quality issues before the merge begins.

How to Use It

1. *Click **Browse** to select the **Source Dataset (File A)**.*
2. *Select the file containing the first dataset.*
3. *If the file is Excel, choose the appropriate **worksheet**.*
4. *Specify the **header row** if necessary.*
5. *Repeat the process to load the **Target Dataset (File B)**.*

Once both files are loaded successfully, proceed to the next step.

STEP 2 — SELECT MERGE TYPE

What This Step Does

This step defines how the two datasets will be combined. Merger Marjorie supports two merge strategies:

Horizontal Join

Records from two datasets are aligned using a shared key field.

Vertical Join

Datasets with similar structures are stacked together.

Why It Matters

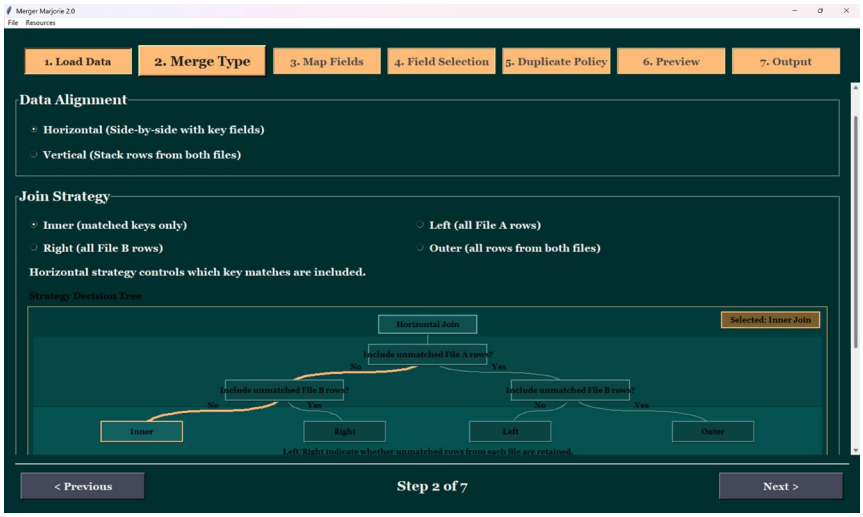
Selecting the correct merge type ensures that data is combined logically.

*Choose **Horizontal Join** when datasets contain related information about the same entities.*

*Choose **Vertical Join** when datasets contain similar records that should be appended into a larger dataset.*

How to Use It

1. *Review the structure of both datasets.*
2. *Select **Horizontal Merge** if records should be matched using a key field.*
3. *Select **Vertical Merge** if records should be appended together.*
4. *Click **Next** to proceed to field mapping.*



STEP 3 — MAP FIELDS

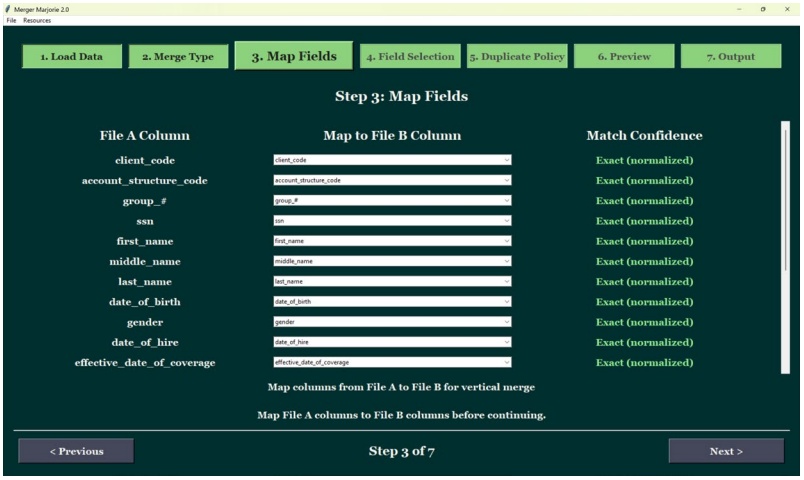
What This Step Does

Field mapping defines how columns from the two datasets correspond to one another.

Users select which columns should:

- *be included in the final dataset*
- *align between datasets*
- *serve as primary key fields (for horizontal joins only)*

Field mappings also allow column ordering and renaming within the output dataset.



Why It Matters

Correct field mapping ensures that the merge aligns the correct information between datasets.

Improper mappings can result in:

- *incorrect joins*
- *missing data fields*
- *duplicate columns*

This step provides full control over how the final dataset will be structured.

How to Use It

1. *Review the list of fields from both datasets.*
2. *Select fields that should be included in the final dataset.*
3. *Map related fields between datasets.*
4. *Identify the **Primary Key field** for horizontal joins.*
5. *Adjust column order if desired.*

After completing the mappings, proceed to join configuration.

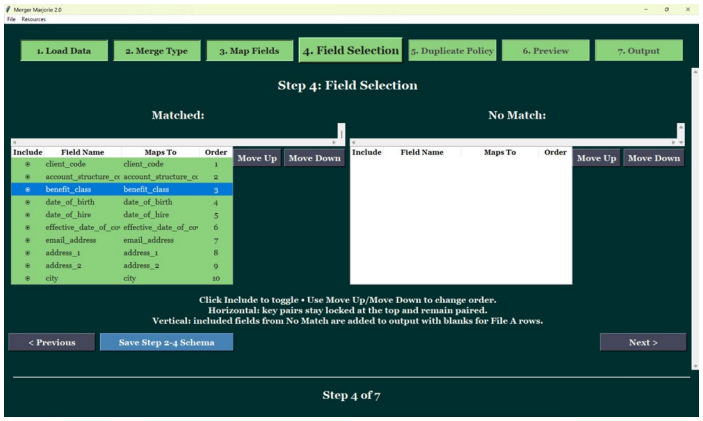
STEP 4 — CONFIGURE JOIN SETTINGS

What This Step Does

This step allows users to define the rules governing the merge operation.

For horizontal joins, supported strategies include:

- ***Inner Join*** *– keep only matching records*
- ***Left Join*** *– keep all records from Dataset A*
- ***Right Join*** *– keep all records from Dataset B*
- ***Outer Join*** *– keep all records from both datasets*



For vertical joins, options include:

- ***Append*** *– stack datasets together*
- ***Union*** *– combine only shared columns*

Why It Matters

Join configuration determines how unmatched records are handled.

Different business use cases require different strategies.

For example:

Payroll reconciliation often uses ***Left Join****, while master record matching may use* ***Inner Join****.*

How to Use It

1. *Select the desired join strategy.*
2. *Confirm the selected primary key fields.*
3. *Review suffix behavior for duplicate column names.*
4. *Verify join configuration settings.*

Once configured, continue to the output configuration step.

STEP 5 ENRICHMENT

What This Step Does

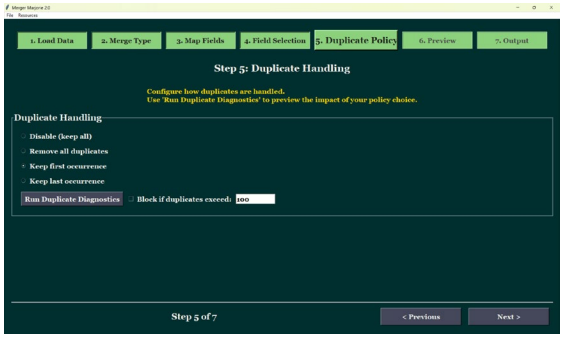
The Duplicate Handling step allows users to define how duplicate records should be managed in the merged dataset.

Duplicates occur when multiple rows contain the same values across one or more fields, typically involving a primary identifier such as an employee ID, customer ID, or transaction key.

Merger Marjorie provides configurable policies to control how duplicates are treated during processing.

Supported duplicate handling options include:

- ***Disable*** *– no duplicate filtering is applied*
- ***Keep First*** *– retains the first occurrence of each duplicate record*
- ***Keep Last*** *– retains the most recent occurrence of each duplicate record*
- ***Remove All*** *– removes all rows that contain duplicate values*



Duplicate detection can be applied to:

- *all fields in the dataset*
- *specific fields selected by the user*

Why It Matters

Duplicate records can lead to inaccurate results in reporting, reconciliation, and analysis. Configuring duplicate handling ensures that the final dataset reflects the intended record structure and prevents inaccurate calculations or reporting errors.

How to Use It

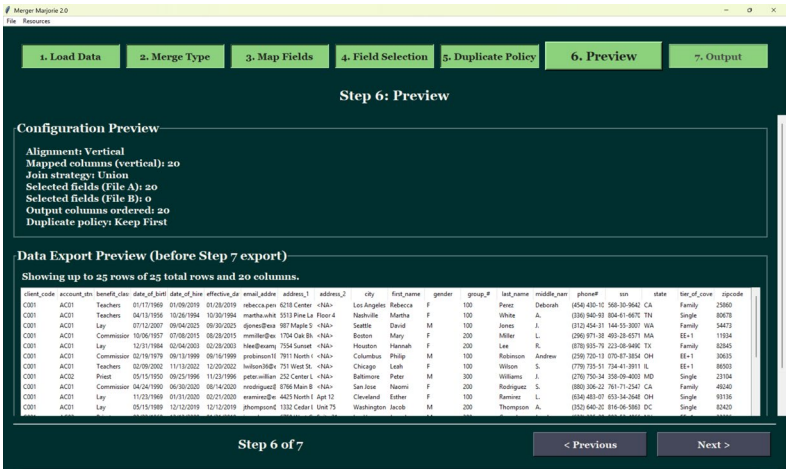
1. *Review the merged dataset to identify whether duplicate records exist.*
2. *Select a duplicate handling policy appropriate for the dataset.*
3. *If needed, specify the fields that should be used to detect duplicates.*
4. *Confirm the duplicate handling configuration.*
5. *Proceed to preview the output dataset before executing the merge.*

Once configured, Merger Marjorie will apply the selected duplicate policy during the merge process and produce a cleaned dataset according to the defined rules.

A small analytical insight: duplicates aren't always errors—they're often ***signals about the structure of the data****. Sometimes they represent multiple dependents, multiple transactions, or historical records. The important thing is having a tool that lets you* ***decide intentionally how they should be handled****, rather than letting spreadsheet formulas make that decision accidentally.*

STEP 6 — PREVIEW RESULTS

What This Step Does



This step displays a preview of the merged dataset before execution.

Users can inspect:

- *merged rows*
- *column alignment*
- *join behavior*
- *unmatched records*

Why It Matters

Previewing the dataset allows users to verify that the merge behaves as expected.

This step acts as a validation checkpoint before generating the final dataset.

It reduces the risk of producing incorrect output due to mapping or configuration mistakes.

How to Use It

1. *Review the dataset preview window.*
2. *Confirm column structure and data alignment.*
3. *Verify that key fields merged correctly.*
4. *Check for unexpected duplicates or missing values.*

If adjustments are needed, return to earlier steps and modify the configuration.

STEP 7 — EXECUTE THE MERGE

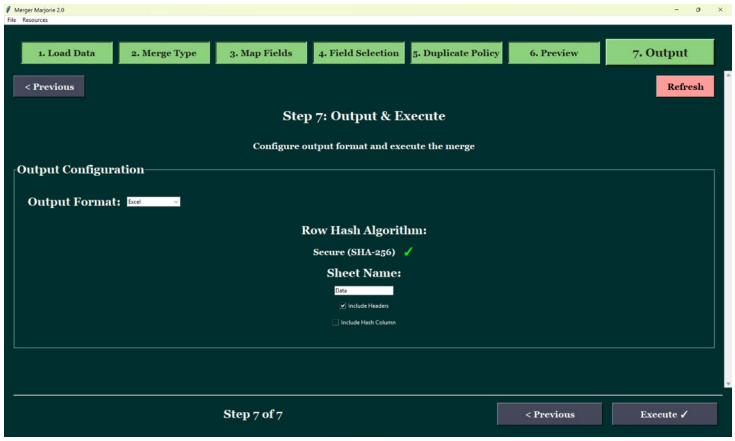
What This Step Does

This step runs the full merge pipeline and generates the final dataset.

During execution the application performs:

- *dataset alignment*
- *type validation and coercion*
- *duplicate handling*
- *data enrichment*
- *output generation*

The final dataset is exported to the selected output format.



Why It Matters

Execution produces the final analysis-ready dataset while ensuring that all validation and transformation rules have been applied.

The system also generates an audit-friendly output including optional row hashing for data provenance.

How to Use It

1. *Click **Execute Merge**.*
2. *The application processes the merge pipeline.*
3. *Once completed, the output file will be saved to the designated location.*
4. *Open the exported dataset to review the final results.*

AUDIT SUPPORT AND OUTPUT

Merger Marjorie is designed to support auditability and transparency in data integration workflows. The application produces structured output datasets while maintaining information that helps users understand how the merged data was generated.

During each merge execution, the system tracks key processing details including dataset structure, merge configuration, and processing timestamps. These records help ensure that merge operations can be reviewed, validated, and reproduced if necessary.

Traceable Merge Configuration

Each merge operation is performed using user-defined configuration settings, including:

- *selected datasets*
- *primary key mappings*
- *field mappings*
- *join strategy*
- *duplicate handling settings*

Capturing these parameters ensures that the logic used to generate the output dataset can be clearly understood and repeated if the merge needs to be run again.

Execution Metadata

Merger Marjorie records execution metadata that may include:

- *dataset schema signatures*
- *record counts for input and output datasets*
- *merge type and join configuration*
- *processing timestamps*
- *application version used for the merge*

This metadata provides a structured summary of the merge process and can support internal reviews, documentation requirements, or audit inquiries.

Structured Output Dataset

The final merged dataset is exported in a clean, structured format suitable for further use in:

- *data analysis*
- *financial reconciliation*
- *compliance reporting*
- *operational reporting*
- *downstream system imports*

Output files are generated locally and remain within the organization's controlled environment.

Reproducible Data Integration

Because merge configurations and validation checks are applied consistently, users can reproduce the same merge process when working with updated datasets or recurring reports.

This repeatability supports stronger data governance practices and reduces the risks commonly associated with manual spreadsheet merging.

CHOOSING THE CORRECT JOIN STRATEGY

Selecting the correct join strategy is one of the most important decisions when merging datasets. The join strategy determines how records from two datasets will be aligned and how unmatched records are handled.

*Merger Marjorie supports both **horizontal joins (key-based matching)** and **vertical joins (dataset stacking)**.*

Join Strategy Decision Guide

Scenario	***Recommended Strategy***	***Why***
*You are combining two datasets containing **different attributes for the same records***	*Horizontal Join*	*Aligns records based on a shared primary key*
*You are combining **two lists with identical structure***	*Vertical Join (Append)*	*Stacks records into one dataset*
*You want **only records that exist in both datasets***	*Inner Join*	*Keeps matched records only*
*You want **all records from dataset A and matching data from dataset B***	*Left Join*	*Maintains the full primary dataset*
*You want **all records from both datasets***	*Outer Join*	*Retains all records whether matched or not*
*You want **only columns shared between datasets***	*Vertical Union*	*Ensures structural consistency*

HEALTH INSURANCE CENSUS CREATION

Scenario

Human Resources and benefits administrators often need to build a **health insurance census file** for insurance carriers, brokers, or actuarial analysis.

This census typically combines information from multiple sources, including:

- payroll system exports
- employee eligibility lists
- medical plan enrollment files
- dependent enrollment records

Manually assembling a census often requires significant spreadsheet work and increases the risk of data inconsistencies.

Merger Marjorie allows these datasets to be aligned and merged quickly using a shared employee identifier.

Objective

Create a consolidated **health insurance census dataset** that includes:

Field	Description
Employee ID or SSN	Unique identifier for the employee
Employee Name	Full legal name
Gender	Employee gender
Date of Birth	Used for actuarial rating
Residential ZIP Code	Used for geographic rating
Date of Hire	Eligibility verification
Medical Plan	Plan selected by the employee

Field	Description
Medical Plan Tier	Coverage level (EE, EE+Spouse, EE+Child, Family)
Dependents	Dependents associated with the employee

Source Datasets

Dataset A — Eligible Employees (Payroll Export)

This dataset is typically exported from the payroll or HRIS system and includes employees eligible for benefits.

Example structure:

| Employee ID | Name | Gender | DOB | ZIP | Hire Date |

This dataset defines **who is eligible for coverage**.

Dataset B — Medical Plan Enrollment

This dataset contains employees who have elected coverage.

Example structure:

| Employee ID | Medical Plan | Plan Tier |

This dataset defines **who is enrolled and which plan they selected**.

Dataset C — Dependent Enrollment

This dataset contains dependent information associated with the employee.

Example structure:

| Employee ID | Dependent Name | Dependent DOB | Relationship |

This dataset identifies **covered dependents tied to the employee record**.

Merge Strategy

The census is created through **two horizontal joins**.

Step 1 — Employee Eligibility + Medical Enrollment

Join Dataset A and Dataset B using:

Employee ID or SSN

Join Type:

Left Join

Why Left Join?

This ensures all **eligible employees remain in the dataset**, even if they did not enroll in coverage.

Result:

| Employee ID | Name | Gender | DOB | ZIP | Hire Date | Medical Plan | Plan Tier |

Step 2 — Add Dependent Records

Next, merge the dependent enrollment dataset.

Join Dataset C using:

Employee ID

Join Type:

Left Join

Result:

| Employee ID | Name | Gender | DOB | ZIP | Hire Date | Medical Plan | Plan Tier | Dependent Name | Dependent DOB | Relationship |

Each dependent record will appear as a row associated with the employee identifier.

Resulting Census Dataset

The final census contains all required fields for most insurance carriers.

Example output:

Employee ID	Name	Gender	DOB	ZIP	Hire Date	Medical Plan	Plan Tier	Dependent Name
12345	Jane Smith	F	1985-03-21	50312	2019-04-02	PPO	Family	John Smith
12345	Jane Smith	F	1985-03-21	50312	2019-04-02	PPO	Family	Emily Smith
56789	David Lee	M	1990-07-10	60614	2021-06-15	HSA	EE	—

Business Outcome

Using Merger Marjorie to build the census provides several operational benefits:

Faster Census Creation

The census can be generated in minutes rather than hours of manual spreadsheet work.

Improved Data Accuracy

Primary key matching ensures dependent records align with the correct employee.

Complete Eligibility Visibility

Left joins allow HR teams to see:

- eligible employees not enrolled in coverage
- incomplete enrollment records
- missing dependent information

Repeatable Census Generation

The same merge configuration can be reused for:

- annual renewals
- quarterly reporting
- carrier requests
- compliance audits

Best Practices for Census Preparation

For the most accurate results:

- Ensure employee identifiers are consistent across datasets
- Verify date formats are standardized
- Confirm dependent records contain the correct employee identifier
- Remove duplicate employee records before merging

Merger Marjorie's validation and alignment engine helps identify and correct many of these issues automatically during ingestion and processing.

Small nerdy insight: insurance carriers love census files because they're the **raw input for actuarial risk models**. A messy census doesn't just create admin headaches—it can actually skew premium calculations. Tools like Merger Marjorie quietly prevent that kind of chaos.

ACA 1095-C REPORTING DATASET

Scenario

Applicable Large Employers (ALEs) must report employee health coverage information to the IRS each year using Forms **1094-C and 1095-C** under the Affordable Care Act (ACA).

This reporting requires employers to track, for each employee and each month of the year:

- employment status
- full-time eligibility
- offer of coverage
- coverage enrollment
- employee contribution amounts

These data elements typically reside in multiple systems including payroll, HRIS, and benefits enrollment platforms.

Merger Marjorie enables organizations to consolidate these datasets into a single structured reporting dataset suitable for ACA compliance reporting.

Objective

Create a consolidated ACA reporting dataset that includes:

Field	Description
Employee ID or SSN	Unique employee identifier
Employee Name	Employee legal name
Date of Birth	Required for reporting verification
Date of Hire	Used to determine eligibility period
Employment Status	Active / terminated
Full-Time Status	Determined from hours worked

Field	Description
Offer of Coverage Indicator	Whether coverage was offered
Medical Plan	Plan offered or selected
Employee Contribution	Required employee premium share
Coverage Start Date	When coverage began
Coverage End Date	When coverage ended

This dataset forms the basis for generating **IRS Forms 1095-C**.

Source Datasets

Dataset A — Payroll Hours and Employee Status

This dataset is typically exported from the payroll system and includes employee hours worked.

Example structure:

| Employee ID | Pay Period | Hours Worked | Employment Status |

This dataset is used to determine **full-time status under ACA rules**.

Under ACA guidelines, employees averaging **30 or more hours per week (130 hours per month)** are considered full-time.

Dataset B — Employee Master Record (HRIS)

This dataset contains core employee demographic information.

Example structure:

| Employee ID | Name | DOB | Hire Date |

This dataset ensures all employees are properly identified and reported.

Dataset C — Benefits Enrollment

This dataset contains the employee's health plan election information.

Example structure:

| Employee ID | Medical Plan | Coverage Start | Coverage End | Employee Contribution |

This dataset identifies whether coverage was offered and whether the employee enrolled.

Merge Strategy

The ACA reporting dataset is created using **two horizontal joins**.

Step 1 — Combine Payroll and Employee Master Data

Join Dataset A and Dataset B using:

Employee ID

Join Type:

Left Join

Why Left Join?

The payroll dataset typically represents the most complete list of employees who worked during the reporting period.

Result:

| Employee ID | Pay Period | Hours Worked | Status | Name | DOB | Hire Date |

This dataset now contains both **hours worked and employee demographic information**.

Step 2 — Add Benefits Enrollment Data

Join Dataset C using:

Employee ID

Join Type:

Left Join

Why Left Join?

This allows employers to identify:

- employees who were eligible but did not enroll
- employees offered coverage but declined
- employees who enrolled in coverage

Result:

| Employee ID | Name | Hours Worked | Hire Date | Medical Plan | Coverage Start | Coverage End | Contribution |

Resulting ACA Reporting Dataset

The final dataset contains all required information for ACA reporting preparation.

Example output:

Employee ID	Name	Hours Worked	Full-Time Status	Medical Plan	Coverage Start	Coverage End
12345	Jane Smith	168	Full Time	PPO	01-01-2025	—
67890	Michael Lee	95	Part Time	—	—	—
54321	Carlos Ramirez	145	Full Time	HSA	03-01-2025	—

This dataset can then be used to generate **IRS Form 1095-C reporting records**.

Business Outcome

Using Merger Marjorie for ACA reporting preparation provides several operational advantages.

Simplified Data Consolidation

Multiple datasets from payroll, HR, and benefits systems can be aligned quickly.

Improved Compliance Accuracy

Primary key matching ensures coverage data is correctly associated with each employee.

Full Eligibility Visibility

Employers can identify:

- full-time employees without coverage offers
- coverage start and termination inconsistencies
- employees missing required reporting fields

Faster Annual Reporting Preparation

The consolidated dataset can be generated in minutes rather than hours of manual spreadsheet manipulation.

Best Practices for ACA Reporting Preparation

For best results when preparing ACA reporting datasets:

- Ensure employee identifiers are consistent across all source systems
- Verify payroll hours include all paid hours for the reporting period
- Confirm coverage start and end dates match carrier records
- Validate employee status (active vs terminated)

Merger Marjorie's structured merge workflow helps identify data inconsistencies before reporting datasets are finalized.

Here’s a small strategic thought from the analytics side of the universe: ACA reporting datasets are essentially **compliance-grade data pipelines disguised as spreadsheets**. When HR teams build them manually, they’re doing ETL work without realizing it. Merger Marjorie turns that hidden ETL process into a controlled pipeline, which is exactly what auditors like to see.

GLOSSARY OF TERMS

The following glossary defines common terms used throughout the Merger Marjorie user guide and within the application workflow.

Append (Vertical Join)

A method of combining two datasets by **stacking rows from one dataset beneath another**. This is used when both datasets have similar structures and represent the same type of record.

Audit Trail

A record of the operations performed during a merge process, including configuration settings, input datasets, timestamps, and output results. Audit trails support compliance, traceability, and reproducibility.

Merger Marjorie tracks execution metadata such as schema signatures, merge configuration, and processing timestamps.

Column (Field)

A column represents a **specific data attribute** within a dataset.

Examples include:

- Employee ID
- Date of Birth
- Department
- Salary

Columns contain related values for each row in the dataset.

Dataset

A structured collection of rows and columns representing tabular data.

Examples include:

- payroll exports
- HR employee rosters
- financial transaction reports
- benefits enrollment lists

Within Merger Marjorie, datasets are loaded from CSV or Excel files and validated before merging.

Duplicate Record

A record that appears more than once within a dataset based on the same identifier or set of fields.

Duplicate records can cause unexpected merge results and may need to be removed or consolidated prior to merging.

Field Mapping

The process of aligning columns from one dataset with corresponding columns in another dataset.

Field mapping determines:

- which columns will appear in the output
- how columns from each dataset correspond
- which field acts as the primary key

Horizontal Join

A merge operation that combines two datasets **side-by-side** by matching records using a shared key field.

Inner Join

A type of horizontal join that keeps **only records that exist in both datasets**.

Records without matching keys are excluded from the final dataset.

Left Join

A type of horizontal join that retains **all records from the primary dataset (Dataset A)** while adding matching information from Dataset B.

Records without matches appear with blank values for Dataset B fields.

Outer Join

A join strategy that retains **all records from both datasets**, including unmatched records.

Unmatched fields will appear as blank values in the output dataset.

Primary Key

A field that uniquely identifies each record in a dataset.

Examples include:

- Employee ID
- Customer ID
- Account Number

Primary keys are used to match records between datasets during horizontal joins.

Row (Record)

A row represents a **single record or observation** within a dataset. Each row corresponds to one entity, such as an employee, customer, or transaction.

Schema

The structural definition of a dataset, including:

- column names
- data types
- column order

Schemas help ensure compatibility when merging datasets.

Merger Marjorie validates dataset schemas before executing a merge operation.

Union

A vertical join strategy that combines rows from two datasets **while retaining only shared columns**.

This ensures structural consistency when datasets contain different columns.

Validation

The process of checking datasets for structural and logical correctness before merging.

Examples include:

- verifying primary keys exist
- checking data types
- confirming required columns are present

Validation helps prevent errors during the merge process.

Vertical Join

A merge operation that combines datasets **by stacking rows** rather than aligning columns.

Vertical joins are typically used when datasets contain similar fields and represent the same type of records.

Output Dataset

The final merged dataset produced by Merger Marjorie after executing the merge workflow.

This dataset can be exported for:

- reporting
- analytics
- compliance reporting
- reconciliation

A small philosophical aside from the land of data engineering: most data problems aren't computational—they're **semantic**. People think they're disagreeing about numbers, but they're actually disagreeing about definitions. A glossary quietly solves that before the first merge ever runs.

TECHNICAL REQUIREMENTS

Merger Marjorie is a Windows-based desktop application designed to run within an organization's local computing environment. The software is lightweight and does not require specialized infrastructure, servers, or cloud connectivity.

System Requirements

Merger Marjorie operates on standard business workstations running Microsoft Windows.

Operating System

- Windows 10
- Windows 11

Both 64-bit systems are supported.

Hardware Requirements

Merger Marjorie is designed to run on typical office workstations.

Minimum recommended specifications:

- **Processor:** Dual-core CPU or higher
- **Memory (RAM):** 8 GB recommended
- **Storage:** 500 MB available disk space
- **Display:** 1280 × 720 resolution or higher

Larger datasets may benefit from additional memory and processing capacity.

Software Dependencies

Merger Marjorie relies on a lightweight Python runtime environment and several widely used open-source libraries for structured data processing.

Core runtime dependencies include:

• pandas
• openpyxl
• xlrd

These libraries enable the application to read, validate, and process structured spreadsheet data.

requirements

All required dependencies are packaged with the application during installation.

Users do not need to install or configure these components manually.

Supported File Formats

Merger Marjorie supports the most common structured data formats used by business systems.

Supported input formats:

• Microsoft Excel (.xlsx)
• Microsoft Excel (.xls)
• CSV (Comma-Separated Values)

Output datasets can be generated in Excel or CSV format.

Network Requirements

Merger Marjorie does **not require an internet connection** to perform merge operations.

The application runs entirely within the user’s local environment and processes datasets directly on the workstation.

This design supports organizations that require strict control over sensitive data such as payroll records, employee information, or financial transactions.

Data Size Considerations

Merger Marjorie is optimized for structured spreadsheet datasets commonly used in business reporting.

Typical dataset sizes include:

- thousands to tens of thousands of rows
- dozens of columns per dataset

Performance will depend on dataset size and workstation resources. For very large datasets, additional system memory may improve processing speed.

SOC APPLICABILITY STATEMENT

Overview

Merger Marjorie is a locally installed desktop application designed to perform structured data merging within a user's internal computing environment. The software operates entirely on the user's workstation and does not rely on external servers, hosted infrastructure, or cloud-based services to perform its functionality.

Because Merger Marjorie is not delivered as a hosted service and does not process or store customer data within infrastructure operated by the software provider, **SOC reporting frameworks are generally not applicable to the product architecture.**

Deployment Model

Merger Marjorie is deployed as a **standalone Windows desktop application** installed directly on a user's workstation.

All data processing occurs locally within the organization's controlled environment.

The application:

- does not transmit data to external servers
- does not store customer data in vendor-managed infrastructure
- does not provide multi-tenant or hosted services
- does not require internet connectivity to perform merge operations

All input files, processing operations, and output datasets remain within the organization's internal environment.

SOC Reporting Considerations

SOC 1 and SOC 2 reports are designed primarily for **service organizations that host, process, or manage customer data within vendor-operated systems or cloud infrastructure**.

Since Merger Marjorie:

- is not a Software-as-a-Service (SaaS) platform
- does not operate vendor-managed infrastructure
- does not store or process customer data outside the customer's environment

there is **no service organization control environment that would fall within the scope of a SOC examination.**

As a result, **SOC 1 and SOC 2 reports are not applicable for the Merger Marjorie software product.**

Customer Data Responsibility

Organizations using Merger Marjorie maintain full control and responsibility over their own data environment, including:

- data storage locations
- workstation security controls
- internal access permissions
- data governance policies

Because all data remains within the customer's infrastructure, security and compliance controls are governed by the organization's internal IT and data protection policies.

Summary

Merger Marjorie operates as a local desktop application that performs data processing entirely within the user's environment. The software does not host, store, or transmit customer data through vendor-managed systems.

Due to this architecture, **SOC reporting frameworks such as SOC 1 and SOC 2 are not applicable to the Merger Marjorie product.**

MERGER MARJORIE — END USER LICENSE AGREEMENT (EULA)

Merger Marjorie Version: 2.0.26

Date: 2026-03-02

EULA Version: 1.0.0

Effective Date: The date you accept this Agreement by clicking “I Accept” (or equivalent) or by installing, accessing, or using the Software.

IMPORTANT: Please read this End User License Agreement (“Agreement”) carefully. This Agreement is a binding legal contract between **KnollEdge Consulting and Solutions LLC** (“KnollEdge,” “we,” “us,” or “our”) and the individual and/or entity accepting this Agreement (“Customer,” “you,” or “your”).

BY CLICKING “I ACCEPT,” OR BY INSTALLING, ACCESSING, OR USING THE SOFTWARE, YOU ACKNOWLEDGE THAT YOU HAVE READ, UNDERSTAND, AND AGREE TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT INSTALL, ACCESS, OR USE THE SOFTWARE.

1.Definitions For purposes of this Agreement, the following definitions apply:

•“Software” means the Merger Marjorie application, including any updates, upgrades, patches, documentation, and any related materials provided by KnollEdge.

•“Authorized User” means your employees, contractors, or agents who are permitted by youto use the Software and who are bound by obligations at least as protective of KnollEdge as those in this Agreement.

•“Documentation” means user guides, manuals, release notes, and other written or electronic materials KnollEdge makes available for the Software.

•“Customer Data” means any data, files, spreadsheets, content, or information that you input, upload, import, or otherwise make available to the Software.

2.License Grant

Subject to your ongoing compliance with this Agreement and any applicable order form, statement of work, or subscription terms (if any), KnollEdge grants you a limited, non-exclusive, non-transferable,

non-sublicensable, revocable license to install and use the Software solely for your internal business purposes.

All rights not expressly granted to you are reserved by KnollEdge and its licensors.

3.Use Restrictions

You agree that you will not, and will not permit any third party to:

•Copy, modify, translate, or create derivative works of the Software or Documentation, except as expressly permitted by applicable law that cannot be waived by contract;

•Reverse engineer, decompile, disassemble, or otherwise attempt to derive the source code, underlying ideas, algorithms, structure, or organization of the Software;

•Rent, lease, lend, sell, redistribute, sublicense, assign, publish, or otherwise make the Software available to any third party (including as a service bureau or managed service) without KnollEdge's prior written consent;

•Remove, alter, or obscure any proprietary notices, labels, or marks on the Software or Documentation;

•Use the Software to develop, train, benchmark, or improve any competing product or service;

•Use the Software in violation of any applicable law, regulation, or third-party rights.

4.Customer Responsibilities

You are responsible for (a) selecting the Software to achieve your intended results, (b) the accuracy, quality, integrity, legality, reliability, and appropriateness of Customer Data, and (c) maintaining appropriate administrative, physical, and technical safeguards for your systems, accounts, and Customer Data.

You are solely responsible for verifying outputs, results, calculations, reconciliations, and reports produced by the Software before relying on them for any business, payroll, benefits, financial, compliance, or other purpose.

5.Customer Data; Privacy; Security

As between the parties, you retain all right, title, and interest in and to Customer Data. KnollEdge does not claim ownership of Customer Data.

Unless otherwise expressly agreed in writing, the Software is intended to operate locally on your systems. If you voluntarily provide Customer Data to KnollEdge for support, troubleshooting, or other

services, you authorize KnollEdge to use such Customer Data solely as necessary to provide those services.

You represent and warrant that you have all rights, permissions, and lawful bases necessary to provide Customer Data and to allow the processing contemplated by this Agreement.

6.Support; Updates

KnollEdge may, but is not obligated to, provide support, maintenance, or updates for the Software. Any support or maintenance that is provided may be subject to separate terms, service levels, or fees.

KnollEdge may modify, update, or discontinue the Software (in whole or in part) at any time, provided that this Agreement will continue to apply to any version of the Software you use.

7.Fees; Taxes

If the Software is provided for a fee, you agree to pay all applicable fees as set forth in an applicable order form, invoice, or statement of work. Fees are non-refundable except as expressly stated in writing by KnollEdge.

You are responsible for any sales, use, value-added, withholding, or similar taxes or levies that may apply, excluding taxes based on KnollEdge's net income.

8.Intellectual Property; Ownership

The Software and Documentation are licensed, not sold. KnollEdge (and its licensors, as applicable) owns and retains all right, title, and interest in and to the Software, Documentation, and all related intellectual property rights.

You may provide suggestions, feedback, or ideas regarding the Software ("Feedback"). You grant KnollEdge a perpetual, irrevocable, worldwide, royalty-free license to use, reproduce, modify, distribute, and otherwise exploit Feedback without restriction or obligation to you.

9.Confidentiality

Each party may receive or have access to the other party's confidential or proprietary information ("Confidential Information"). Each party agrees to protect the other's Confidential Information using at least the same degree of care it uses to protect its own similar information, and not less than reasonable care.

Confidential Information does not include information that (a) is or becomes publicly available through no breach of this Agreement, (b) was known to the receiving party prior to disclosure, (c) is received from a third party without breach of any obligation, or (d) is independently developed without use of the disclosing party's Confidential Information.

10.Term; Termination

This Agreement begins on the date you accept it and continues until terminated.

KnollEdge may terminate this Agreement immediately upon notice if you breach this Agreement. You may terminate this Agreement by ceasing all use of the Software and destroying all copies in your possession or control.

Upon termination, the license granted in Section 2 ends immediately and you must stop using the Software. Sections that by their nature should survive termination (including but not limited to ownership, confidentiality, warranty disclaimers, limitation of liability, indemnification, and governing law) will survive.

11.Disclaimer of Warranties

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE SOFTWARE AND DOCUMENTATION ARE PROVIDED “AS IS” AND “AS AVAILABLE,” WITH ALL FAULTS, AND WITHOUT WARRANTIES OF ANY KIND.

KNOLLEDGE DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, QUIET ENJOYMENT, ACCURACY, AND NON-

INFRINGEMENT, AND ANY WARRANTIES ARISING OUT OF COURSE OF DEALING OR USAGE OF TRADE.

KNOLLEDGE DOES NOT WARRANT THAT THE SOFTWARE WILL BE ERROR-FREE, UNINTERRUPTED, SECURE, OR THAT ANY RESULTS OR OUTPUTS WILL BE COMPLETE, ACCURATE, OR RELIABLE.

12.Limitation of Liability

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT WILL KNOLLEDGE (INCLUDING ITS OWNERS, OFFICERS, DIRECTORS, EMPLOYEES, CONTRACTORS, AFFILIATES, LICENSORS, OR SUPPLIERS) BE LIABLE FOR ANY INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, EXEMPLARY, OR PUNITIVE DAMAGES, OR FOR ANY LOSS OF PROFITS, REVENUE, DATA, GOODWILL, BUSINESS INTERRUPTION, OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, ARISING OUT OF OR RELATED TO THIS AGREEMENT OR THE SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND REGARDLESS OF THE THEORY OF LIABILITY (CONTRACT, TORT, STRICT LIABILITY, OR OTHERWISE).

Some jurisdictions do not allow certain limitations of liability; in such cases, the above limitations will apply to the maximum extent permitted by applicable law.

13.Hold Harmless; Indemnification (Customer Obligations)

To the maximum extent permitted by applicable law, you agree to RELEASE, DEFEND, INDEMNIFY, AND HOLD HARMLESS KnollEdge from and against any and all claims, demands, causes of action, suits, proceedings, damages, losses, liabilities, penalties, fines, costs, and expenses (including reasonable attorneys' fees) arising out of or related to:

•Your access to, use of, or inability to use the Software;

•Customer Data, including any allegation that Customer Data infringes or misappropriates third party's rights or violates any law;

•Your breach of this Agreement, negligence, willful misconduct, or violation of law;

•Any decisions, actions, or omissions you take based on outputs or results generated by the Software.

This hold harmless and indemnification obligation is intended to allocate risk so that KnollEdge has zero liability for claims arising from Customer's use of the Software, except to the limited extent such liability cannot be waived under applicable law.

14.Compliance with Laws; Export Controls

You agree to comply with all applicable laws, rules, and regulations in connection with your use of the Software, including privacy, employment, tax, benefits, and record retention requirements.

You may not use, export, re-export, or transfer the Software except as authorized by United States law and the laws of the jurisdiction in which the Software is obtained or used. You represent that you are not located in, under the control of, or a national or resident of any prohibited country and are not on any prohibited or restricted party list.

15.Third-Party Materials

The Software may include or interact with third-party software, services, libraries, or components ("Third-Party Materials"). Third-Party Materials may be subject to separate license terms. To the extent a third-party license grants you additional rights or imposes additional obligations, those terms will apply to the Third-Party Materials.

KnollEdge is not responsible for Third-Party Materials and disclaims all liability arising from Third-Party Materials to the maximum extent permitted by applicable law.

16.Governing Law; Venue; Jurisdiction

This Agreement and any dispute, claim, or controversy arising out of or relating to this Agreement or the Software will be governed by and construed in accordance with the laws of the State of Iowa, without regard to conflict of laws principles.

Exclusive venue and jurisdiction for any legal action or proceeding will lie in the state courts located in Crawford County, Iowa. If a federal court has subject matter jurisdiction, venue will lie in the applicable federal court for the district encompassing Crawford County, Iowa. Each party irrevocably submits to the personal jurisdiction of such courts and waives any objection based on inconvenient forum.

17.Notices

Notices under this Agreement must be in writing and will be deemed given when delivered personally, sent by certified mail (return receipt requested), or sent by reputable overnight courier, in each case to the addresses designated by the parties.

KnollEdge notice address:

KnollEdge Consulting and Solutions

1417 Broadway, Penthouse

Denison, Iowa 51442

staciknoll@knolledgehub.com

18.Assignment

You may not assign or transfer this Agreement or any rights or obligations under it without KnollEdge’s prior written consent. Any attempted assignment in violation of this section is void.

KnollEdge may assign this Agreement to an affiliate or in connection with a merger, acquisition, reorganization, or sale of all or substantially all of its assets.

19.Entire Agreement; Severability; Waiver

This Agreement constitutes the entire agreement between the parties regarding the Software and supersedes all prior or contemporaneous agreements, proposals, or communications, whether oral or written, relating to the Software.

If any provision of this Agreement is held unenforceable, that provision will be enforced to the maximum extent permissible and the remaining provisions will remain in full force and effect.

No waiver of any breach or default will be deemed a waiver of any subsequent breach or default.

20.Electronic Acceptance; Authority to Bind

If you are accepting this Agreement on behalf of an entity, you represent that you have the authority to bind that entity, and “you” and “your” refer to that entity.

You agree that your electronic acceptance (including clicking “I Accept”) constitutes your signature and forms a binding contract.

Acceptance Record

By accepting this Agreement, you affirm that you have read and agree to all terms above.

Electronic Acceptance Record with Date and Time Stamp at time of purchase.

THIRD-PARTY SOFTWARE AND DEPENDENCY NOTICES

This is provided for notice and attribution purposes and forms part of the End User License Agreement (the “Agreement”). The Application includes or may interface with certain third-party software components, including open source software (“Third-Party Components”). Third-Party Components are licensed to you under their respective license terms (“Third-Party Licenses”), not under the Agreement. In the event of a conflict between the Agreement and a Third-Party License with respect to a Third-Party Component, the Third-Party License will control for that component.

1.Third-Party Component Notices

The following table identifies Third-Party Components that are commonly used by the Application and/or its build tooling, along with their applicable license(s). This list may not be exhaustive, and additional transitive dependencies may be present. Where available, full license texts are provided by the respective projects’ repositories or package indexes. Component	License	Project / License Reference
Python (runtime interpreter)	Python Software Foundation License v2 (PSF-2.0)	https://docs.python.org/3/license.html
Pandas	BSD 3-Clause	https://pypi.org/project/pandas/
NumPy	BSD 3-Clause	https://pypi.org/project/numpy/
Openpyxl	MIT	https://pypi.org/project/openpyxl/
et-xmlfile (openpyxl dependency)	MIT	https://pypi.org/project/et-xmlfile/

XlsxWriter	BSD 2-Clause	https://xlsxwriter.readthedocs.io/license.html

2.No Third-Party Warranties

Third-Party Components are provided by their respective authors and contributors “AS IS” and “AS AVAILABLE,” without warranties of any kind. To the maximum extent permitted by applicable law, the authors and contributors of Third-Party Components disclaim all warranties and conditions, whether express, implied, or statutory, including without limitation warranties of merchantability, fitness for a particular purpose, title, and non-infringement.

3.Microsoft Excel and Outlook Integration Notice

The Application may integrate with Microsoft Excel and/or Microsoft Outlook through Microsoft Office automation interfaces (e.g., COM). Microsoft Excel and Microsoft Outlook are trademarks of Microsoft Corporation. Microsoft is not affiliated with, and does not sponsor, endorse, or warrant the Application. Use of Microsoft Office features requires that you have a properly licensed and installed version of Microsoft Office (including Excel and/or Outlook) and an appropriately configured user account. No Microsoft code is distributed as part of the Application unless expressly identified in Microsoft-provided redistributables subject to separate terms.

4.Obtaining Third-Party License Texts

You may obtain the complete corresponding license text for each Third-Party Component from the project’s repository or package index page referenced above. Where a Third-Party License requires preservation of notices, those notices are satisfied by inclusion of this Addendum and by retaining license and attribution information in the applicable Third-Party Component distributions.