Gemini

Summarize this document

Katie Selasco

10:09 AM Jan 8

Let's consider adding a paragraph on how custom units are handled in the bulk loader

11
10
9
8
7
6
5
4
3
2
1
1
2
3
4
5
6
7
8
9
10

General Guidelines

The Scope 5 bulkloader can be used to load the following objects to a user's Scope 5 account:

- Activity Records
- Trackers
- Resources
- Conversion Factor Sets
- Tracking Nodes

Users can load these objects using either an MS Excel file or a CSV file created in any text editor.

Creating Objects vs. Updating Objects

The bulkloader always creates new objects. It cannot be used to edit or update existing records. Users may find the various bulk edit options available in the standard UX useful for this purpose.

Working With Excel Files

If using MS Excel files, users can start by downloading the template on this page. There is one template for each of the objects listed above. The template for each object includes a minimal set of standard columns, the headers for these columns and preconfigured drop down controls for many of them that can be used to conveniently populate a subset of the values to be loaded.

In some cases, the pre-configured template will be sufficient to load the user's objects. In other cases, the user may need to build on the templates by adding columns of their own.

The application will look for suffixes of *.xls* or *.xlsx* to identify Excel files to be loaded.

Working With CSV Files

Users choosing to work with CSV files will need to create files from scratch and include all the required column headers. The application will look for the suffix *.csv* to identify CSV files to be loaded.

General File Structure

In the next few paragraphs, we'll describe the general structure that the application expects for either type of bulkloader file. Subsequently, we'll describe specific requirements for each of the loadable objects.

Required Elements Common to Both File Types

The application will look for certain common elements to properly interpret and load the contents of either type of bulkloader file. Users should bear this in mind regardless of the file type they are loading. Some of these elements will already be in place when using the preconfigured Excel template. The user will need to create all elements explicitly when using CSV files.

The Primary Header Row

The application will start at the top row of any file and will scan down the file row by row to identify a required *primary header row*. In the case of Excel formatted bulkloader files made from the template, the application allows for rows containing certain explanatory text preceding the preconfigured primary header row.

If the user chooses to create CSV bulkloader files, the primary header row is not required to be the *first* row in the file but it must be the last row before any loadable content rows. Users may use the first rows in the file for titles or explanatory text. The application will interpret the presence of certain expected keywords to identify the primary header row so, if it is not the first row in the file, users are cautioned not to inadvertently use keywords in the introductory rows.

Note that the application expects to find the primary header row in the first 50 rows of any bulkloader file. If it does not, it will reject the file with an error message.

Keywords Identifying the Primary Header Row

As it scans each row, the application will look for an identifying keyword starting with the leftmost column. If it does not find a keyword identifying the primary header row in the first ten columns, it will advance to the next row. Again, users are not *required* to start CSV file content in the first column but for simplicity's sake, we recommend that they do.

The keyword that the application looks for is a *parent* object indicating the objects intended to be loaded with the file. This is considered an *anchor* keyword in the sense that it 'orients' the application and defines how the remaining content in the file should be interpreted.

Table 1 lists each of the loadable objects and the name of their parent object that is the anchor keyword. When the application finds one of these keywords, it will interpret that keyword to indicate that the bulkloader file is being used to load one or more of the corresponding child objects from subsequent *content* rows.

Keyword (parent object)	**Object Loaded**
Tracker	Activity Records
Tracking Node	Trackers

Parent Node	Tracking Nodes
Resource	Conversion Factor Sets
Library	Resources

Table 1 - Keywords and Corresponding Objects

When using Excel files, users should be careful to leave any preconfigured column headers intact (except for the *variable* columns as discussed below). When creating CSV files, users should be sure to include one of the keywords in the leftmost (ideally) column of one of the first rows in the file and should be aware that the application will assume all subsequent rows to contain content for objects of the corresponding type.

Keyword Format

Keywords are case insensitive. In addition, the application will ignore whitespace around keywords. For these objects, users must include all words that make up the keyword, with zero or more blank spaces between them.

Columns

Once the application has identified the primary header row, it will seek to identify other column headers in the row. The first column header found that contains any of the exact keywords from Table 1 will be interpreted to indicate that the file is being used to load the corresponding object. Subsequent column headers are also keywords that indicate required or optional attributes on the objects to be loaded. The set of additional keywords expected varies from one object type to another. These keywords follow the same formatting rules as the anchor keyword.

Note that keywords must be specified in column headers exactly as they are documented here. No additional content is allowed in column headers.

Headers for Required Columns

Note that the Excel templates prepend an asterisk character to certain column headers to indicate that they are required columns. Users must include values in the corresponding column for each and every row after the primary header row. Otherwise, the object represented by that row will be considered insufficiently specified and will be rejected with an error message.

Variable Columns

Excel templates for each object are preconfigured with a set of *standard* columns for the object. These are columns, required or optional, that occur exactly once and for which there is a specific correct header keyword (or keywords). Examples of such columns are the *Tracker* column or the *Activity Date* column for the *Activity Record* object. Each activity record is loaded to one tracker[1] and has just one activity date.

By contrast, *variable* columns may be included any number of times and may have many different keywords identifying them. An example of variable columns is the *inputs* or *independent variables* on the *Activity Record* object. Many trackers are configured to expect activity records with a single input, such as *weight.* However, others may expect two or more inputs, such as *weight* and *cost* or *volume* and *heat content.*

The rules governing variable columns may vary from object to object. Continuing with the example of inputs on activity records, the rules dictate that the user may include columns for any number of inputs in an activity record bulkloader. At least one header is required that must contain a keyword corresponding to the *unit category* or *handle* for one of the inputs on the tracker to which the object will be loaded. A column specifying units may be included immediately following (to the right) the column that names the input. The header for that column must repeat the same keyword that names the metric, followed by the keyword *Unit* or *Units*. This pair of columns is considered a column *group* because they must always occur together in a specific order.

There are several such variable columns for the different objects. These will be discussed in detail in subsequent object specific sections of this document.

Column Order

Columns may be in any order so long as the following rules are met.

First Non-Blank Column

As the application scans each row from left to right, it will look for one of the keywords tabulated in Table 1. The keyword must be in the first (leftmost) non-blank column of the file. The first such cell will be used as the anchor cell.

Column Groups

For most of the objects, there are certain attributes that require a group of columns. In the example illustrated in Figure 1, the weight input requires two columns, one to quantify the metric and the other to indicate the units quantified. Generally, when grouped columns are required, they must be grouped together in a specific order. The column groups must be complete - in this example, two columns are required for the input (even though one of those columns may be left blank). Specific column groups and the number of columns making up the group will be described subsequently in the object specific section of these instructions.

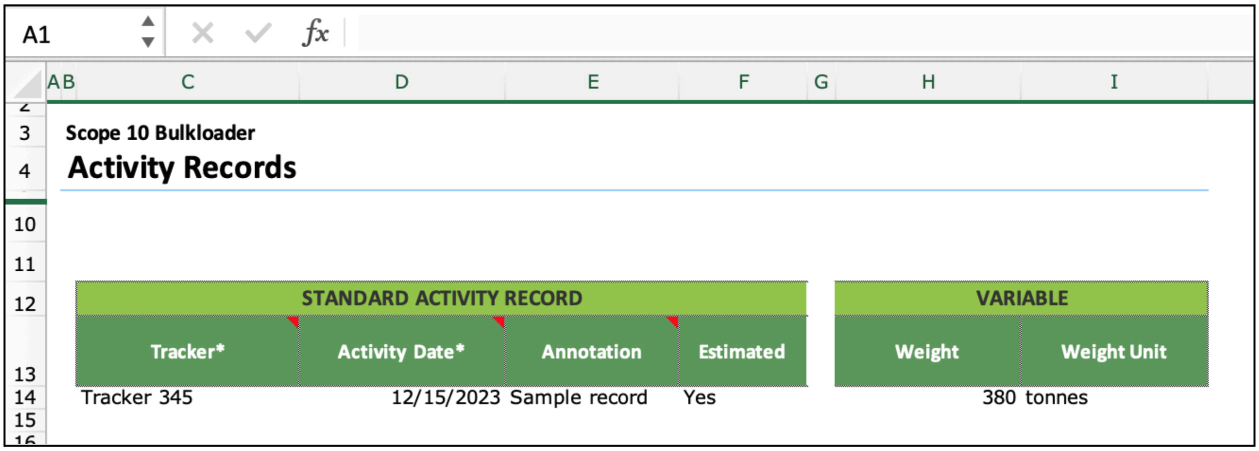
[1] In the case that multiple trackers share a ledger, loading an activity record to one tracker designated in the *Tracker* column will have the effect of making that record available to all trackers sharing the ledger but the record is still considered 'loaded' to a single tracker.

What's Included in the Templates?

We'll conclude this general introduction with a few notes on the downloadable Excel templates and CSV files. The downloadable templates generally include one or more header rows with the last one being the primary header row.

Users are expected to start entering content in the first row following the primary header row. The first column in which users are expected to enter content will be the column containing the *anchor* keyword. This column will be followed by each of the remaining *standard* columns for the corresponding object. Finally, each template will include a single sample column (or group of columns) for each type of *variable* column. In some cases, the template will be usable 'as is' but in most cases, the user will be expected to edit the variable columns to suit the objects that they are loading.

For an example of a template, see Figure 1, which illustrates the template for Activity Record objects. We'll continue to define general rules for bulkloader files for any of the objects using this figure as a reference point.



Scope 10 Bulkloader

Activity Records

STANDARD ACTIVITY RECORD				VARIABLE	
Tracker*	Activity Date*	Annotation	Estimated	Weight	Weight Unit
Tracker 345	12/15/2023	Sample record	Yes	380	tonnes

Figure 1 - Template for Activity Records

The anchor keyword, *Tracker*, is in cell C13. The anchor cell is preceded above by a number of introductory rows (row 1 - 12), and two empty columns, to its left. To the right of the anchor cell, we see the object's standard columns; D, E and F. Next, there is a single pair of variable columns for a *Weight* input and the units of that input (columns H and I respectively).

If the user happens to be loading activity records to trackers with a single input in the weight category, the template may be usable as is. However, it is likely that the user will need, at the very least, to edit the input column headers to correspond to a different unit category (such as *Volume* and *Volume Units*). In many cases, the user may also need to add a second pair of input columns, such as *Cost* and *Cost Units*.

The application assumes actual user generated content for the objects to be loaded to start with row 14, detailing in this case, a record for tracker 345.

For user generated CSV files, the row immediately below the primary header row will be assumed to contain data for the first object to be loaded.

Multiple Rows

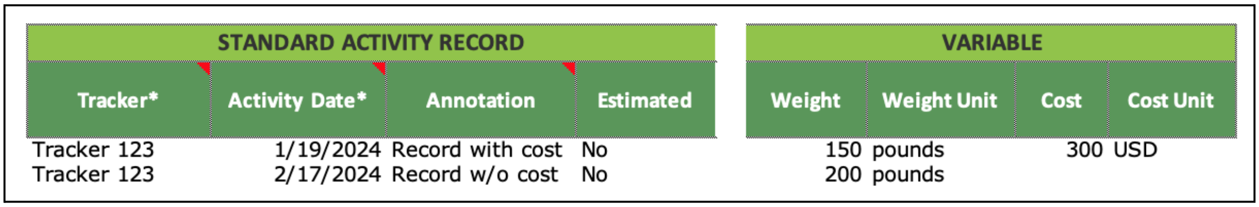
While it is possible to load hundreds and hundreds of rows (objects) in a single Excel or CSV file, we recommend that users start with a trial run with just a few content rows. In this manner, users can confirm that they have organized the data in the bulkloader correctly before proceeding. Once the user is confident that the headers and content are formatted correctly, they can proceed with as many objects as they wish. Keep in mind that there are performance limitations – more rows take longer to load and long bulkloaders can be unwieldy.

Blank Cells

There are several cases in which certain bulkloader cells might be left blank.

Objects WIth Optional Attributes

The first is the case in which not every one of a set of attributes is required for all instances of the same object type. Going back to the *activity record* objects from the previous example, consider a tracker that has two metric inputs - *weight* and *cost*. This is illustrated in Figure 2. The user started with the activity record template and added a pair of columns for an optional cost input. The user specified one activity record for *Tracker 123* which includes the cost metric and a second that does not. In this case, both cells corresponding to the cost metric are left blank for the second record.



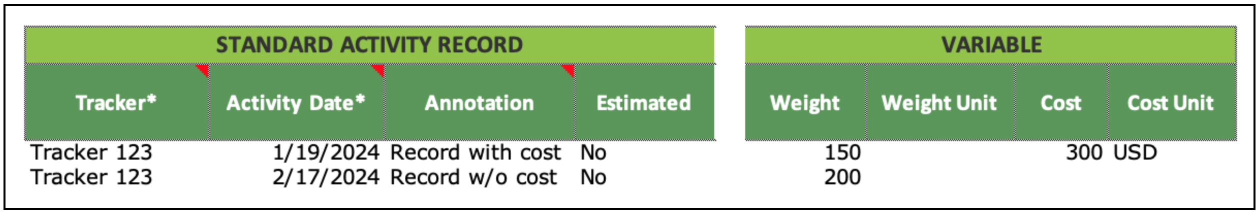
STANDARD ACTIVITY RECORD				VARIABLE			
Tracker*	Activity Date*	Annotation	Estimated	Weight	Weight Unit	Cost	Cost Unit
Tracker 123	1/19/2024	Record with cost	No	150	pounds	300	USD
Tracker 123	2/17/2024	Record w/o cost	No	200	pounds		

Figure 2 - Optional Attributes

If the cost was a required metric for the tracker, the user would see an error for the second record when attempting to load the sample file illustrated.

Default Values

The next case in which blank cells might be present is the case when one or more columns represent attributes that have a *default* value. Continuing with the now familiar example of activity records, recall that trackers are configured with *default units* for each of the tracker's metrics. For the tracker considered in Figure 3, the default *weight* units are *pounds*. The user can leave the units column blank for the weight input on each record that is quantified in the default units. If the user wants to load a record that quantifies the weight input in units other than pounds, the user would have to explicitly specify those units in the corresponding cell.



STANDARD ACTIVITY RECORD				VARIABLE			
Tracker*	Activity Date*	Annotation	Estimated	Weight	Weight Unit	Cost	Cost Unit
Tracker 123	1/19/2024	Record with cost	No	150		300	USD
Tracker 123	2/17/2024	Record w/o cost	No	200			

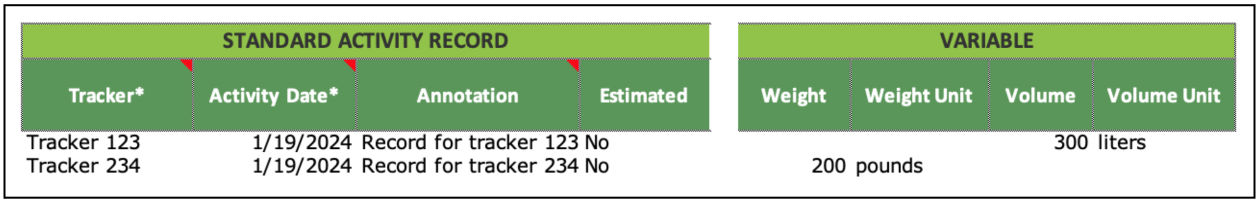
Figure 3 - Default Units

If every record to be loaded quantified weight in the default units of pounds, it might be tempting to assume that the *Weight Unit* column can be omitted entirely but this is not the case. Going back to our discussion of column *groups*, remember that groups must be complete. The application expects a pair of columns for each input metric, even if the cells in that column are left blank.

With respect to the example illustrated in Figure 3, note that the default value for the *Estimated* attribute is 'no'. As such, the user could also leave the *Estimated* column blank. In this case, if none of the activity records to be loaded are estimated, the user could omit the *Estimated* column entirely.

Different Sets of Attributes for Different Instances of the Same Object

The third case we'll consider is another case in which the loaded objects are of a type that requires variable columns. In this example, we use the same file to load activity records for different trackers. This is illustrated in Figure 4.



STANDARD ACTIVITY RECORD				VARIABLE			
Tracker*	Activity Date*	Annotation	Estimated	Weight	Weight Unit	Volume	Volume Unit
Tracker 123	1/19/2024	Record for tracker 123	No			300	liters
Tracker 234	1/19/2024	Record for tracker 234	No	200	pounds		

Figure 4 - DIfferent Objects With Different Attributes

The user has added a pair of columns to the standard template so that they can use the same bulkloader file to load activity records to different trackers that have different inputs. In this case, *Tracker 123* has a single *volume* input and *Tracker 234* has a single *weight* input. Cells corresponding to inputs that do not exist on a particular tracker are left blank for that tracker's row.

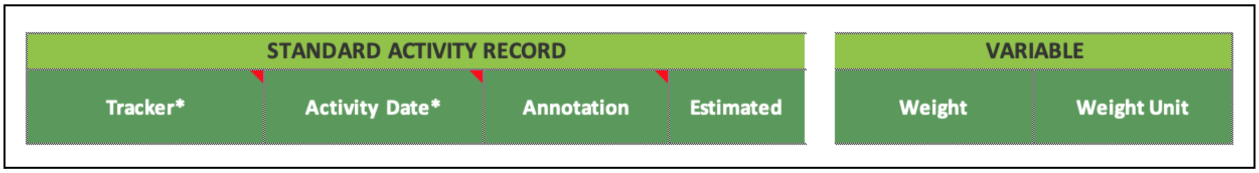
One could imagine a file containing many activity records for many different trackers, each with a single but different metric. In this case, all but one of the column pairs representing tracker metrics would be left blank on each row.

Column Ordering

In the following paragraphs, we'll describe each of the columns that are recognized for each of the loadable objects. For the sake of simplicity, we recommend adhering to the order in which the columns are described. However, as noted previously, columns can, for the most part, be in any order as long as the object parent keyword appears in the first populated column and grouped columns are kept together in the required order. Any exceptions will be noted in the description.

Activity Records

By now, the template for activity records should be familiar to the reader. Nonetheless, we'll specify them formally here. The basic set of columns for activity records is illustrated in Figure 5.



STANDARD ACTIVITY RECORD				VARIABLE	
Tracker*	Activity Date*	Annotation	Estimated	Weight	Weight Unit

Figure 5 - Columns for Loading Activity Records

Standard Columns

Tracker Name

identifying keyword: *Tracker*
required

The anchor column for the activity record template must be identified with a header that contains the string *Tracker* in the header text. Each row under the header must contain the name of the tracker to which the attribute values in that row will be loaded. The tracker name should exactly match the name of the tracker in the account. Users will have the opportunity to correct names for which the application cannot find a matching tracker.

Activity Date

identifying keyword: *Date*
required

The second column must have the string *Date* in the header. Each row under the header must contain the date for the activity record loaded, The following formats are acceptable:

- *mm/dd/yyyy*
- *mm/dd/yy*

Attempts to load activity records with an activity date preceding the tracker's start date or later than the tracker's end date (if specified) will be rejected with an error. In case of linked trackers, the activity record date must be no earlier than the earliest of all linked trackers' start dates and no later than the latest of all linked trackers' end dates.

Annotation

identifying keyword: *Annotation*
optional
default value: blank

The *Annotation* column is optional. If it is used, it must be identified by the keyword *Annotation* in the column header. Users can enter annotations for any record in this column. Annotations are limited to 255 characters in length. If none of the records to be loaded have an annotation, the user may omit the column entirely for both Excel or CSV files.

Supporting Documents

When more than a 255 character annotation is required, users can load documents to support each record. However, documents cannot be loaded using the bulkloader. Instead, users will have the ability to load documents manually and to associate them with specific activity records after those records have been loaded via the bulkloader. For further detail on loading documents to support activity records, see Appendix B.

Estimated Values

identifying keyword: *Estimated*
optional
default value: not estimated (actual)

The last standard column for the Activity Record template is the *Estimated* column. It must be identified by the keyword *Estimated* in the column header. The string 'yes' in any row in the estimated column will be interpreted to indicate that the record for the corresponding row is an estimated value. The string 'no' or a blank cell will be assumed to indicate that the record for the corresponding row is *not* estimated. Because the estimated column is optional, if none of the records to be loaded are estimated, the user may omit the column entirely for both Excel or CSV files.

Variable Columns

After the standard columns (tracker name, activity date and the optional annotation and estimated columns), the application expects a variable number of columns containing input values and units. Generally, a *pair* of columns is expected for each input metric. However, the *units* column, while recommended, is optional for each input. If the units column is omitted, the application will assume the default units for which the corresponding tracker is configured for the named input metric. At least one input must be specified for loaded activity records.

Input Name

identifying keyword: varies, depending on the input unit category or handle
required

The header for each value column must specify the name of the input to which the corresponding values in the rows below will be loaded. The name must be exactly as it appears on the tracker's manual loading form. Refer to Figure 6 below:

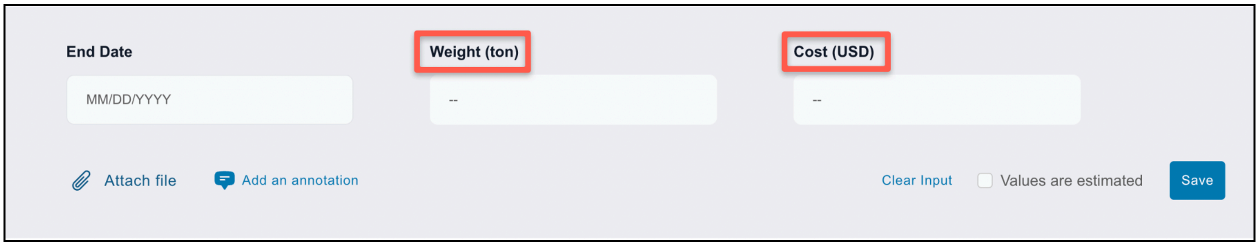


Figure 6 - Manual Loading Form for Activity Records

Figure 6 illustrates the manual loading form for a tracker that has two inputs; *weight* and *cost*. The corresponding columns in the bulkloader spreadsheet must have at least one column for each of these inputs with the headers *Weight* and *Cost*. These columns may be in any order. If only these input name columns are provided, the bulkloader will assume units of *ton* for the *Weight* column and *USD* for the *Cost* column.

Note that if either of these inputs are optional for the tracker, then it is not necessary to include any columns for the corresponding metric.

Handles

In general, for many trackers, the input name will be the same as the unit type (or category) of the input. However, users can assign 'handles' to name an input in which case, that handle must be named in the corresponding column header. Handles are required in order to disambiguate inputs when an activity record includes two inputs of the same unit category.

Recognized Unit Types

The following unit types or categories are recognized by the application:
area, count, currency, distance, energy, freight, gas, time, volume, weight

Additional inputs may occur on any tracker, reflecting custom unit types or disambiguating multiple columns of the same unit type.

Input Units

identifying keyword: *Unit*
optional

If this column is included, the header must include the string *Unit* or *Units* in the header. No other identifying information is required in the header - the application will assume that each column with the string *Unit* corresponds to the input named in the column immediately to the left of it in the bulkloader.

Unit Rows

Each tracker is configured to expect a default unit for each input. The default units can easily be found by inspecting a tracker's manual loading form (see Figure 6), where the default units are indicated parenthetically immediately following the input name. These units will be assumed for each value unless an alternate (allowable) unit is specified in the corresponding cell.

Cost is Special

For activity records on trackers that belong to a tracking node in a certain country, the bulkloader will generate a warning if the currency specified inthe units column is not the standard currency for that country. Const units shoud be specified uisng the ISO three letter code for the corresponding currency. See appendix G for a list of acceptable currency codes.

Allowable Units

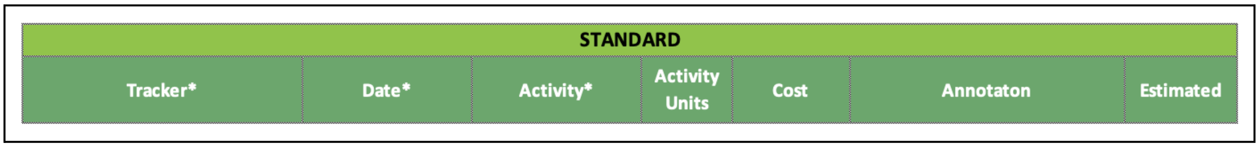
Appendix A lists the allowable units grouped by unit type.

Quick Activity Records

The *Quick Activity Records* bulkloader continues with the simplicity of the v1 *Activity Records* bulkloader. It is a simplified version of the full v2 *Activity Records* bulkloader. This template can be used to efficiently load activity records to a set of one or more trackers so long as each has just a single activity input and, optionally, a single cost input.

Whereas the full v2 *Activity Records* bulkloader requires the user to specify each non-cost input across the set of loaded trackers using a distinct column named for that input, the *Quick Activity Records* bulkloader allows a single *Activity* column to be used to specify the single non-cost inputs for all trackers.

Attempts to use this form to load activity data to a tracker with more than one non-cost inputs will be rejected with an error.



STANDARD						
Tracker*	Date*	Activity*	Activity Units	Cost	Annotaton	Estimated

Figure 7 - Alternate Form for Loading Activity Records With a Single Non-Cost Input

Standard Columns

Tracker Name

identifying keyword: *Tracker*
required

The anchor column for the activity record template must be identified with a header that contains the string *Tracker* in the header text. Each row under the header must contain the name of the tracker to which the attribute values in that row will be loaded. The tracker name should exactly match the name of the tracker in the account. Users will have the opportunity to correct names for which the application cannot find a matching tracker.

Activity Date

identifying keyword: *Date*
required

The second column must have the string *Date* in the header. Each row under the header must contain the date for the activity record loaded, The following formats are acceptable:

- *mm/dd/yyyy*
- *mm/dd/yy*

Attempts to load activity records with an activity date preceding the tracker's start date or later than the tracker's end date (if specified) will be rejected with an error. In case of linked trackers, the activity record date must be no earlier than the earliest of all linked trackers' start dates and no later than the latest of all linked trackers' end dates.

Activity

identifying keyword: *Activity*
required

The *Activity* column is used to quantify *activity* on the tracker. Users are expected to enter a numeric value quantifying the activity for the record. It is recommended that users always use the next column, *Activity Units* to specify the units in which the activity is quantified. However, the user may omit activity units, as described under the next column, in which case, the bulkloader will assume the current default units for the tracker. Users may refer to the tracker's manual input form (illustrated in Figure 6) to determine the current default units.

Activity Units

identifying keyword: *Units*
optional

The *Activity Units* column may be used to specify the units in which the corresponding tracker's *activity* (quantified in the column immediately to the left of the *Activity Units* column) is specified. The user may omit this column or leave the cell for a particular record blank, in which case, the bulkloader will assume the current default units for the tracker. Users may refer to the tracker's manual input form (illustrated in Figure 6) to determine the current default units.

Cost

identifying keyword: *Cost*
optional

The *Cost*column may be used to quantify cost for the activity record. The bulkloader assumes that there is a single cost input on the tracker referenced. Attempts to load a row that specifies a cost value to a tracker that has either no cost input or more than one cost input will be rejected with an error.

Annotation

identifying keyword: *Annotation*
optional
default value: blank

The *Annotation* column is optional. If it is used, it must be identified by the keyword *Annotation* in the column header. Users can enter annotations for any record in this column. Annotations are limited to 255 characters in length. If none of the records to be loaded have an annotation, the user may omit the column entirely for both Excel or CSV files.

Supporting Documents

When more than a 255 character annotation is required, users can load documents to support each record. However, documents cannot be loaded using the bulkloader. Instead, users will have the ability to load documents manually and to associate them with specific activity records after those records have been loaded via the bulkloader. For further detail on loading documents to support activity records, see Appendix B.

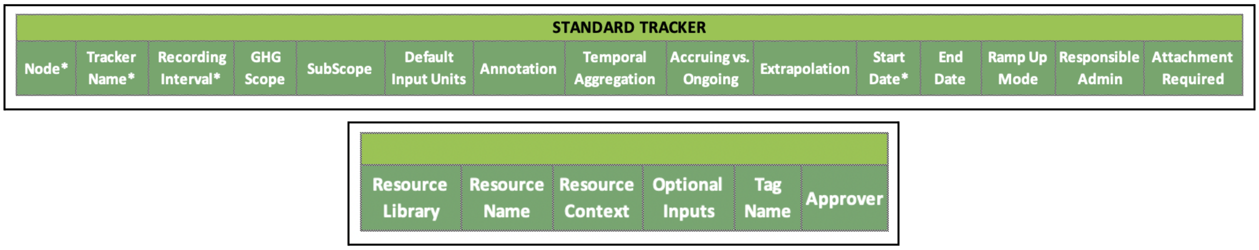
Estimated Values

identifying keyword: *Estimated*
optional
default value: not estimated (actual)

The last standard column for the Activity Record template is the *Estimated* column. It must be identified by the keyword *Estimated* in the column header. The string 'yes' in any row in the estimated column will be interpreted to indicate that the record for the corresponding row is an estimated value. The string 'no' or a blank cell will be assumed to indicate that the record for the corresponding row is *not* estimated. Because the estimated column is optional, if none of the records to be loaded are estimated, the user may omit the column entirely for both Excel or CSV files.

Trackers

The basic set of columns for trackers is illustrated in Figure 8.



STANDARD TRACKER														
Node*	Tracker Name*	Recording Interval*	GHG Scope	SubScope	Default Input Units	Annotation	Temporal Aggregation	Accruing vs. Ongoing	Extrapolation	Start Date*	End Date	Ramp Up Mode	Responsible Admin	Attachment Required

Resource Library	Resource Name	Resource Context	Optional Inputs	Tag Name	Approver

Figure 8 - Columns for Loading Trackers

Standard Columns

Node Name

identifying keyword: *Node*
required

The first column is indicated by including the keyword *Node* in the column header of the primary header row. The values in the content rows for this column indicate the tracking node to which the tracker will be attached. The application will attempt to match node names from each row to an existing tracking node. Whitespace will be ignored. Users will have the chance to correct node names if a name in the file is not recognized.

Tracker Name

identifying keyword: *Tracker*
required

The tracker name column must contain the keyword *Tracker* in the header. Content rows should specify the name of each tracker to be loaded. Header and content rows are case insensitive. Whitespace will be ignored except that multi-word tracker names from content rows will be trimmed to include a single blank between words.

Note that tracker names must be unique within accounts. As such attempts to load a tracker with the same name as an existing tracker will be rejected by the application. Users will have a chance to correct such duplicates.

Recording Interval

identifying keyword: *Interval*
required

This column must contain the keyword *Interval* in the header. Content rows should specify an expected recording interval from the set of allowable intervals:

weekly, monthly, bi-monthly, quarterly semi-annual, annual or *aperiodic*

GHG Scope

identifying keyword: *Scope*
optional
default value: no scope

This column, if used, must contain the keyword *Scope* in the header. Content rows should specify a GHG scope. Allowable values are one of the integers *1, 2* or *3*.

SubScope

identifying keyword: *Subscope*
optional
default value: blank

This column, if used, must also contain the keyword *Subscope* in the header. If used, it must be inserted immediately to the right of a *Scope* column. The content rows in this column are used to specify a scope 3 *sub scope* from the allowable set of 15 scope 3 categories. The 15 scope 3 categories are:

1. *purchased goods & services*
2. *capital goods*
3. *energy*
4. *upstream transport*
5. *waste*
6. *business travel*
7. *employee commuting*
8. *upstream leased assets*
9. *downstream transport*
10. *processing of sold products*
11. *use of sold products*
12. *end-of-life of sold products*
13. *downstream leased assets*
14. *franchises*
15. *investments*

Users specify a subscope by including the category number, as an integer, in each content row for the column.

Note that content in this column is valid only in rows in which the *Scope* column to the left indicates scope *3*. Non-blank values in the *Subscope* column in rows which do not indicate scope *3* will be rejected with an error.

Default Input Units

identifying keyword: *Units*
required

Default units must be specified for each of the inputs across the set of resources attached to the tracker, whether optional or required. To do so, the user would include a column with the keyword *Units* in the header. For the values for this column, the user would include a comma separated string identifying the input name or handle, followed by a colon and the default unit.

For example - if the set of inputs provided for by the attached resources are of type *energy* and *cost*, then the value string for the default input units for this tracker might read:

```
energy:therms, cost:usd
```

Refer to the appendices for the set of supported units.

Annotation

identifying keyword: *Annotation*
optional
default value: blank

This column, if used, must be identified by the keyword *Annotation* in the column header. Users can enter annotations for any tracker in the content rows in this column. Annotations are limited to 255 characters in length. If none of the trackers to be loaded require an annotation, the user may omit the column entirely.

Temporal Aggregation

identifying keyword: *Temporal*
optional
default value: summing

This column, if used, is identified by the keyword *Temporal* in the header. The values in the content rows for this column indicate whether activity records for the tracker should be rolled up by summing them or averaging them (weighted by the time spanned by each record).

Allowable values for content rows for the *Temporal Aggregation* column are *summing* or *averaging*. Rows with a blank value in the column will default to *summing*. Rows with any other value will be rejected with an error.

Accruing or Ongoing

identifying keyword: *Accruing*
optional
default value: accruing

This column, if used, is identified by the keyword *Accruing* in the header. The values in the content rows for this column indicate how activity dates on activity records loaded for the tracker should be interpreted. Allowable values are:

- *accruing* - indicates that the activity date on records loaded to this tracker indicate the *end* of the activity period quantified by the record (this is typical for utility trackers, such as trackers tracking an electricity account or a natural gas account).

- *ongoing* - indicates that the activity date on records loaded to this tracker indicate the *start* of the activity period quantified by the record (this is typical for trackers tracking fuel tanks or refrigerants)
- *instantaneous* - indicates that the activity date on records loaded to this tracker indicate a single day during which the activity quantified by the record can be assumed to have occured (this is typical for trackers tracking instantaneous events such as methane flares).

Rows with a blank value in the column will default to *accruing*. Rows with any other value will be rejected with an error.

Extrapolation

identifying keyword: *Extrapolation*
optional
default value: off

This column, if used, determines whether the tracker will interpolate and extrapolate to fill missing records and if so, what algorithm it will use. At present, the only extrapolaton algorithm used is *seasonal.* It follows that content rows must contain one of the values *seasonal*, *off* or be blank (defaulting to *off*).

Start Date

identifying keyword: *Start*
required

This column must specify the date on which activity tracked by the corresponding tracker should be considered to first start. The following formats are acceptable:

- *mm/dd/yyyy*
- *mm/dd/yy*

Note that in case a tracker is loaded to a tracking node with a specified start date, the corresponding row will be rejected by the bulkloader if the start date of the tracker is earlier than the start date of the tracking node.

End Date

identifying keyword: *End*
optional
default value: blank

This column, if used, specifies the date on which activity tracked by the corresponding tracker should be considered to end. It can also be thought of as a *retirement* date for the tracker. If no value is specified or the column is omitted, the corresponding trackers are considered to still be active.

Date values in this column should be formatted using the same format as defined for the *Start Date* column.

Note that in case a tracker is loaded to a tracking node with a specified end date, the corresponding row will be rejected by the bulkloder if the end date of the tracker is later than the end date of the tracking node.

Ramp Up Mode

identifying keyword: *Ramp*
optional
default value: normal

This column, if used, tells the application how to quantify activity immediately preceding the corresponding tracker's start date. Allowable values are:

- *normal* - meaning that the start date of the tracker indicates the date on which the activity source represented by the tracker truly started operating (such as the date a new boiler or a new electricity meter was installed).
- *immediate* - meaning that the activity source represented by the tracker should be considered to have been humming along indefinitely and that the tracker's start date merely indicates the date on which *measurement* or *tracking* of the activity began.

Rows with a blank value in this column will default to *normal* ramp up.

The ramp up mode is primarily used to inform queries that quantify a running average activity over several periods. For trackers in *normal* mode, these queries will show a gradual ramp up from zero. Queries on trackers configured for *immediate* ramp up will backfill the starting intervals of the query averaging period with the average value of the first few periods following the tracker's start date.

Responsible Admin

identifying keyword: *Responsible*
optional
default value: no responsible user (blank)
This column can be used to indicate a single specific user who will be responsible for maintaining data on the tracker and subject to the *Admin Prompt* process. The keyword *Responsible* must be included in the header row. The corresponding cells in the value rows must be blank or contain a valid *username* credential indicating the responsible user.

Attachment Required

identifying keyword: *Attachment*
optional
default value: no
This column can be used to indicate that activity records on the tracker must include an attachment. Allowable values are *yes* or *no*. If *yes* is indicated, the user will not be able to load activity records to the tracker without an attachment documenting each activity record. At present, it is not possible to bulkload activity records with attachments. Therefore, for any trackers requiring attachments, activity records cannot be bulkloaded.

If the *Attachment* column is omitted or the corresponding content rows for any trackers are blank, those trackers will default to not require attachments.

Variable Columns

After the standard columns, Users may include an arbitrary number of columns indicating resources, tags and approvers corresponding to each tracker.

Resource Library

identifying keyword: *Library*
optional
default value: blank

The library column can be used to specify the resource library from which to draw any resources named in columns to the right of the subject library column. If resources named are unique within the account, the *library* column is unnecessary. However, if any of the resources named in subsequent columns are not unique in the account and no *library* column can be found to disambiguate them, then the corresponding row will be rejected with an error.

An arbitrary number of *library* columns may be included so long as there is at least one *resource* column between each *library* column and the next.

Any libraries named in *library* columns must be available to the account. Otherwise, the row will be rejected with an error.

Resource Name

identifying keyword: *Resource*
required
default value: no associated resource

The resource name column can be used to name one or more resources that describe the activity tracked by the corresponding tracker and how that activity might be converted to certain impacts (such as GHG emissions). Although most trackers will likely track activity using a single resource, users can attach an arbitrary number of resources to any tracker by repeating the resource name column using the header keyword *Resource*.

The following requirements must be met for each content row or else the row will be rejected with an error:

1. Any non-blank value must match the name of a resource that is available to the tracker (the resource is present either in the account's custom library or in one of the maintained libraries available to the account).
2. Resource names must correspond to a single resource in each account or must be disambiguated by a *library* column to the left of the *resource* column.

Users should bear in mind that resources configured on a tracker will determine the activity inputs (categories) tracked by the tracker.

Resource Context

identifying keyword: *Context*
optional
default value: S2DR:location

The resource context column can be used to specify the *context* in which the corresponding resource should be connected to the tracker. At this time, the only contexts allowed are in the S2DR group and can be specified with one of the strings:

- *S2DR:location*
- *S2DR:market*
- *S2DR:market indicative*

To control the context in which a resource is attached to a tracker, the *resource context* column must be grouped with the relevant *resource* column, immediately to the right of the *resource name* column or following another context column for the same resource. (At this time only the S2DR context group is supported and only a single context in a group is allowed).

Omitting the context column or leaving it blank for a tracker will result in the resource being connected to the tracker in the default *S2DR:location* context.

Optional Inputs

identifying keyword: *Optional*
optional
default value: none

By default, activity records loaded to a tracker will be assumed to quantify a value for each input specified for the set of resources that are attached to the tracker. Users may mark certain inputs as optional for the tracker and by extension, for loaded activity records. To do so, the user would include a column with the keyword *Optional* in the header, immediately to the right of the corresponding *default input units* column. For the values for this column (or columns if there are multiple resources being loaded), the user would include a comma separated string identifying each of the optional inputs for the tracker ⇔ resource association.

For example, a tracker might have a resource attached that specifies *volume, weight* and *cost* as inputs, with the *cost* column marked optional (indicating that the resource can serve its purpose without requiring the cost input). By default, the *cost* column would still be required on any activity record loaded to the tracker unless the user indicates that it is optional for the tracker by including the *optional* column and specifying the string 'cost' in the corresponding value cell.

Note that by default all inputs for all attached resources are assumed to be required. Any attempts to load a tracker indicating optional inputs will be rejected with an error unless all resources on the tracker also indicate that the inputs are optional.

Tag Name

identifying keyword: *Tag*
optional
default value: blank

Any number of tag columns may be included by specifying the keyword *Tag* in the column header. Content rows in each *tag* column can then include the name of a child or parent tag that will be associated with the tracker.

Content in any non-blank *tag* column must match one of the tags configured for the account or else the row will be rejected with an error.

When a child tag is specified, the parent tag will also be automatically attached to the tracker. Users can attach only a single child tag to any tracker for any tag family. Attempts to attach a different child tag for a tracker already tagged with a child tag in the same family will be rejected with an error.

When a tracker is loaded to a tracking node for which a tag has been specified, the tracker may not specify any tags in any tag group existing on the node (parent or child). Attempts to load trackers that violate this rule will be rejected with an error.

Approver

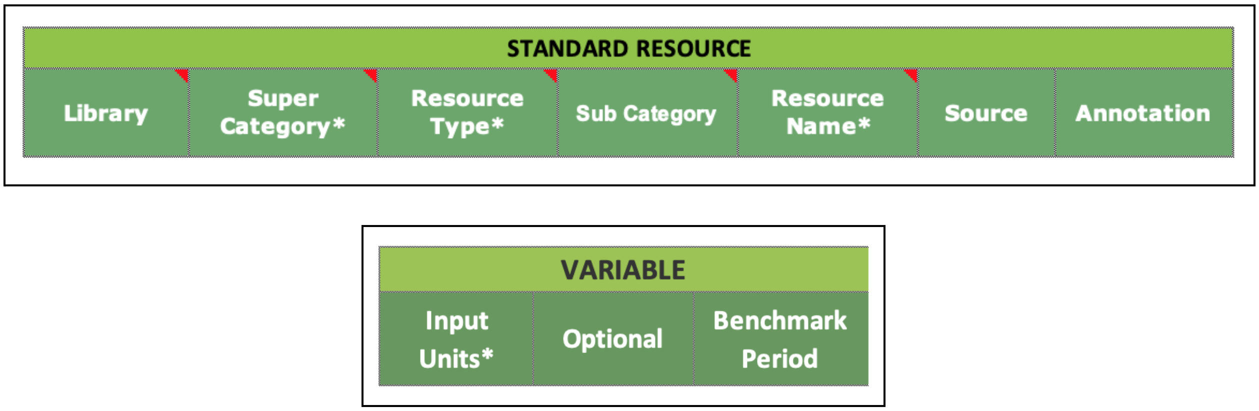
identifying keyword: *Approver*
optional
default value: blank

Any number of *Approver* columns may be included to indicate that the corresponding tracker is subject to the application's approval workflow with certain named approvers. Users should include a single *login name* corresponding to an approver in each column. Any activity records loaded to the tracker will then await approval by the designated approver before the activity is attributed to the tracker.

Content rows for this column must include a string matching the login or username of an authorized user in the relevant account. A blank indicates that the corresponding tracker is not subject to approval workflow. Any non-blank value that does not match a valid login or username will cause the row to be rejected with an error.

Resources

The basic set of columns for resources is illustrated in Figure 9.



STANDARD RESOURCE						
Library	Super Category*	Resource Type*	Sub Category	Resource Name*	Source	Annotation

VARIABLE		
Input Units*	Optional	Benchmark Period

Figure 9 - Columns for Loading Resources

Standard Columns

Library

identifying keyword: *Library*
required

The first column is indicated by including the keyword *Library* in the column header of the primary header row. The values in the content rows for this column must indicate the name of the resource library to which each resource will be loaded. In most cases, this will be the name of the account's custom resource library.

Super Category, Resource Type & Sub Category

identifying keyword: *Super*, *Type*, *Sub* (respectively)
required, required, optional (respectively)

The three columns *Super Category*, *Resource Type* and *Sub Category* are *taxonomy* columns. These can be used to organize resources within each library and can also be used as 'selectors' in queries, selecting for example, all trackers that point to a resource having the resource type *Stationary Combustion*.

Super category and resource type must be specified. There are several built-in values allowed for each of these attributes, as enumerated in Appendix A. Users can define additional custom super categories and resource types for their account simply by entering these in the corresponding column when loading resources. Note that for any non-standard values for these attributes, the loader will ask the user to confirm their addition to the account the first time they are encountered.

The sub category is optional. There are no built in values for sub category. Users may use this attribute for further classification and/or selection.

Resource Name

identifying keyword: *Resource*
required

The resource name column must contain the keyword *Resource* in the header. Content rows should specify the name of each resource to be loaded.

Source

identifying keyword: *Source*
optional

The *Source* column can optionally be used to indicate references used in defining the corresponding resource. These are typically hyperlinks referring to online documentation. Hyperlinks will be recognized by the application and will be made clickable in the resource's 'show' page.

Note that users can also specify sources for any individual conversion factor set belonging to each resource. These would be specified on the bulkloader template for conversion factor sets.

Annotation

identifying keyword: *Annotation*
optional

The *Annotation* column can optionally be used to describe a resource. Annotations are visible in the resource's 'show' page and are typically used to explain details regarding the activity described by a resource and how it is quantified or converted to impacts.

Similar to *Source*, *Annotation*s are also available for individual conversion factor sets.

Variable Columns

After the standard columns, Users may include an arbitrary number of columns indicating inputs and corresponding benchmark periods.

Input Units

identifying keyword: *Input*
required

At least one input must be specified for any resource. The input indicates a unit type or unit category (such as *weight* or *volume*) that qualifies (and quantifies) the activity tracked by the

resource. A resource's inputs correspond to the *inputs* on activity records in the ledgers of the trackers that point to the resource. As such, the same unit types or categories allowed for activity record inputs are allowed for resource inputs. These include:
area, count, currency, distance, energy, freight, gas, time, volume, weight

Additional inputs may be allowed on any resource, reflecting custom unit types or disambiguating multiple columns of the same unit type (in the form of *handles*).

Optional

identifying keyword: *Optional*
optional
default value: *no (meaning the column is required or *not* optional)*

By default, all inputs are assumed to be required. However, the user can designate inputs as optional by including a column with the keyword *Optional* in the header and entering the value *yes* in the row for the corresponding resource.

Benchmark Period

identifying keyword: *Benchmark*
optional
default value: *365*

Any input can also have a *benchmark period*. The *Benchmark Period* column corresponding to a input must contain the keyword *Benchmark* in the column header and must be located immediately to the right of the corresponding input.

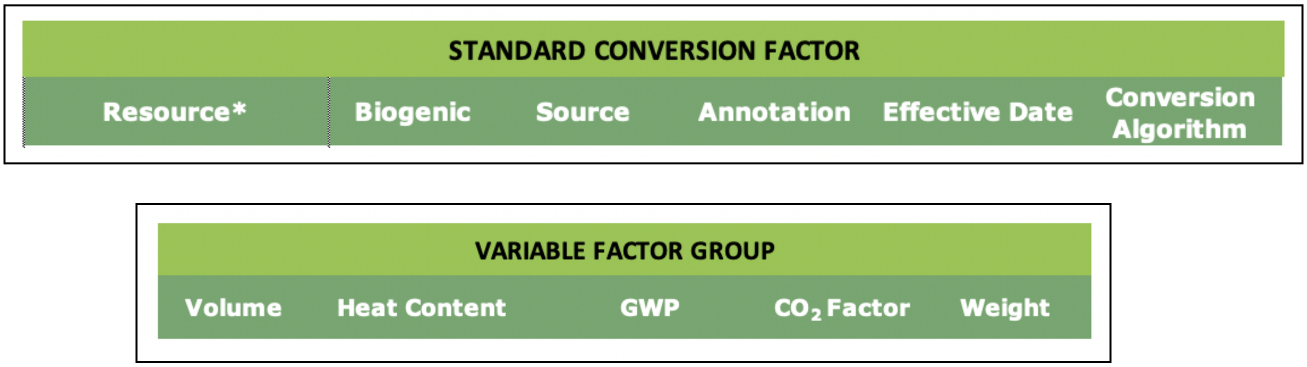
The benchmark period is used to control how activity is quantified for a *benchmarking resource*. Consider for example, a *CBECS* resource chain*.* Such a resource chain consists of two resources - the first converts facility area to electricity energy or heating fuel energy. The next resource converts that energy to GHG emissions.

That first resource is considered a benchmarking resource because it quantifies on average, how much energy is used per unit area for a certain building type in a certain climate zone. Note the missing parameter - time. In order to be useful, the application needs to know the *time period* over which a benchmarking resource is quantifying the metric that it is benchmarking, In the case of the CBECS example, the benchmarking resource benchmarks energy used over a one year period.

The benchmark period is quantified as an integer number of days, As such, for that first resource, the user would include a *Benchmark Period* column for the *area* input on the resource. The user would populate the corresponding content cells with the value 365, indicating that the resource is benchmarking energy for a one year period based on the area value in tracked activity records.

Conversion Factor Sets

The basic set of columns for conversion factor sets is illustrated in Figure 10.



STANDARD CONVERSION FACTOR					
Resource*	Biogenic	Source	Annotation	Effective Date	Conversion Algorithm

VARIABLE FACTOR GROUP				
Volume	Heat Content	GWP	CO_2 Factor	Weight

Figure 10 - Columns for Loading Conversion Factors

Each row in the conversion factor bulkloader combines one or more conversion factors (each represented by a group of variable columns called a *factor group*) into a *conversion factor set* (CFS). Each CFS will include (at least) the required subset of standard columns once, followed by one or more factor groups. Each factor group can be thought of as defining an *output* of the CFS. The factor group example illustrated in Figure 9 defines a single factor or output (CO_2), that is quantified in *weight* and is converted from two inputs; volume and heat content.

Standard Columns

Resource

identifying keyword: *Resource*
required

The first column is indicated by including the anchor keyword *Resource* in the column header of the primary header row. The values in the content rows for this column indicate the name of the resource to which the CFS in the corresponding row will be loaded.

Biogenic

identifying keyword: *Biogenic*
optional
default value: *0*

The *Biogenic* column can optionally be used to indicate that the corresponding CFS is biogenic. Biogenic CFSs get special treatment as described subsequently. If a biogenic column is included, the column header must contain the keyword *biogenic*. Allowable values are blank (indicating that the corresponding CFS is non-biogenic) or an integer from 0 to 100 that indicates the percentage of the activity record submitted for conversion that should be considered biogenic.

If a CFS is loaded to a resource with preexisting CFSs and no value specified for the biogenic attribute, it will inherit the value of the prior CFS (most recent *effective date* value).

Source

identifying keyword: *Source*
optional

The *Source* column can optionally be used to indicate references used in defining the corresponding CFS. These are typically hyperlinks referring to online documentation. Hyperlinks will be recognized by the application and will be made clickable in the resource's 'show' page.[2]

Annotation

identifying keyword: *Annotation*
optional

The *Annotation* column can optionally be used to describe the corresponding CFS. Annotations are visible in the resource's 'show' page and are typically used to explain details regarding the CFS s on the corresponding row.[3]

Effective Date

identifying keyword: *Date*
optional
default value: 0*1/01/1970*

If used, this column must have the string *Date* in the header. Each row under the header must contain an effective date for the CFS specified in the row, The following formats are acceptable:

- *mm/dd/yyyy*
- *mm/dd/yy*

The effective date specifies the *starting date* on which the CFS in the corresponding row should be applied.

Conversion Factor Set Effective Dates

Let's take a moment to consider the use of the *effective date* attribute of a CFS. There are CFSs for which an effective date is generally meaningless. Consider for example, a CFS that converts a volume of fuel into a certain weight of GHG emissions (a.k.a. an *emissions factor set*). The emissions calculated will generally be the same regardless of the date on which it is combusted.

Consider, on the other hand, a CFS that converts a kWh of electricity to a weight of GHG emissions. Typical electricity providers change generating equipment and fuel mix frequently.

[2] Note that users can also specify sources for the resource that contains the CFS specified. These would be specified on the bulkloader template for resources.
[3] Similar to *Source*, *Annotation*s are also available for the resource that contains the CFS specified.

For this reason, CFSs quantifying specific GHGs for a certain electricity provider (or many other time varying activities) must be considered in the context of a specific date (the *effective* date of the CFS).

The Initial Conversion Factor Set

It's important to understand how the application applies CFSs' effective dates. Generally, an effective date tells the application to apply the CFS to convert activity from activity records dated *on or after the effective date*. This raises the question as to how the application determines which CFS to apply to activity occurring *before* the earliest effective date. The answer is that (somewhat counterintuitively) the CFS with the earliest effective date is considered to be the *initial set* and is applied not just to activity dated *on or after* that date but also to activity *before* that date.[4]

Using the Default Effective Date

Omitting an effective date for a CFS is allowable, whether for all rows in the bulkloaded sheet (in which case the column can be omitted in its entirety) or for a specific subset of rows corresponding to specific CFSs. The bulkloader will assign the default effective date of 1/1/1970[5] to any CFS loaded without an explicitly defined effective date. Doing so indicates that the corresponding CFS(s) will be the initial set(s) for the corresponding resource(s).

As such, if any CFS is loaded to a resource without an effective date, the user will never be able to bulkload a subsequent CFS as the initial set. Subsequent CFSs for the resource will have to be loaded with explicitly specified effective dates greater than 1/1/1970 or else they will be rejected with an error.

Conversion Algorithm

identifying keyword: *Algorithm*
optional
default value: *simple product*

If used, this column must have the keyword *Algorithm* in the header. Omitting a value for the conversion algorithm column for a CFS is allowable, whether for all rows in the bulkloaded sheet (in which case the column can be omitted in its entirety) or for a specific subset of rows corresponding to specific CFSs. Doing so indicates that the conversion *algorithm* applying to the corresponding CFS will be the default (*simple product*) algorithm described below.[6]

Other algorithms may be specified using the appropriate string for the algorithm. Supported algorithms are documented in Appendix C. Most algorithms (but not the default) will require that the input columns be specified in a certain order. That order is defined with each supported algorithm.

[4] As such, if there is a single conversion factor set for a resource, that conversion factor set applies across all time.
[5] 1/1/1970 is considered the 'start of time' by the application's database.
[6] The default algorithm can also be indicated specifically by using the string *simple product* in the column.

Note that all CFSs for a given resource must use the same conversion algorithm. Attempts to load a CFS to a resource that has existing CFSs will be rejected with an error if the submitted CFS indicates an algorithm that conflicts with previously loaded CFSs.

Variable Columns

After the standard columns, Users may include an arbitrary number of variable columns corresponding to *factor groups*, each specifying one conversion factor (output). Each group may specify one or more *inputs* but always has one single *output*.

When a Loaded CFS Has a Different Set of Factor Groups

It's not necessary that a CFS to be loaded have exactly the same set of factor groups as those already existing on prior CFSs for the resource.

When a Loaded CFS is Missing Factor Groups

Newly loaded CFSs may omit factor groups that exist in earlier CFSs that have already been loaded. In this case, the application will automatically copy any missing factor groups from the most recent CFS, with the same factors.

When a Loaded CFS has Factor Groups that are not Present on Preexisting CFSs

Users may add factor groups to a resource's CFS over time. For example, the first few CFSs specified for a resource may not include one or more factor groups for GHG emissions but not a factor group for *energy*. At some later date, the user may load a CFS that does include a factor group for energy. In this case, the application will retroactively copy the new factor group(s) to all prior CFSs. However, factors created on prior CFSs will have a value of 0.

Applying the Conversion Algorithm

We will describe the rules and requirements around the variable columns shortly but first, we will digress to explain how the factor groups they represent are used by the application.

The application will use the values loaded with a CFS to convert activity from one or more inputs on any activity record submitted for conversion to one or more outputs, applying the specified conversion algorithm. This concept is illustrated in Figure 11 for a single activity record and a single CFS with three factor groups.

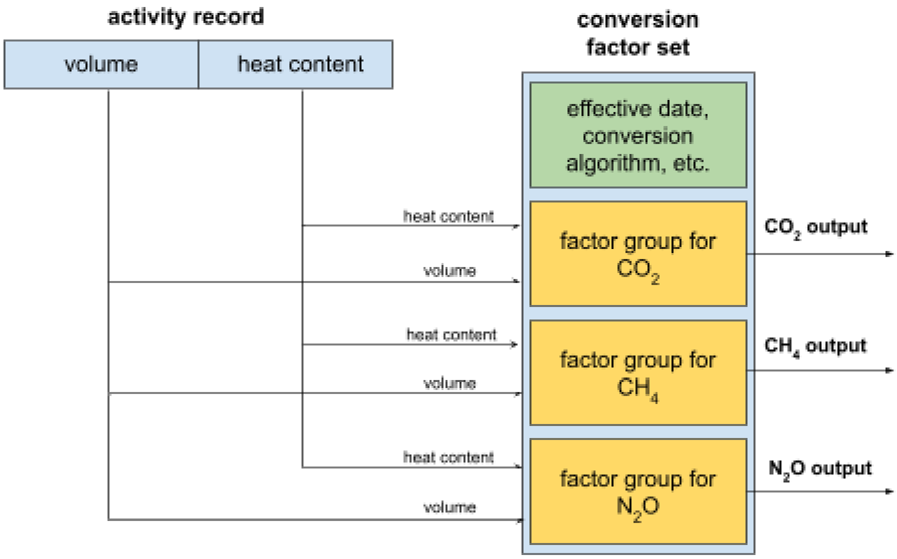


Figure 11 - Conversion Algorithm is Applied Separately for Each Factor Group

The Simple Product Conversion Algorithm

The default conversion algorithm is also known as the *simple product* algorithm. The simple product algorithm is applied across each factor group independently. It is described below for a CFS with a single factor group.

1. The algorithm first matches each CFS *input* in each row to a corresponding input in any activity record submitted for conversion, based on the input unit category or unit handle, subject to these rules:
   a. All unit categories or handles listed for a single CFS must be unique within a factor group.
   b. If no input is found on the activity record to match a certain input column on the CFS, then the value ‘1’ will be assumed for the activity record’s input for the group.
   c. If there exists an input on the activity record for which no match can be found in a column group then that column from the activity record will be ignored for the purpose of calculating the output for the group.
   d. If there are no columns in common between the activity record and any of the output groups in the CFS, an error will be generated when attempting to connect the tracker and the CFS.
2. The algorithm will then convert the numeric value of each *input* column from the activity record’s units to the units defined in the bulkloader input column for the CFS.
3. Next, it will multiply all the matched and converted input values from the activity record together to yield a single product for each group.
4. Next, it will multiply that product by the value loaded from the *output* or *factor* column of the group.

5. That final product will be interpreted by the application to be the result of the activity conversion expressed in the output units indicated for that group.

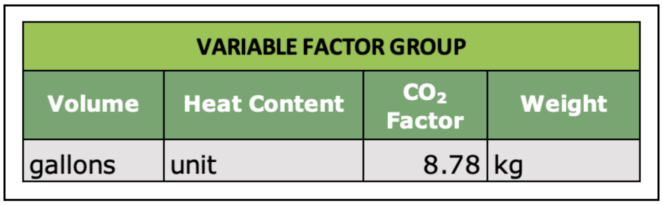
The simple product algorithm is expressed with the following equation that is applied separately for each column group:

$$output = input_0 \times ...\ input_n \times output\,factor$$

Equation 1 - Calculating the Converted Value (Output) for Each Group

The Simple Product Conversion Algorithm in Operation

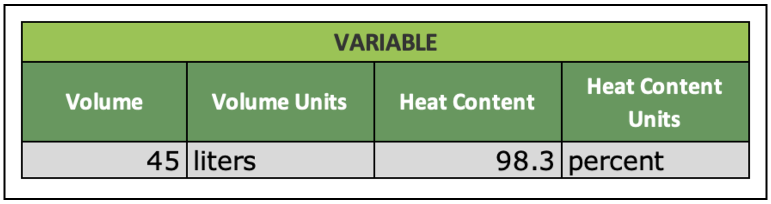
To illustrate the simple product algorithm in operation, consider the following example. The CFS for this example is illustrated below in Figure 12.



VARIABLE FACTOR GROUP			
Volume	Heat Content	CO_2 Factor	Weight
gallons	unit	8.78	kg

Figure 12 - Sample CFS Bulkloader Row

In this example, the CFS is applied to activity records from a ledger with two inputs; *fuel volume* and *heat content*. The activity record considered for this example was loaded with the bulkloader indicated in Figure 13.



VARIABLE			
Volume	Volume Units	Heat Content	Heat Content Units
45	liters	98.3	percent

Figure 13 - Sample Activity Record

The steps of the simple product algorithm are applied as follows:

Step 1
The CFS *Volume* input is matched to the activity record column for the unit category *Volume*. Then, the CFS *Heat Content* input is matched to the activity record column for the handle *Heat Content*.

Step 2
Units for each input are converted from activity record units to CFS units:

45 liters = 11.8877 gallons
98.3 percent = 0.983

Step 3

The two converted inputs are multiplied together:

$$11.8877 \times 0.983 = 11.6856091$$

Step 4

That product is multiplied by the *factor* from the CFS:

$$11.6856091 \times 8.78 = 102.5996479$$

Step 5

Finally, the converted result is *102.5996479 kg* CO_2

This sample application of the simple product algorithm is for a CFS with a single factor group (for the CO_2 factor). Figure 11 illustrates a CFS with *three* factor groups. To load that CFS, it would be necessary to repeat the variable column group two more times, once for the CH_4 factor and again for the N_2O factor.

To this point, note that the CFS in this example omits the GWP column for the factor group. We will discuss this column subsequently as we define the variable columns that make up a factor group.

Input Units

identifying keyword: varies, depending on the input unit category or handle
required

Each factor group starts with one or more columns indicating in the header, the unit *category* or *handle* to match from an activity record submitted for conversion. In the example described above, the CFS has a single factor group with two columns; one with the header *volume* and another with the header *heat content*. These are the *inputs* used by the factor group.

Note that any inputs specified in a CFS must be defined for the resource to which the CFS belongs. Attempts to load a CFS with input units that are not defined for the corresponding resource will be rejected with an error.

The *values* for the input columns, indicated in each of the rows in the bulkloaded sheet, will specify the input *units* that the CFS is assuming for the purpose of conversion. In the example illustrated in Figure 11, the value of *gallons* for the column input (ignoring the *heat content* column for the moment) tells us that the conversion to CO_2 expressed in the output or factor column is 8.78 kg *per gallon*. The value of *unit* for the heat content tells us that the conversion factor of 8.78 assumes a heat content of ‘1’ (or one hundred percent).

GWP

identifying keyword: *GWP*
optional
default value: none

GWP is an optional column, indicated by the string *GWP* in the column header. This column is used to indicate a special type of factor group. Specifically, a non-blank value in the GWP column indicates that the factor group is an *emissions factor* for a greenhouse gas (GHG). Emissions factors get special treatment by the application. Note that emissions factors must specify *weight* as their output unit category. Factor groups that have a GWP value specified and an output unit category that is not *weight* will be rejected with an error by the bulkloader.

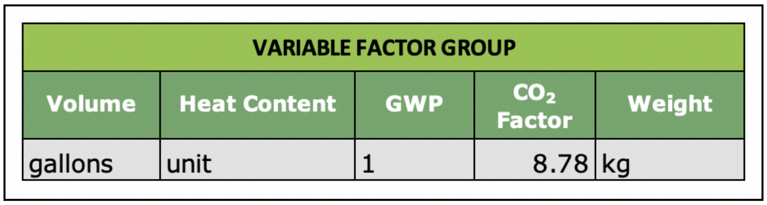
When the application detects an emissions factor in a CFS, it generates one additional output factor for the CFS (exceptions/special cases noted subsequently). That factor is *CO_2e* and is calculated from all emissions factors in the CFS. The CO_2e output factor is calculated in one final step after the simple product algorithm described above (or after any other algorithm). CO_2e is calculated by:

1. Finding all the emissions factor output values previously calculated for the CFS
2. Converting them to the common weight unit *kg*
3. Multiplying each by a constant indicated by their GWP value
4. Summing the products.

Note that the value for the GWP column may be quantified directly as an integer or as a string specifying the IPCC assessment report from which to draw the GWP. Acceptable strings are:

- *sar* (second assessment report)
- *ar4* (fourth assessment report)
- *ar5* (fifth assessment report)
- *ar6* (sixth assessment report)

Use of the GWP column is illustrated below in Figure 14.



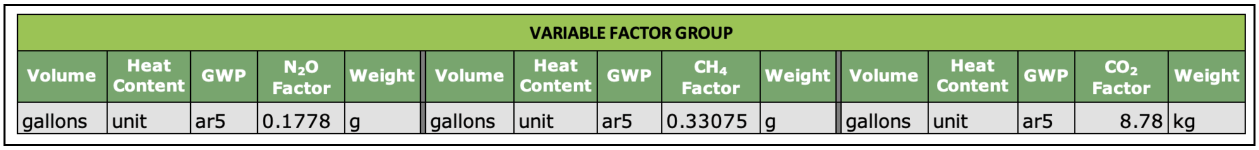
VARIABLE FACTOR GROUP				
Volume	Heat Content	GWP	CO_2 Factor	Weight
gallons	unit	1	8.78	kg

Figure 14 - Using the GWP Column to Indicate that the Factor is for a GHG

By including the GWP column to qualify the factor column for this CFS, we're telling the application that the output for this factor group is greenhouse gas. As a result, when the application converts activity to CO_2 using the factor specified it will also generate a CO_2e output. In the specific example illustrated, the GWP is '1' and there are no other GHGs that make up the CFS so the CO_2e output value will be the same as the CO_2 output value.

Multiple Component GHGs

A more complete example is the CFS illustrated below in Figure 15.



VARIABLE FACTOR GROUP														
Volume	Heat Content	GWP	N_2O Factor	Weight	Volume	Heat Content	GWP	CH_4 Factor	Weight	Volume	Heat Content	GWP	CO_2 Factor	Weight
gallons	unit	ar5	0.1778	g	gallons	unit	ar5	0.33075	g	gallons	unit	ar5	8.78	kg

Figure 15 - a Complete CFS

In this example, there are *three* factor groups for three different GHGs; one for CO_2, another for CH_4 and another for N_2O. Let's apply the five conversion steps for the simple product conversion algorithm (listed earlier in this section) to each of the three GHGs. Doing so for the same activity record illustrated in Figure 13 produces the results in table 2, below (ignoring the GWP values for the moment):

N_2O	CH_4	CO_2
2.0777 g	3.865 g	102.599 kg

Table 2 - Weights of Each GHG After Conversion

Now that we have the converted weight for each GHG, we can apply steps 1 - 4 (listed under the GWP input column description) to get their CO_2e contributions. (For reference, the *ar5* GWP values for N_2O and CH_4 are 265 and 28 respectively. The value for CO_2 is of course, as always, 1.)

FIrst, converting to *kg*, we get

- 0.002077 kg N_2O
- 0.003865 kg CH_4
- 102.599 kg CO_2

Then, multiplying by the corresponding GWPs, we get:

- 0.002077 kg x 265 = 0.55 kg
- 0.003865 kg x 28 = 0.10822 kg
- 102.599 kg x 1 = 102.599 kg

Finally, summing the values, we get 0.55 + 0.10822 + 102.599 = 103.357 kg CO_2e. The resulting converted outputs are tabulated in Table 3.

N_2O	CH_4	CO_2	CO_2e
2.0777 g	3.865 g	102.599 kg	103.357 kg

Table 3 - Final Result of Conversion Considering GWPs

Biogenic Resources

When the *Biogenic* column on a CFS is non-blank and greater than zero, then the CFS is considered biogenic.[7] A biogenic CFS that includes an emissions factor (a factor group with a GWP column) will produce a *Biogenic CO_2* output factor similar to the CO_2e factor defined earlier.

The Biogenic CO_2 factor produced will be exclusively a product of the CO_2 factor (if present) and the value in the Biogenic column. To clarify, any biogenic CO_2 contributes to the Biogenic CO_2 column *instead* of the CO_2e column (in proportion to the biogenic portion of the CFS. Other GHGs contribute exclusively to the CO_2e factor.

Revisiting the example just considered and summarized in Table 3, if the CFS in Figure 15 were 100% biogenic (as indicated by a value of 100 in its Biogenic column), then the result of the conversion would be as tabulated in Table 4.

N_2O	**CH_4**	**CO_2**	**CO_2e**	**Biogenic CO_2**
2.0777 g	3.865 g	102.599 kg	0.758 kg	102.599 kg

Table 4 - Conversion to Biogenic CO_2

In this case, the N_2O and CH_4 GHG components are multiplied by their GWPs and summed to yield the CO_2e value but the CO_2 GHG component is excluded from that sum and is directed entirely to Biogenic CO_2.

If the value for the CFS's Biogenic column were 50, then half of the CO_2 would be directed to CO_2e and the other half, to Biogenic CO_2, as tabulated in table 5.

N_2O	**CH_4**	**CO_2**	**CO_2e**	**Biogenic CO_2**
2.0777 g	3.865 g	102.599 kg	52.0575 kg	51.2995 kg

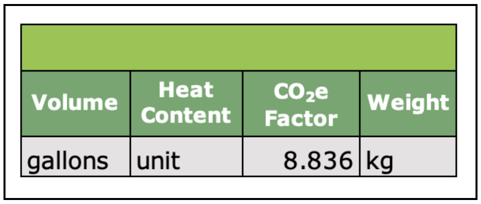
Table 5 - Results of Conversion Using a Partially Biogenic CFS

In this case, *half* of the CO_2 GHG accrued to Biogenic CO_2 and the other half accrued to CO_2e, alongside the contributions of the N_2O and CH_4 GHGs.

Specifying Factors for CO_2e or Biogenic CO_2 Directly

In certain cases, users may want to specify an emissions factor for CO_2e or for Biogenic CO_2 directly, without letting the application derive it from 'component' GHGs and their GWPs. This example is illustrated for CO_2e only in Figure 16.

[7] Note that all CFSs following a biogenic CFS (as indicated by a letter *effective date*) are also considered biogenic unless they are explicitly non-biogenic as indicated by a value of 0 in their Biogenic column.



Volume	Heat Content	CO_2e Factor	Weight
gallons	unit	8.836	kg

Figure 16 - Specifying CO_2e Directly

In previous examples CO_2e and Biogenic CO_2 are considered *derived* factors because *component* GHGs are calculated first and then CO_2e and Biogenic CO_2 are derived from the results (considering the GWPs of each component GHG and the percent biogenic.

CO_2e and Biogenic CO_2 can either be derived or not. If a user specifies a component GHG (any factor with a GWP column), then the user cannot also specify a derived factor. The application will calculate derived factors for them. Conversely, if the user specifies CO_2e or Biogenic CO_2 directly, then the user cannot specify any component GHGs. Attempts to bulkload a CFS that specifies both a component GHG and either or both of the derived factors will be rejected with an error.

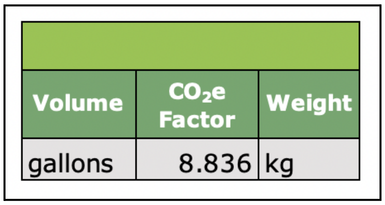
Factor

identifying keyword: *Factor*
required

Each factor group must specify exactly one output factor. If the factor group is for a GHG emissions factor, then there will be a GWP column separating the set of inputs and the factor output. If the factor group is not for a GHG emissions factor, then the output factor will be specified in the column immediately following the last input.

The factor column is identified by the keyword *Factor* in the column title. The column title must include one or more words that name the output factor *before* the keyword *Factor*. Thus, in the examples illustrated so far, the factor names used are *N_2O* or *CH_4* or *CO_2*. (In the examples in which CO_2e or Biogenic CO_2 are specified directly instead of being derived, the factor names are *CO_2e* or *Biogenic CO_2*).

The *value* in the output factor columns tell the application by how much to multiply the *input* value (or the product of multiple input values) to produce the output or the result of the conversion for the CFS.



Volume	CO_2e Factor	Weight
gallons	8.836	kg

Figure 17 - A Trivial CFS

Referring to Figure 17 above for example, the value of 8.836 in the *CO_2e Factor* column tells the application that the emissions factor quantified should be construed to mean:

8.836 kg CO_2e per gallon

WHere, the output or the result is kilograms of CO_2e. The same applies to the single factor group for the CFS illustrated in Figure 17 except that the input is the product of *volume* and *heat content* instead of being the single volume value.

Output Units

identifying keyword: varies, depending on the output unit category or handle
required

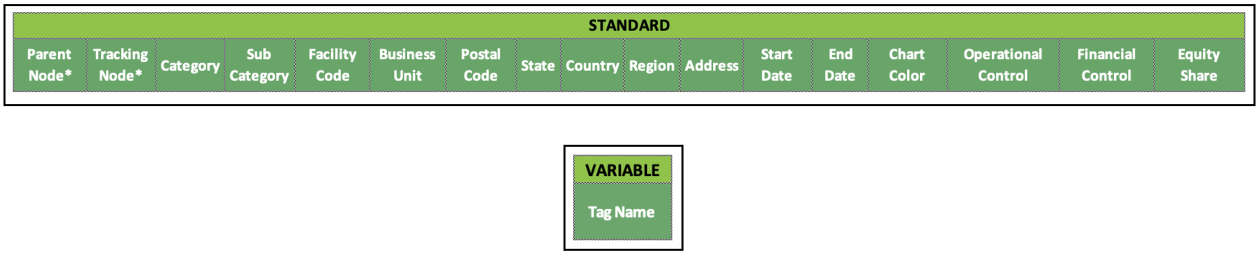
The output units column specifies the units in which the number in the factor column will express the output for the corresponding factor group. Looking back at the example in Figure 17, if the input is expressed in gallons and is multiplied by the factor 8.836, then the output will be quantified in *kg* or *kilograms*.

The *header* of the output units column must be one of the allowable unit categories indicated in Appendix A or a recognized handle.[8] It might be helpful for the reader to think of a simple single input factor group as specifying a numeric factor having a *numerator* specified in the output units and a *denominator* specified in the input units:

$$CO_2e \ = \ 8.836 \, \frac{kg}{gallon} \, x \ input \ value$$

[8] Recall that for emissions factor groups (when a GWP is specified) the output units column must specify an allowable unit in the *weight* category.

Tracking Nodes



STANDARD																
Parent Node*	Tracking Node*	Category	Sub Category	Facility Code	Business Unit	Postal Code	State	Country	Region	Address	Start Date	End Date	Chart Color	Operational Control	Financial Control	Equity Share

VARIABLE
Tag Name

Figure 18 - Columns for Loading Tracking Nodes

Standard Columns

Parent Node

identifying keyword: *Parent*
required

The first column is indicated by including the keyword *Parent* in the column header of the primary header row. The values in the content rows for this column indicate the name of the parent node in the organizational hierarchy under which the subject tracking node will be located.

Tracking Node

identifying keyword: *Node*
required

The *Tracking Node* or *Facility* column is where the new tracking node to be loaded is named. A tracking node is a *leaf* node in the organization's hierarchy, typically representing a brick and mortar *facility* but often representing a *virtual* node such as a subset of a vehicle fleet. The leaf nodes are the only nodes to which trackers can be attached, hence the term *tracking node.*

The header of this column is identified with the keyword *Node.* The cell values in the column contain the name of the node and must be unique within the account.

Category & Sub Category

identifying keyword: *Category*, *Sub*
optional
default value: blank

The columns *Category* and *Sub Category* are *taxonomy* columns. These are used to describe the type of tracking node. Most commonly, the values for the category and sub category column will be from the set of facility types enumerated by the CBECS standard (see Appendix D). These can be used as ‘selectors’ or grouping terms in queries. As an example, the selector *Educational Facility* would select all trackers under facility nodes with the category *Educational Facility*.

There are several built-in values allowed for *Category* and *Sub Category*, including the CBECS standard values. Users can define additional custom values for these attributes for their account simply by entering them in the corresponding column when loading tracking nodes. Note that for any non-standard values for these attributes, the loader will ask the user to confirm their addition to the account the first time they are encountered.

Hierarchical Relationship Between Categories and Sub Categories

A one-to-many, parent-child relationship is assumed for categories and sub categories. This relationship is defined authoritatively for the CBECS categories and sub categories, meaning that rows specifying a standard CBECS *category* must have either an empty value for the corresponding sub category or one of the *sub categories* of the parent category that is defined by CBECS.

If a standard CBECS category is specified in the category column and a sub category is specified in the same row that does not match one of the corresponding CBECS standard subcategories, that row will be rejected with an error.

Sometimes users may want to organize their facilities under categories without the specificity of a sub category. In this case, the user may simply omit the sub category column entirely or may leave the column in place but leave the corresponding cell blank in certain rows.

User Defined Categories and Sub Categories

In the previous paragraphs, we’ve focused on CBECS categories and sub categories. There are also certain built-in categories, as defined in the appendix. In certain cases, the user may want to define custom categories or sub categories. Users can do so much in the way that they can for *Resources*.

Note that similar rules relative to the hierarchical nature of categories and sub categories apply to user defined custom values as to built-in values; if a facility exists with only a *category* defined, then users will be unable to load additional facilities under that category if they have a sub category defined. Similarly, if a facility exists under a certain category and has a sub category assigned, then users will be unable to load additional facilities under that category without specifying a sub category.

Facility Code

identifying keyword: *Code*
optional
default value: blank

Users may include a *Facility Code* column to specify any coded name to be associated with a facility. Any alphanumeric characters are allowed in the coded name.

Business Unit

identifying keyword: *Business*
optional
default value: blank

Users may include a *Business Unit* column to specify a user defined business unit under which the facility operates. Users may define any value for the business unit - there are no predefined values. The set of business units configured for an account will be available as selectors and grouping attributes when executing queries in that account.

Postal Code

identifying keyword: *Postal*
optional
default value: blank

Users may include a *Postal Code* column to specify a user defined postal code under which the facility is located. Any alphanumeric string may be specified for the value. Postal codes may be used by the app to, for example, suggest an applicable eGrid electricity resource.

In the USA, 5 digits are expected to form a postal code. In the UK a six character combination of letters and numbers will generally be used. However, these should be interpreted only as guidelines; any alphanumeric string will be accepted as postal codes.

State

identifying keyword: *State*
optional
default value: blank

Users may include a *State* column to specify the geographical state or province under which the facility is located. Depending on any values specified in the *Country* or *Postal Code* column, the allowable values for *State* may be restricted. For example, if the country column specifies *USA*, then one of the fifty states must be chosen for the state column. If the country column specifies *Canada* then one of the Canadian provindes must be specified. The restriction to specific states exists only for a subset of country values. For most countries, users may name any state.

Users may *name* the state verbatim or may specify a state using acceptable codes,, such as *WA* for *Washington State*. See appendix F for recognized codes for different states.

Country

identifying keyword: *Country*

optional
default value: blank

Users may include a *Country* column to specify the country in which the facility is located. Depending on any values specified in the *Country* column, the allowable values for *State* or *Postal Code* may be restricted. For example, if the state column specifies *Vermont*, then *USA* will be expected for the country column. The set of countries configured for an account will be available as selectors and grouping attributes when executing queries in that account.

Users may *name* the country using either ISO recognized full country names or the standard two (alpha-2) or three (alpha-3) letter ISO country codes. Any other values for the *Country* column will be rejected with an error. See appendix F for recognized codes for different countries.

Region

identifying keyword: *Region*
optional
default value: blank

Users may include a *Region* column to specify a region in which the facility is located. Typically, these will be geographical *continents* or business regions such as *EMAIR* or *LATAM.* However, users are free to name any regions they wish (alpha characters only). The set of regions configured for an account will be available as selectors and grouping attributes when executing queries in that account.

Address

identifying keyword: *Address*
optional
default value: blank

Users may include an *Address* column to specify the address of a facility. Address fields may contain any combination of numbers, letters, spaces and punctuation up to 250 characters in length. The address field is a *text field* formatted by row. Formatting will be preserved from te bulkloader.

Start Date

identifying keyword: *Start*
optional
default value: blank

This column may specify the date on which the facility was brought online. The following formats are acceptable:

- *mm/dd/yyyy*
- *mm/dd/yy*

End Date

identifying keyword: *End*
optional
default value: blank

This column may specify the date on which a facility was retired.

Date values in this column should be formatted using the same format as defined for the *Start Date* column.

Chart Color

identifying keyword: *Color*
optional
default value: no color

Users may include a *Chart Color* column to specify the color assigned to the tracking node as represented in certain charts that are grouped by tracking node (e.g. the corresponding column, pie slice or line). The color must be specified using the six-digit hex code for the color.

Operational Control

identifying keyword: *Operational*
optional
default value: yes

This optional column can be used to indicate that a particular tracking node and all the trackers under that node should be considered to be under operational control. Omitting the column or the value for a particular tracking node will be assumed to indicate that all trackers under the node are under operational control. Allowable values include *yes*, *no* or *blank*.

Financial Control

identifying keyword: *Financial*
optional
default value: no

This optional column can be used to indicate that a particular tracking node and all the trackers under that node should be considered to be under financial control. Omitting the column or the value for a particular tracking node will be assumed to indicate that all trackers under the node are not under financial control. Allowable values include *yes*, *no* or *blank*.

Equity Share

identifying keyword: *Share*

optional
default value: 100

This optional column can be used to specify an equity share percent for the corresponding facility. The percent value will be used to prorate all queried metrics for trackers under the facility. Any value in the equity share column must be an integer between 0 and 100.

Variable Columns

After the standard columns, Users may include an arbitrary number of columns indicating tags.

Tag Name

identifying keyword: *Tag*
optional
default value: blank

Any number of tag columns may be included by specifying the keyword *Tag* in the column header. Content rows in each *tag* column can then include the name of a child or parent tag that will be associated with the tracking node and, by extension, all the trackers under that node.

Content in any non-blank *tag* column must match one of the tags configured for the account or else the row will be rejected with an error.

When a child tag is specified, the parent tag will also be automatically attached to the tracker. Users can attach only a single child tag to any tracker for any tag family. Attempts to attach a different child tag for a tracker already tagged with a child tag in the same family will be rejected with an error.

Appendix A - Allowable Units

The following tables list all the units recognized by the application, by unit type. Users can use the *coded* value (from the right column) or the *display name* (from the left column) when filling out bulk loaders. These are case insensitive.

Area Units	
acre	acre
hectare	ha
millions square feet	mmsqfeet
thousands square feet	msqfeet
square foot	sq ft
square kilometer	km^2
square meter	m^2
square mile	sq mi

Count Units	
count	count
dozen	dozen
gross	gross
hundreds	hundreds
hundredths	hundredths
millions	millions
percent	percent
tens	tens
tenths	tenths
thousands	thousands
thousandths	thousandths
unit	unit

Currency Units	
dollar	usd

Distance Units	
foot	ft
thousand km	kilokilometer
thousand miles	kilomile

Distance Units (cont'd from previous column)	
kilometer	km
thousand km	kilokilometer
million km	megakilometer
thousand miles	kilomile
million miles	megamile
meter	m
metre	m
mile	mi
foot	ft
yard	yd

Energy Units	
Btu	Btu
giga-Btu	GBtu
gigawatt hour	GWh
kilo-Btu	MBtu
mega-Btu	MMBtu
megawatt hour	MWh
decatherm	Dth
gigajoule	GJ
kilo-Btu	kBtu
kilowatt hour	kWh
kilojoule	kJ
megajoule	MJ
standard cubic foot (gas)	scf
terajoule	TJ
therm	thm

Freight Units	
ton-mile	ton_mile
tonne-kilometer	tonne_km

Gas Units	
sccf	100s std cubic feet
Gas Units (cont'd from previous page)	
sccm	100s std cubic meter
scf	std cubic feet
scm	std cubic meter
scy	std cubic yard
smcf	1000s std cubic feet
smcm	1000 cubic meter
smmcf	millions std cubic feet
smmcm	millions std cubic meter

Time Units	
day	day
hour	hour
microsecond	microsecond
milisecond	milisecond
minute	minute
nanosecond	nanosecond
second	second
week	week

Volume Units	
af	acre foot
100 cubic foot	ccf
1000 cubic foot	mcf
million cubic foot	mmcf
ccm	100 cubic meter
cubic foot	ft^3
cm	cubic meter
cy	cubic yard
dry_gal_us	dry gallon
fbm	board foot
kiloliter	kiloliter
liq_bbl	barrel
liq_cgal_imp	hundred Imperial gallons
liq_gal_imp	Imperial gallon
liq_gal_us	gallon
liq_kilo_gal_us	thousand gallon
liq_mbbl	thousand barrels
liq_mega_gal_us	million gallon
liter	liter
Volume Units (cont'd from previous column)	
mcm	1000 cubic meter
megaliter	million liters
mfbm	1000 board foot
mililiter	mililiter
mmcm	million cubic meter
oil_bbl	oil barrel
oil_mbbl	thousand oil barrels

Weight Units	
Klbs	Klbs
Mlbs	Mlbs
gram	gram
kg	kilogram
lbs	pound
long_ton	long ton
ounce	ounce
short_ton	short ton
ton	ton
tonne	tonne

Appendix B - Loading Documents

As discussed in the section on activity records, we recognize that it is necessary to enable documents to be loaded to the Scope 5 application in order to support or substantiate activity records. We've considered various mechanisms for automating the loading of such documents, potentially as part of the bulkloading process, but have concluded that there each of these mechanisms suffers from deficiencies that render them impractical. Instead, we will support an evolution of functionality that will facilitate loading supporting documents after records have been loaded via the bulkloader.

What About Trackers that Require Attachments?

Certain trackers can be configured to require that documents be loaded with each activity record (see the *Attachment Required* attribute on trackers). Given that it's not possible to bulkload documents with activity records, what does this mean with respect to bulkloading activity records to these trackers? Will any attempt to bulkload activity records to trackers that require attachments be rejected?

No. It will be possible to bulkload activity records to trackers that require attachments. However, these records will be marked as *requiring an attachment*. Warnings will be generated for trackers requiring attachments that have records without attachments. Users will be able to configure an 'exclusion threshold'. Trackers requiring attachments for which a certain percent of their activity records do not have attachments will be excluded.

Creating records for these trackers manually will require that the user load supporting documents at the time each record is created.

Appendix C - Supported Conversion Algorithms