The Rapid Inquiry Facility (RIF)
Version 4.0
Data Loader user guide

Authors (2017):

Parkes, B….

Small Area Health Statistics Unit (SAHSU)
MRC-PHE Centre for Environment and Health
Department of Epidemiology and Biostatistics
School of Public Health
Imperial College London
Medical Faculty Building
St Mary's Campus, Norfolk Place
LONDON W2 1PG

Website www.sahsu.org

Contents

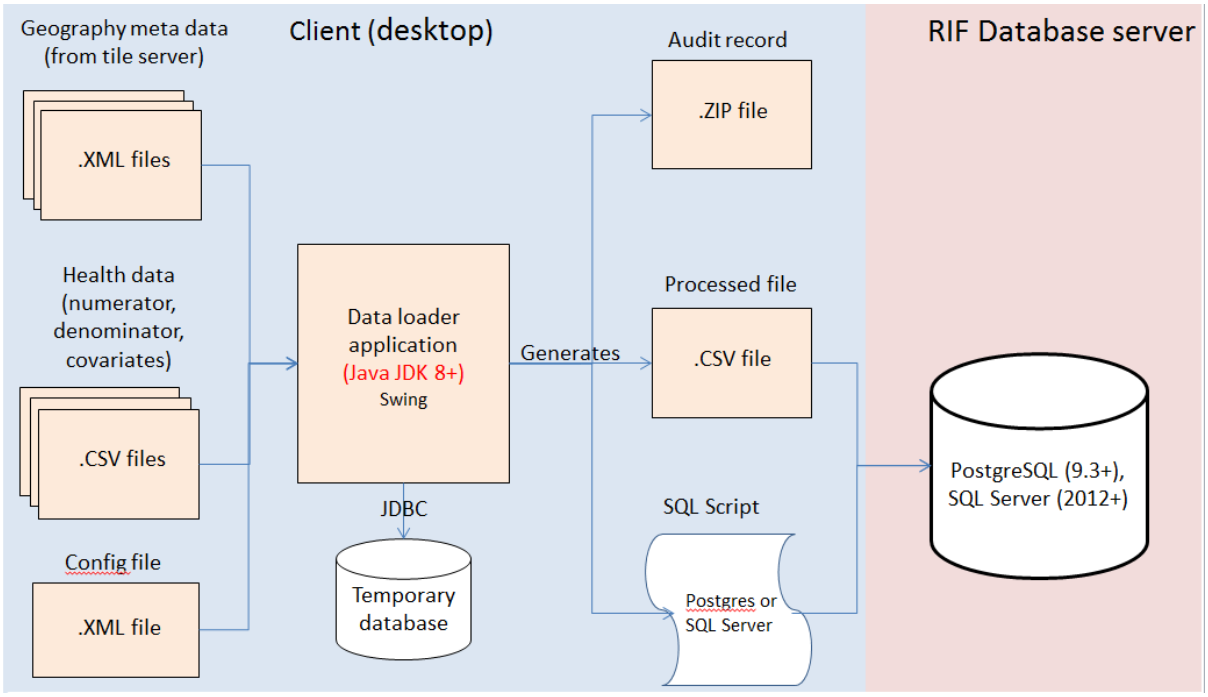
1. Introduction to the Data Loader

The Data Loader tool is a standalone utility to import health and population data from different formats. It is an example for an Extract Transform Load (ETL) tool that can massage different data sources into forms that are needed by the main RIF database. The data loader is used in parallel with the RIF Tile Maker, a separate utility for converting and importing map data into the RIF database.

1.1 Purpose

The data loader tool transforms imported data sets into cleaned files that can then be loaded into the RIF production database using simple scripts it makes for both PostgreSQL and SQL Server databases. The data loader tool itself uses a temporary database to provide a means of iterating through transformation steps that use temporary tables. Because the output of the data loader includes both a finished version of an imported data set and scripts that could load it into the RIF production database, the data loader's database can itself be viewed as a temporary artefact. Once data managers have processed all the files they want to load into the RIF database, they can elect to delete the data loader's temporary database.

1.2 Overview Schematic



This schematic gives an overview of the processes, data-files and components involved in the data loader process.

2. Requirements

The RIF Data Loader is a standalone application written in Java. It uses the Java Swing user interface library. The Data Loader is supplied as a JAR file (rifDataLoaderTool.jar) that contains all the necessary dependencies.

The data loader needs read/write/update access to a temporary database, either MS SQL server or Postgres

3. How to use

3.1 Configuring properties

The properties file for the data loader tool is named RIFDataLoaderToolStartupProperties.properties which resides in C:\GitHub\rapidInquiryFacility\rifDataLoaderTool\src\main\resources. The properties file contains various configuration options such as which database to use and the database login details. The following table lists and describes the properties that can be configured.

Table 1. RIFDataLoaderToolStartupProperties.properties.

Property	Valid values (separated by ;)	Description
databaseType	ms;pg	Specifies which temporary database type to use. ms = Microsoft SQL Server; pg = Postgres
databasePasswordFile	A valid path to an existing password file	Separate folders using double forward slashes. E.g. C://rif_scripts//db//RIFDatabaseProperties.txt
pg.driverClassName	org.postgresql.Driver	Only relevant when using Postgres db. Type of Postgres driver to use
pg.jdbcDriverPrefix	jdbc:postgresql	Only relevant when using Postgres db. Prefix of jdbc driver
pg.host	localhost; server name	Only relevant when using Postgres db. Name of server hosting Postgres database
pg.port	5432	Only relevant when using Postgres db. Port to use to connect to Postgres database
pg.databaseName	tmp_sahsu_db	Only relevant when using Postgres db. Name of temporary database to use.
ms.driverClassName	com.microsoft.sqlserver .jdbc.SQLServerDriver	Only relevant when using MS SQL Server. Type of SQL server driver to use
ms.jdbcDriverPrefix	jdbc:sqlserver	Only relevant when using MS SQL server. Prefix of jdbc driver
ms.host	localhost; server name	Only relevant when using MS SQL server. Name of server hosting database
ms.port	1433	Only relevant when using MS SQL server. Port to use to connect to database
ms.databaseName	tmp_sahsu_db	Only relevant when using MS SQL server. Name of temporary database to use.

The Data Loader read a file which contains the userid and password to access the temporary database used by the Data Loader, the location and name of the file is defined by databasePasswordFile set in the properties file described above. The contents of the file should contain a valid username and password to access the temporary database used by the dataloader. E.g:

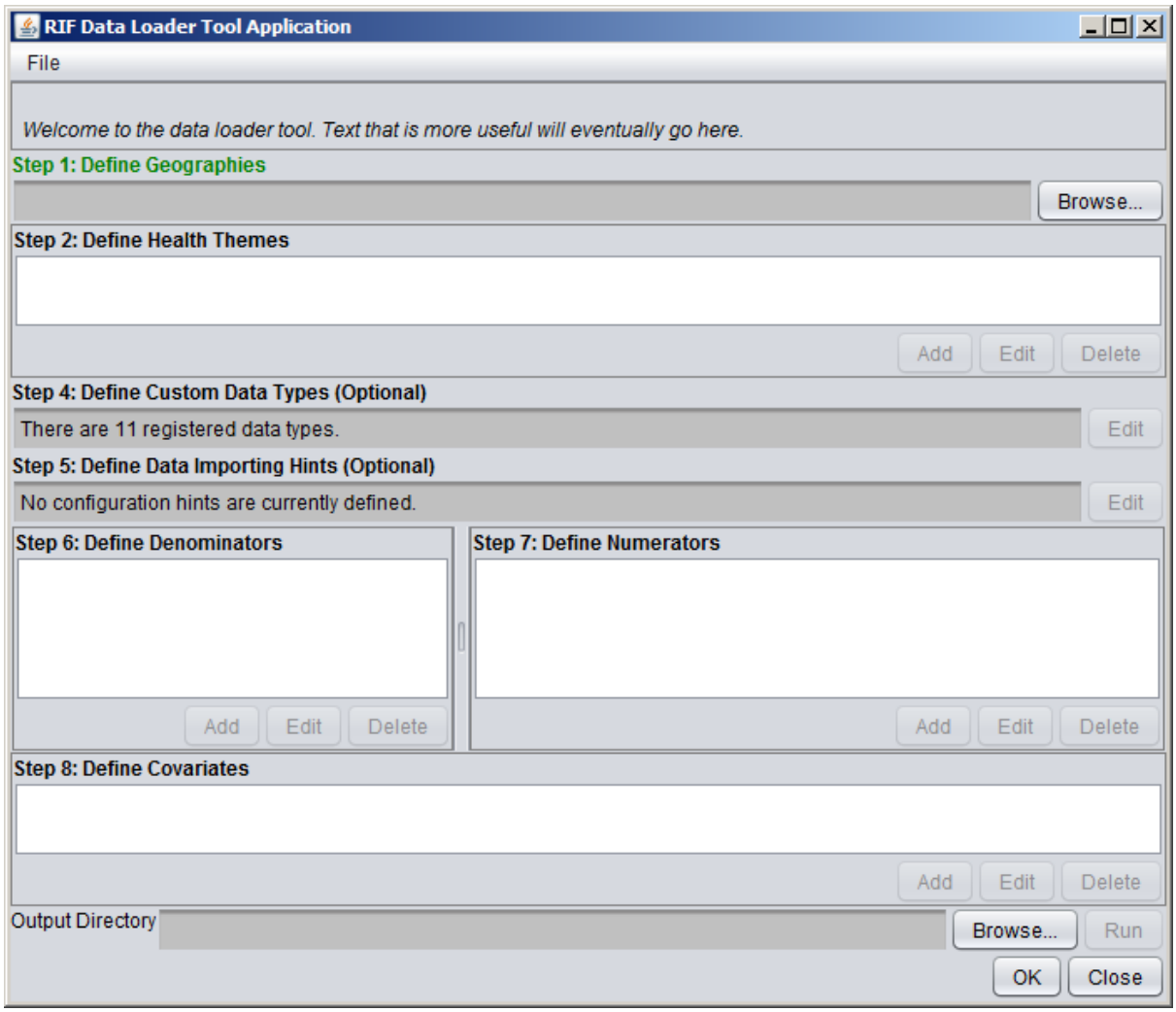
userID=postgres

password=XXXXXXX

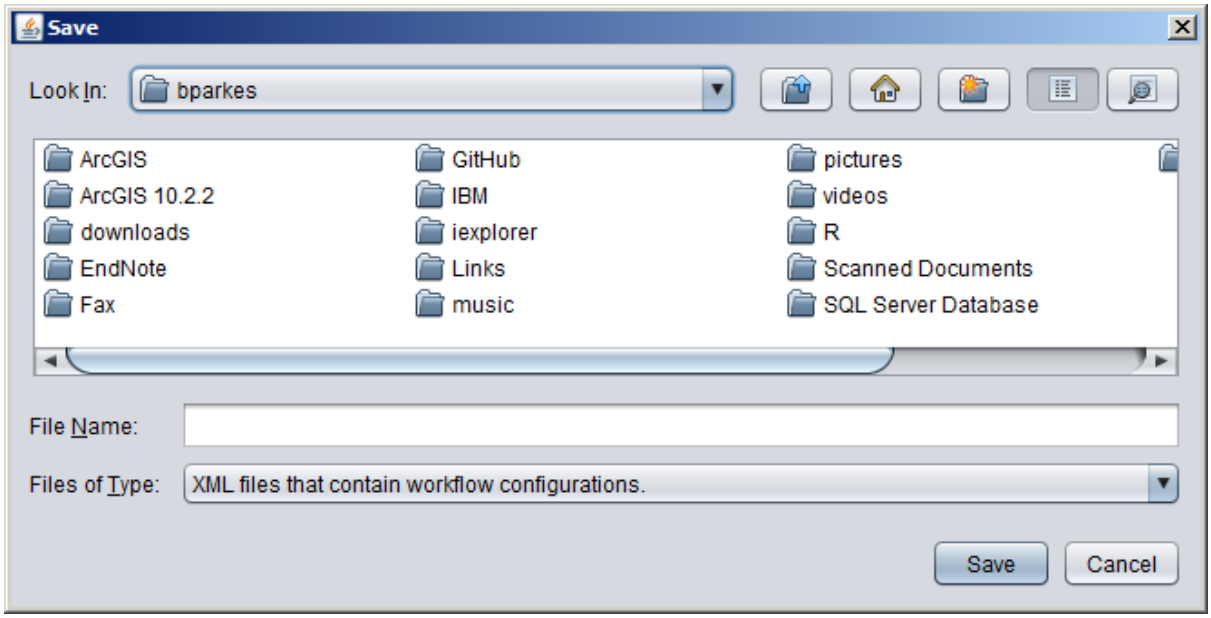
3.2 Starting up

When the data loader is started up, the following screen is loaded:

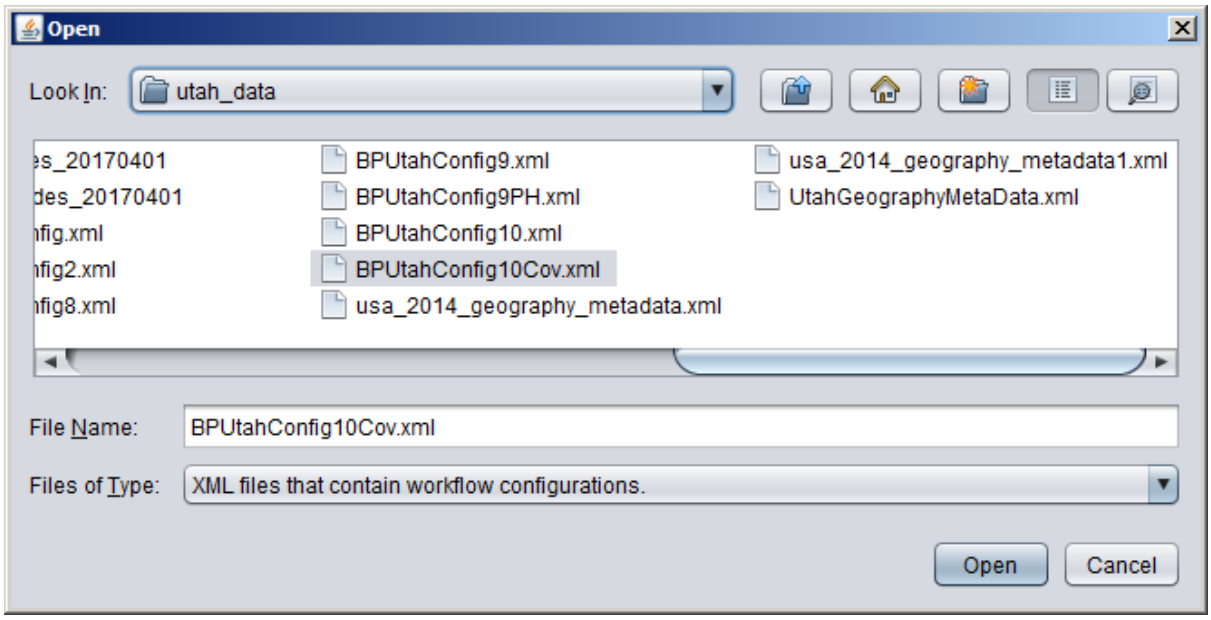
Figure 1. RIF data loader tool, main screen



At any point during a data loader session, all the configuration settings can be saved in an XML file by clicking File-Save As:



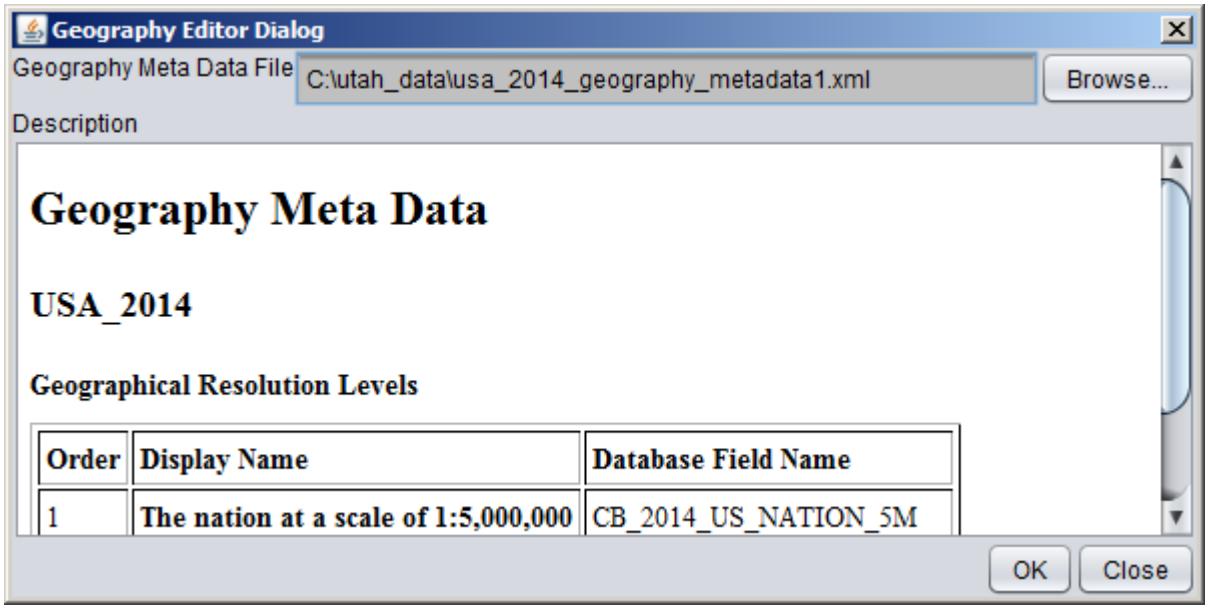
Previously defined configuration data can be loaded using File->Load… and opening an XML file:



The process of configuring the data loader is divided into 8 steps:

3.3. Step 1: Define Geographies

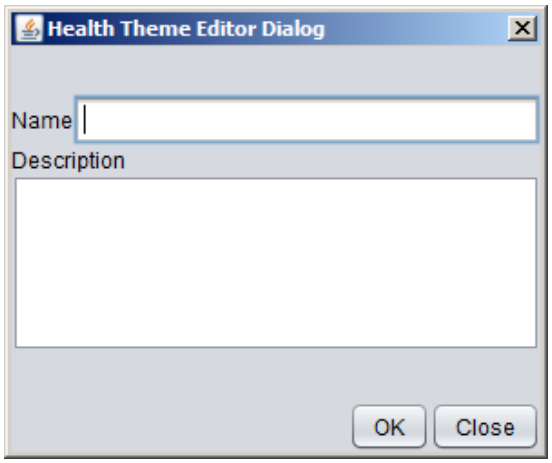
Click the 'Browse' button under the 'define geographies' area and select an appropriate XML file that defines the geographies used by the data:



Once an xml file is selected in the 'Geography Editor Dialog', a summary of the geographies is displayed. Press 'OK' if the geographies file is satisfactory.

3.4 Step 2: Define Health Themes

Once the geographies XML file has been selected, the 'Add' button is enabled in the 'Step 2' section. This brings up the 'Health Theme Editor Dialog':

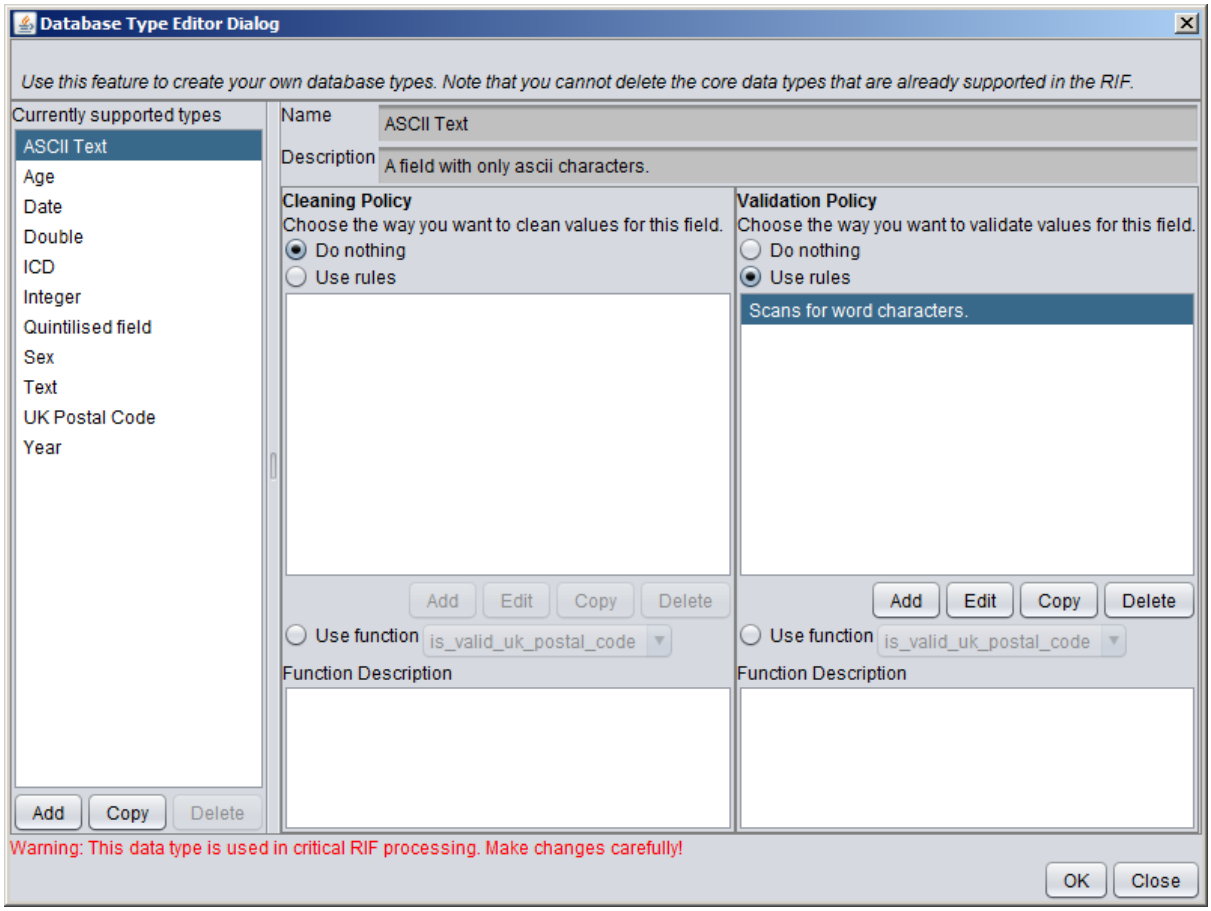


Here the use enters the name and description of the health theme being defined by the data being loaded.

3.5 Step 3: There is no step 3!

3.6 Step 4: Define Custom Data Types (Optional)

Once a health theme has been defined, the 'Edit' button is enabled in step 4 allowing the user to edit and add to the 11 pre-defined data types:

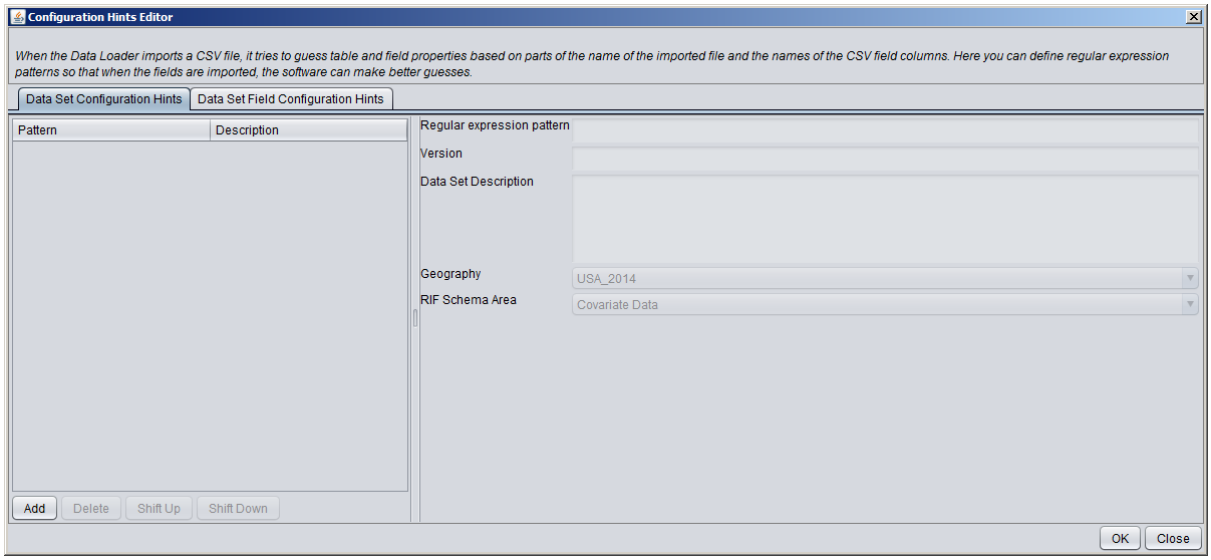


Only use this feature if you are confident in the changes required. New data fields may require new database functions to be written to either clean or validate the data for the new data type.

Examples of database types that might be created include: maternal age (which could have a minimum maximum plausible values defined); birth weight (minimum and maximum values).

3.7 Step 5: Define Data Importing Hints (Optional)

Once a health theme has been defined, the 'Edit' button is enabled in step 5 allowing the user defined configuration hints:



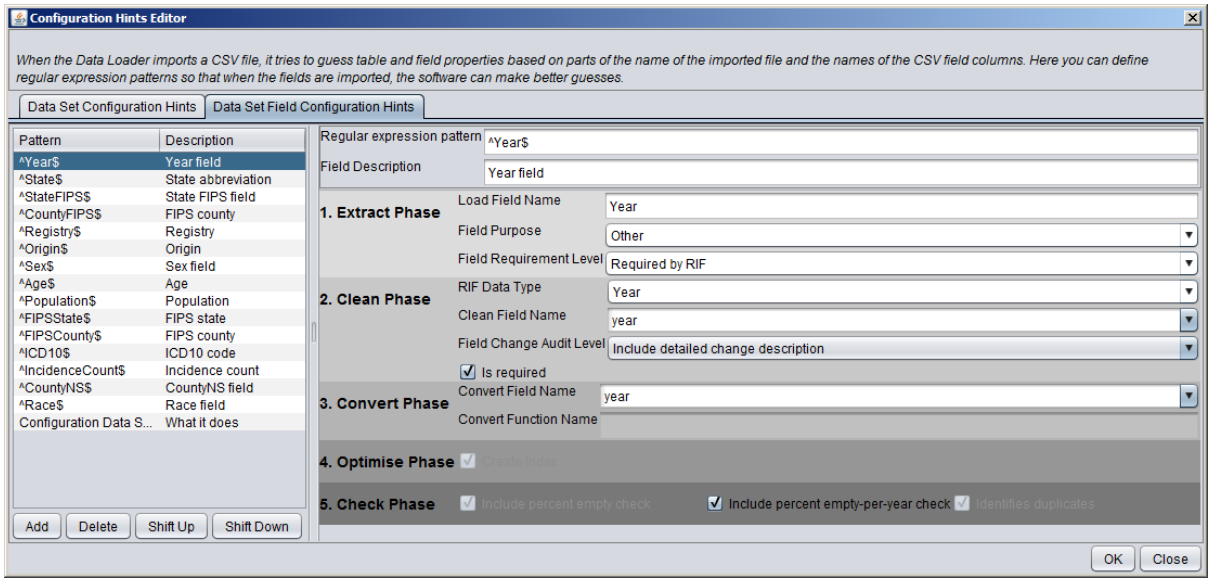
When the Data Loader imports a CSV file, it tries to guess table and field properties based on parts of the name of the imported file and the names of the CSV field columns. Here you can define regular expression patterns so that when the fields are imported, the software can make better guesses.

The more configuration options that the system supports, the more work this can cause the data manager. The data manager might have over 100 separate controls to adjust to configure the CSV file, consequently the data loader includes a hint feature which allows it set intelligent default values for fields.

The hints are based on naming conventions of data sets and fields. Data managers can associate regular expression patterns with default values of general data set properties. These include: version, a description and the target area of the RIF production database. Data managers can also define regular expression patterns to match CSV field names with field properties.

As an example of a data set hint, the regular expression ^cancer.* could be used to set the target area of the CSV file cancer_data_2012.csv so that it is set to “Health Numerator Data”. As an example of a field hint, .*year.* could be used to set the data type to RIF data type “Year” for any field name that contains ‘year’. A field hint of ^year$ would be more specific, meaning that the exact field name was year. In this case, we may decide that this is a required field for the RIF, whose numerator tables expect to have a field by that name.

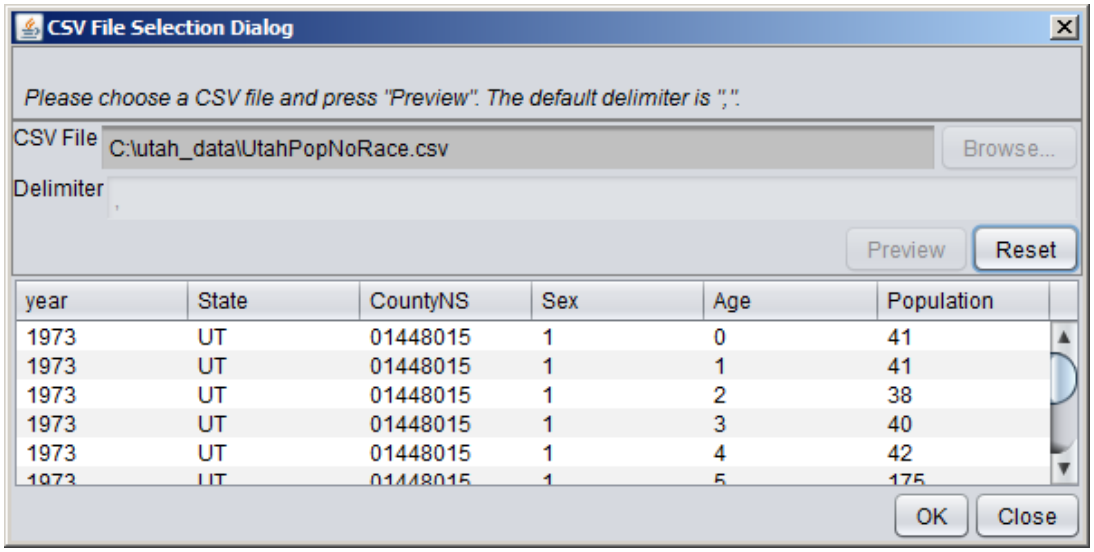
Field configuration hints are defined under the ‘Data Set Field Configuration Hints tab:



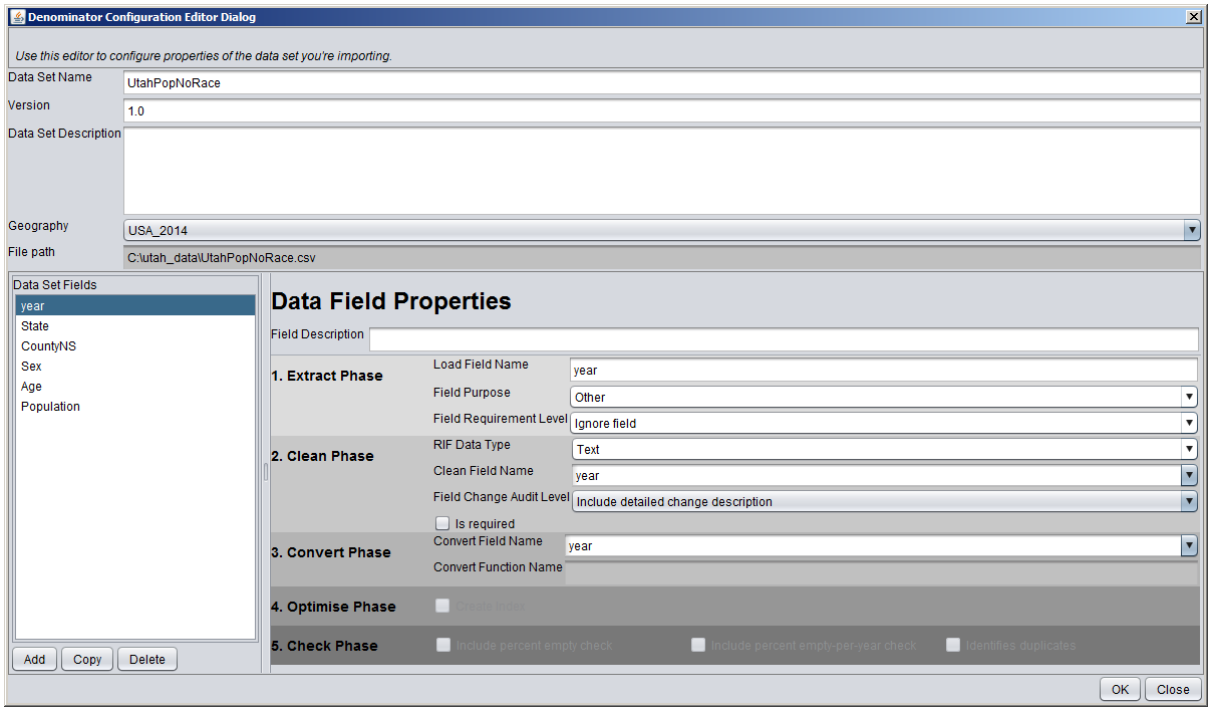
The field hint editor screen sets the default values for the actual data set fields. See section 3.10 to describe the phases and fields in this screen.

3.8 Step 6: Define Denominators

The 'Add' button in step 6 is enabled when the health theme has been defined (step 2). Clicking the add button brings up a file selection box prompting the user to select a CSV file that contains tabular denominator data. The delimiter is a comma by default, but other characters can be selected in the denominator field. Once a suitable denominator file has been selected, the 'preview' button must be clicked before the denominator file will be loaded:



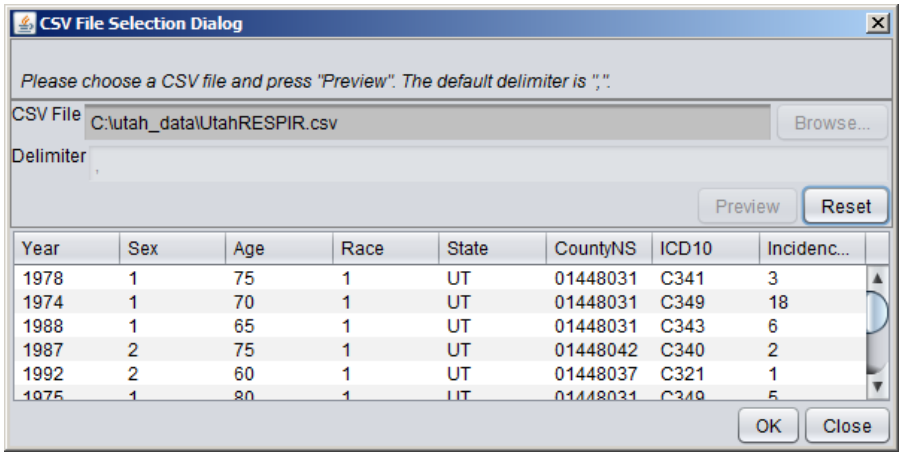
If the data previewed in the selection dialog looks satisfactory, the 'OK' button brings up the 'Denominator Configuration Editor Dialog':



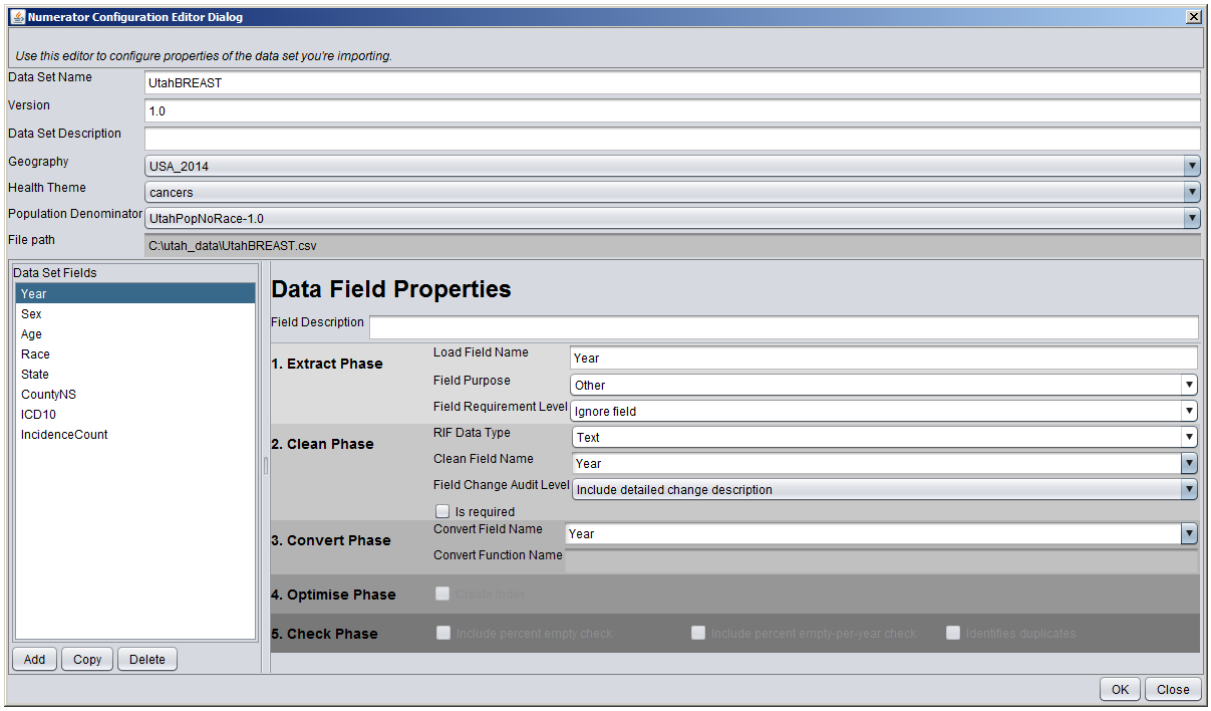
The dialog screen is used (with different titles) for editing all of the numerator, denominator and confounder data sets; its functionality and validations are described for all three purposes in section 3.10.

3.8 Step 7: Define Numerators

The 'Add' button in step 6 is enabled when the denominators have been successfully defined (step 6). Clicking the add button brings up a file selection box prompting the user to select a CSV file that contains tabular numerator data. The delimiter is a comma by default, but other characters can be selected in the delimiter field. Once a suitable numerator file has been selected, the 'preview' button must be clicked before the numerator file will be loaded:



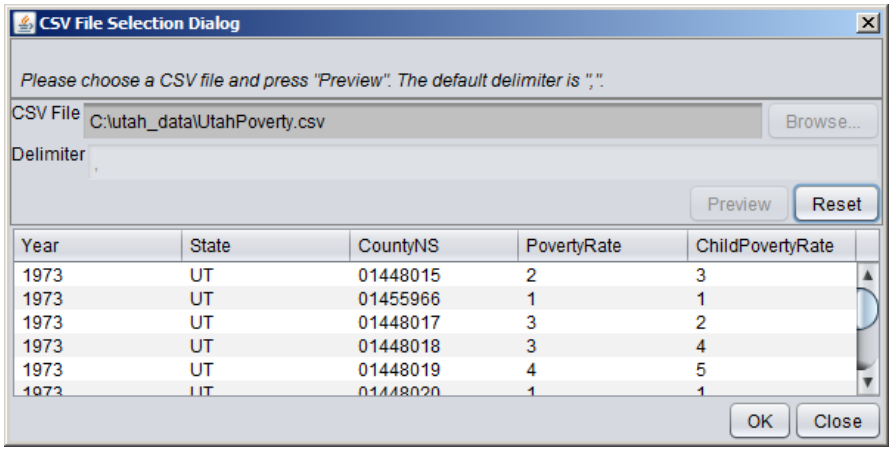
If the data previewed in the selection dialog looks satisfactory, the 'OK' button brings up the 'Numerator Configuration Editor Dialog':



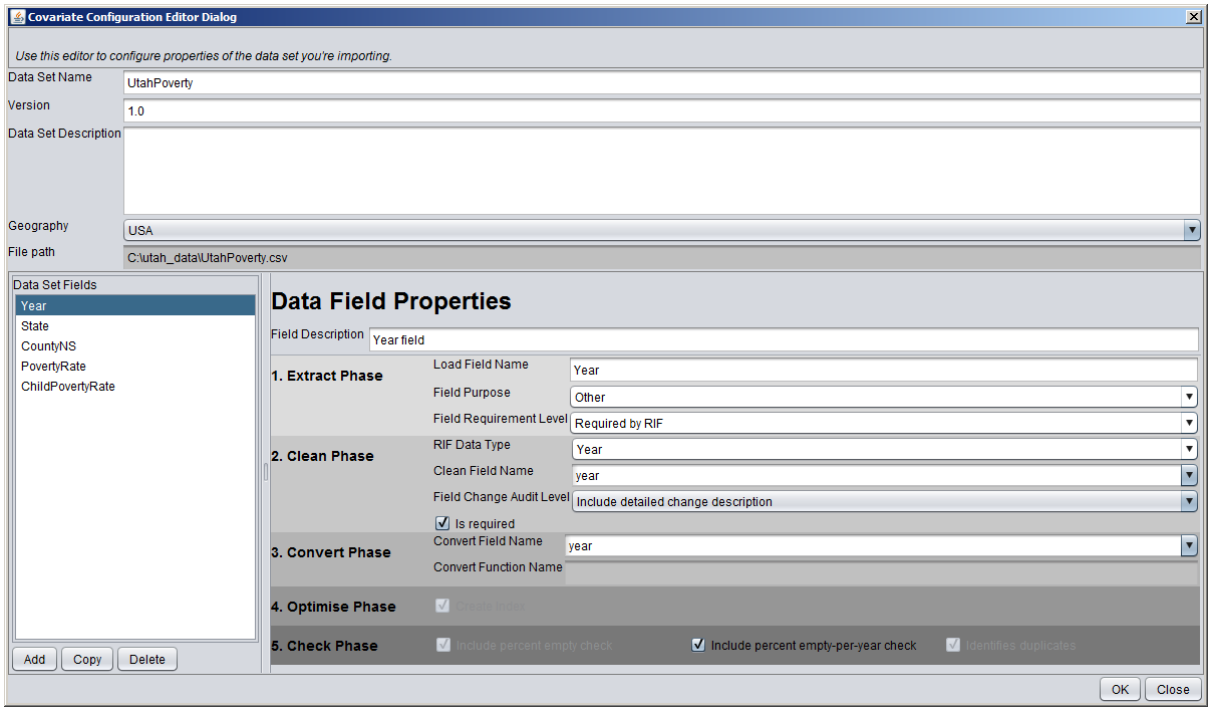
The dialog screen is used (with different titles) for editing all of the numerator, denominator and confounder data sets; its functionality and validations are described for all three purposes in section 3.10.

3.9 Step 8: Define Covariates

The 'Add' button in step 8 is enabled when the denominators have been successfully defined (step 6). Clicking the add button brings up a file selection box prompting the user to select a CSV file that contains tabular numerator data. The delimiter is a comma by default, but other characters can be selected in the delimiter field. Once a suitable covariates file has been selected, the 'preview' button must be clicked before the covariates file will be loaded:



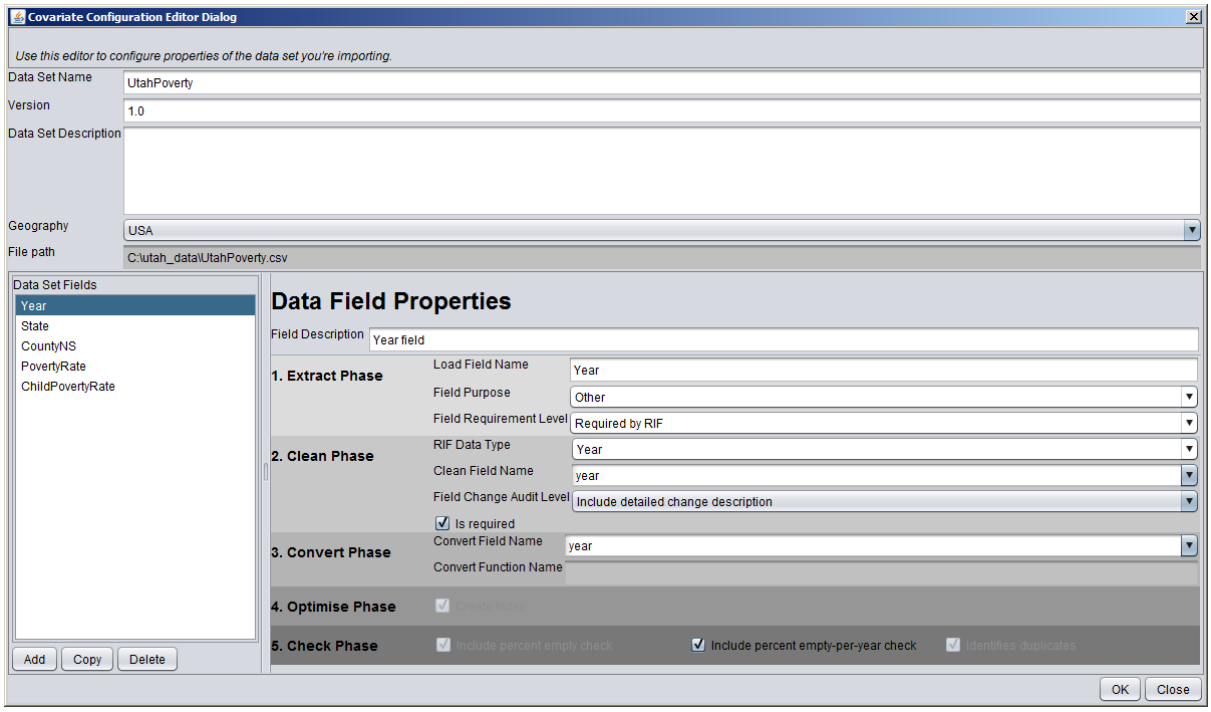
If the data previewed in the selection dialog looks satisfactory, the 'OK' button brings up the 'Covariate Configuration Editor Dialog':



The dialog screen is used (with different titles) for editing all of the numerator, denominator and confounder data sets; its functionality and validations are described for all three purposes in section 3.10.

3.10 Configuration editor dialog

The configuration edit dialog used for editing denominator, numerator and covariate data. The title of the screen and the validation rules change depending on which type pf data is being edited, other than that the functionality stays the same.



Data Set Name defaults to the data file name without the extension. The name data set name can be edited.

Version is used to keep track of the version number when several different file sets are employed.

Data Set Description is a descriptive field. Can contain details of where the data set came from etc.

Geography defines the link to the geography defined. Links to the geography meta data XML file.

The **Data Set Fields** list is populated with the fields defined in the csv file. The properties of each field must be defined carefully before the data loader will run. As well as the **field description** (which is free text), the properties are divided into 5 phases:

3.10.1 Data Field Properties. 1. Extract Phase

Load Field Name. The name of the field when it is first imported into the database. Defaults to the name of the header in the csv file or it can be auto-generated if no header data is available.

Field Purpose. Drop-down list defining the purpose of the field. For some destination areas of the RIF schema, processing the data set requires that a field satisfy a role. Selectable values are:

Field Purpose	Description	Validation rules
Other	Default setting. No special meaning in the data, will be loaded as is.	
Covariate	Used if this field is to be a covariate.	Covariate data requires at least one column that serves as a covariate.
Geographical resolution	Used when the column defines the geography of the record. Examples would be state, county, district, country	At least one field must be a Geographical Resolution for denominator, numerator and covariate data. The denominator and numerators screens will validate that the geographical resolution fields match those defined in the geographies file (step 1). It has to be an exact match.
Health code	Used for fields that define a health condition. For example a column containing the ICD-10 code would be of type 'health code'. Typically used for numerator data.	Numerator screen requires at least one field which is a health code.
Total count	Used for fields that contain the number of subjects. For example the population in denominator data, or the incidence count for numerator data.	The denominator and numerator screens will validate that there is exactly 1 field whose field purpose is 'Total Count'. 'Total count' type fields must have a RIF Data Type of 'Integer'

Field requirement level. Drop down list defining the level of validation required for this field. Selectable values are:

- Required by RIF – (default value) fields defined with this requirement level must have valid values to successfully be loaded using the data loader. This means that the **convert field name** must be one of a set of expected pre-defined values. Used for most denominator, numerator and covariate data.
- Extra field – will be loaded, but is not validated to contain valid values. . An extra field is one the RIF manager wants to promote in a data set but which is not required by the RIF.
- Ignore field – field is not to be loaded in the RIF and is ignored.

3.10.2 Data Field Properties. 2. Clean Phase

RIF Data Type. Drop-down list defining the data type this field should have in the RIF. Describes a set of predefined data types, each of which has a data type, a validation aspect and a cleaning aspect. Selectable values are:

RIF Data Type	Description	Database functions	Validation rules
Integer	Must contain valid integer values	is_valid_integer	Fields whose purpose is 'Total Count' must of RIF Data Type 'Integer'
Year	the year that the data applies to	clean_year	Must be exactly 1 field whose type is 'Year' for all three data types.
UK Postal Code	UK post code field	clean_uk_postal_code is_valid_uk_postal_code	
Sex	Sex field	convert_age_sex*	Must be exactly 1 field whose type is 'Sex' for denominator and numerator data.
Quintilised field	A field with 5 valid values		
Text	Free text field		
ASCII Text	ASCII text field		
ICD	ICD code for disease identification	clean_icd (or clean_icd_code)	
Double	Numeric field	is_numeric (or is_valid_double?)	
Age	Age.	clean_age; convert_age_sex*	Must be exactly 1 field whose type is 'Age' for denominator and numerator data.
Date	Date field	clean_date (or date matches_format?)	

* The convert_age_sex function combines the age and sex fields and groups the age into 5 year groups. The format of the age_sex field is: xyy where x defines the sex (1 = m, 2 = f, 3 = other), and

yy represents the age group: 00 to 04 are ages 0 to 4; 05 is age 5 to 9, 06 is age 10 to 14 etc. up to age 85 and above which is 21. So code 211 is female, aged 35 to 39.

Clean Field Name. The name of the field that the RIF manager would want to use to rename an imported field. Often it is the same as **Load Field Name** but it can be different if the load field names are auto-generated when data are first imported. Drop down field that can be edited (combo box?), behaviour and validation changes depending on the 'Field Purpose':

- If Field Purpose is 'Covariate', 'Health Code' or 'Other' the clean field name will default to the same as the '**Load Field Name**' (or the value set in the 'Data set Field configuration Hints' section if it exists) and can be edited.
- If field Purpose is geographical resolution, the clean field name drop-down will be populated with the geography display names defined in the geographies XML file, and the field cannot be edited. The field is validated such that clean field names for geographies exactly match the values in the XML file
- If field purpose is 'Total Count' the clean field name will default to 'total' (or the value set in the 'Data set Field configuration Hints' section if it exists) and can be edited.

Field Change Audit Level. Drop down that defines the level of auditing the data loader is to do when loading the data. Describes the extent to which changes in the field should be recorded in audit logs. Values are:

- 'None' - will not record any field changes at all
- 'Include field name only' - simply records that the field was changed
- 'Include detailed change description' - A detailed change description will include the field, original and revised values

The audit records are held in the temporary database (tables aud_chg_*; aud_val_*; cln_val_*) and also written as output for the dataloader session in the num_XXnameXX_date.zip file, 'audit_trail' folder, aud_chg_XXnameXX.csv and aud_val_XXnameXX.csv files.

Is required. Check box indicating if the data in this field can be empty. If the option is ticked then validation will detect errors when this field is empty.

3.10.3 Data Field Properties. 3. Convert Phase

Once the data sets have been cleaned, they need to be mapped to fields that are expected in part of the RIF schema. Sometimes this step involves combining fields.

Convert Field Name. Name of a field that is expected in the RIF. Drop-down box that can be edited. Will always default to the value in **Clean Field Name** but can be changed.

Convert Function Name. The name of the function used to map clean fields to fields in a converted version of the dataset. Will have the value 'convert_age_sex' for 'age' and 'sex' fields, blank for everything else.

3.10.3 Data Field Properties. 4. Optimise Phase

Create Index. Check box indicating if an index will be created for this field. For fields that have a **Field Requirement Level** of 'Required by RIF', the create index is checked and cannot be edited. For

fields that have a **Field Requirement Level** of 'Extra field', the create index is enabled and can the user can decide to check it or leave it blank. For fields that have a **Field Requirement Level** of 'Ignore', the create index is disabled and unchecked.

3.10.3 Data Field Properties. 5. Check Phase

This is the stage when the user decides what data quality checks will be generated to supplement the finished data set.

The percentage of records, by year, which have an empty field anywhere in the record.

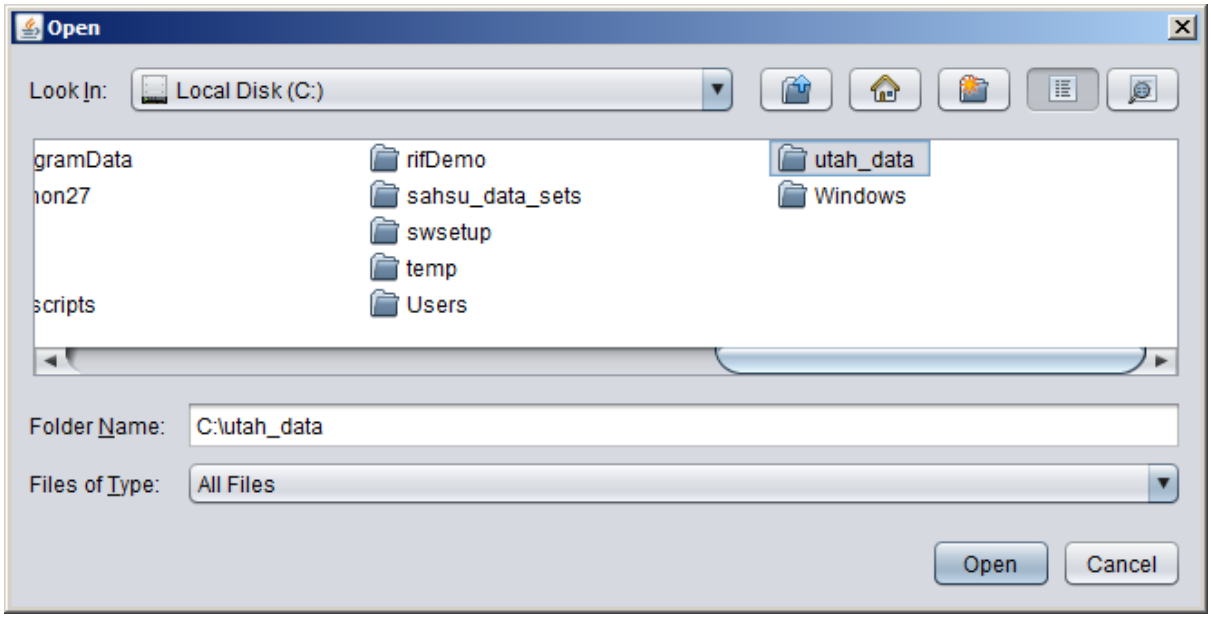
The behaviour and default values of the 3 check boxes in this section a defined in the following table:

Check box:	**Include percent empty check**	**Include percent empty-per-year check**	**Identifies duplicates**
Description:	If ticked, a data quality check will be produced field which calculates the percentage of values that are empty.	If ticked, a data quality check will be produced which calculates the percentage of values that are empty grouped by year. Only applies numerator and denominator tables.	Can the field be used to help identify duplicates in a collection of records? Sometimes multiple fields together are needed to identify whether two records are duplicates.
Enabled/disabled	Enabled when **Field Requirement Level** is set to 'extra field'.	Enabled when **Field Requirement Level** is set to 'extra field' or when editing covariate data properties	Enabled when **Field Requirement Level** is set to 'extra field'.
Default value	Checked when **Field Requirement Level** is set to 'extra field'. Unchecked when **Field Requirement Level** is set to 'ignore field'.	Depends on setting in the 'configuration hints' section.	Checked when **Field Requirement Level** is set to 'extra field'. Unchecked when **Field Requirement Level** is set to 'ignore field'.

Once all the data field properties have been edited, clicking 'OK' will return the user to the main application screen.

3.11 Running a data load

When the denominator, numerator and covariate data properties have been set (i.e. step 8 is complete), the user needs to select an output directory by clicking '**Browse**' then navigating to a suitable directory using the 'open' dialog:



Once a suitable directory has been selected the user can finally click '**Run**' on the main application screen to start the data loader.

Depending on the amount of data to be loaded, the process may take several minutes to run. All the data is read into the temporary database, checked, converted, optimised then exported to processed csv files along with xml files describing the content of the data and database scripts (MS SQL and Postgres) to load the data into the RIF database. A zip files containing audit trail, original data, processing stages and results are also generated. All the output is generated in a new folder with the name format:

run_DD-Mmm-yyyy_hh_mm_ss

This folder contains two master database scripts:

ms_run_data_loader_DD-Mmm-yyyy_hh_mm_ss.sql

pg_run_data_loader_DD-Mmm-yyyy_hh_mm_ss.sql

For each numerator, denominator and covariate data set there are 4 files:

XXX_datasetname.csv (CSV file of cleaned data to be load)

XXX_datasetname.fmt (XML file describing the format of the CSV file)

run_XXX_datasetname.sql (Postgres SQL script)

XXX_datasetname_DD-Mmm-yyyy_hh_mm_ss.zip (audit file)

Where XXX is 'covar' for covariates, 'num' for numerators, 'pop' for denominators data sets.

3.12 Database functions for tmp_sahsu_db

Are defined in

rapidInquiryFacility\rifDataLoaderTool\src\main\resources\MSCreateRIFDataLoaderToolDatabase.sql

and

rapidInquiryFacility\rifDataLoaderTool\src\main\resources\PGCreateRIFDataLoaderToolDatabase.sql

for MS SQL and Postgres respectively.