Overview

There are five types of data that need to be imported/exported at the same time: SKUs, ingredients, product lines, formulas, and manufacturing lines. At the customer's request, manufacturing lines do not need to be imported, but the relation between SKUs and manufacturing lines still does. In addition, for interoperability with existing spreadsheets, the formulas and their ingredients are specified in the same file.

CSV files do a good job representing table data, and are extremely easy to parse. In addition, CSVs are among the most space-efficient formats, so long as the tables they are representing are themselves the minimal. As discussed above, **four** CSV (comma-separated values) files are specified, and their headers are defined below:

Filename Prefix	Header
skus	`SKU#`,`Name`,`Case UPC`,`Unit UPC`,`Unit size`,`Count per case`,`PL Name`,`Formula#`,`Formula factor`,`ML Shortnames`,`Rate`,`Comment`
ingredients	`Ingr#`,`Name`,`Vendor Info`,`Size`,`Cost`,`Comment`
product_lines	`Name`
formulas	`Formula#`,`Name`,`Ingr#`,`Quantity`,`Comment`

File format

1. The names of the files <u>must</u> be prefixed by an identifying string, as defined above. The filenames <u>must</u> end in ".csv". More precisely, the filenames <u>must</u> constitute matches for the following regular expressions conforming to <u>PCRE</u> standard:
   - SKUs: `^skus(\S)*\.csv$`
   - Ingredients: `^ingredients(\S)*\.csv$`
   - Product Lines: `^product_lines(\S)*\.csv$`
   - Formulas: `^formulas(\S)*\.csv$`
2. The CSV files <u>must</u> be properly delimited and escaped. <u>RFC4180</u> specifies the format of a CSV file. Note specifically:
   - Spaces (and other space-like characters) <u>must not</u> be ignored if they are part of a field.
   - Null values, if they are used, <u>must</u> be denoted by an empty string (i.e., two consecutive commas, or a comma followed by a `CRLF` character).
   - The double quote symbol (ASCII 034) must be properly quoted.
3. A valid header, as defined above, <u>must</u> be included as the first line in each file. The header is case-insensitive.
4. Data types, uniqueness, requiredness, autogenerability of attributes in the files <u>must</u> conform to specifications in the **Evolution requirements**. If an attribute is not specified, it should follow the requirements below:
   - All-string-typed attributes <u>must</u> be no more than 1000 characters.
   - All Integer-typed attributes <u>must</u> be within range of $[-2^{31}, 2^{31})$ (i.e., a 32-bit signed integer).
   - All floating-point numbers <u>must</u> be in decimal, and have a leading 0 (before the decimal place) if the value is less than 1.0.
   - The column `ML Shortnames` in `skus.csv` contains CSV cells. Each of the CSV cell items <u>must</u> be a string with no whitespace.
5. Each file <u>must</u> contain no more than 1,000,000 lines. The records <u>may</u> be in no particular order.
6. For **mixed-unit expressions**, e.g., ones for the "Size" attribute of an ingredient, the following processing algorithm <u>must</u> be used.
   - The numbers (whole or decimal) are separated from non-number (the unit part). More formally, the expressions <u>must</u> constitute full matches for the following <u>PCRE</u>-standard regular expression: `1(\d*\.?\d+)\s*(\D.*|)$`, where the number part will be taken from group 1 of the match, and the unit part from group 2.
   - The unit part is then sanitized as follows:
     i. Trimmed of whitespace on both ends
     ii. Dots (".") and whitespaces (" ") are replaced with empty string
     iii. Converted to lowercase
     iv. Any trailing 's' is removed

○ The unit part (after sanitization) is then interpreted by matching it exactly to the list below.

Unit	Accepted forms
Mass-based units	
Ounce	"oz", "ounce"
Pound	"lb", "pound"
Imperial ton	"ton"
Gram	"g", "gram"
Kilogram	"kg", "kilogram"
Volume-based units	
Fluid ounce	"floz", "fluidounce"
Pint	"pt", "pint"
Quart	"qt", "quart"
Gallon	"gal", "gallon"
Milliliter	"ml", "milliliter"
Liter	"l", "liter"
Count-based units	
Count	"ct", "count"

Upload Specifications

1. Only one file <u>may</u> be submitted in one import **session**. This file <u>may</u> be of any type.
2. All submitted **CSV file(s)** <u>may</u> be assumed to have MIME type `text/csv`, as stipulated in <u>RFC4180 §3</u>.
3. The following checks <u>will</u> be performed on each file individually. If any of the check fails, the file <u>will</u> be rejected as a whole, and an appropriate error message <u>will</u> be displayed for the administrator.
   - File validity: the file <u>must</u> be standards-compliant, and contain the required headers in the order specified. The matching of headers should be case-insensitive.
   - **Duplicate records**: the file <u>must not</u> contain **duplicate records** in itself. The file <u>may</u> contain **collision(s)** to **records** existing in the system. Refer to the Collision Resolution algorithm in the <u>next section</u> when a **collision** is detected.
   - Data validity: where required, inputs must conform to standards.
     - UPC numbers <u>must</u> conform to the UPC-A standard
     - Sanitized units in **mixed-unit expressions** <u>must</u> be one in the accepted list. Further, units used in `formulas.csv` <u>must</u> be logically convertible to the original unit used in the ingredients. Similarly, if a record constitutes a **collision** in `ingredients.csv`, the unit, if different than that of the existing record, <u>must</u> be logically convertible to the existing unit.
   - Empty attributes: all required attribute(s) <u>must</u> be supplied, except when both of the following are true:
     - It is marked as **autogeneratable** in the **Evolution requirements**;
     - The record whose required attribute is empty contains no **collision**.
4. **Referential integrity** <u>must</u> be preserved in the file.
   Note that the following relations exist among the files / corresponding entries in the system:
   - In `skus.csv`,
     - `PL Name` <u>must</u> be existing in the system.
     - `Formula#` <u>must</u> be existing in the system.
     - All **CSV cell items** in `ML Shortnames` must be existing in the system.
   - In `formulas.csv`:
     - `Ingr#` <u>must</u> be existing in the system.

Example formulas.csv:

Formula#,Name,Ingr#,Quantity,Comment

1001,Formula1,2001,0.1250000000,

1001,Formula1,2002,0.2500000000,

1002,Formula2,2001,0.1250000000,

1002,Formula2,2006,0.0005000000,

Example ingredients.csv:

2001,Bread,Butternut,340g,2.29,This bread is used to make all the burger

2002,Beef,Beef Choice,2lb,12.25,

2006,Turkey,,424 pounds,666.66,

Example product_lines.csv:

Hamburger

Salad

Pasta

Example skus.csv:

SKU#,Name,Case UPC,Unit UPC,Unit size,Count per case,PL Name,Formula#,Formula factor,ML Shortnames,Rate,Mfg setup cost,Mfg run cost,Comment

1001,Burger Beef,042272010264,725272730706,600g,50,Hamburger,1001,1,"line1,,,,",1,$1,$2,

1002,Burger Turkey,987654321098,705632085943,250g,80,Hamburger,1001,1,"line1",1,$1,$2,