TRIM.FaTE USER'S GUIDE
MODULE 4: ADDING NEW COMPONENTS TO A LIBRARY[1]

While planning or working with a TRIM.FaTE application, users may find they want to expand a TRIM.FaTE library beyond its current objects (e.g., compartment types, algorithms, chemicals, sources). TRIM.FaTE is designed to be flexible enough to allow users to add new objects or modify existing objects (e.g., add properties) relatively easily. This flexibility allows TRIM.FaTE to be easily adapted to different applications, refined in various user-desired ways, and updated with new information or mass transfer algorithms (e.g., from the literature).

This module describes the process for adding new chemicals, properties, property types, algorithms, compartment types, and composite compartments to a TRIM.FaTE library.[2] The steps involved in adding new sources to a library are described in Module 5, Source Characterization. This module is divided into the following sections.

> **LIBRARY COMPONENTS**
>
> Libraries are comprised of the following types of objects:
>
> - algorithms;
> - chemicals;
> - compartment types;
> - composite compartment types;
> - sources; and
> - property types.
>
> All of these types of objects, with the exception of property types, have associated properties.
>
> **Properties** are used to describe the different characteristics of these TRIM.FaTE objects and can be added to new or existing objects as needed.
>
> **Property types** are the templates from which properties are created.

Section 1.	***Adding Chemicals***	This section describes how chemicals are added to a library.
Section 2.	***Adding Properties***	This section describes how properties are added to existing objects in a library.
Section 3.	***Adding Property Types***	This section describes how property types are added to a library.
Section 4.	***Adding Algorithms***	This section describes how algorithms are added to a library.

---

[1]Descriptions of library-specific algorithms and properties presented in this module pertain to the July 2005 version of the TRIM.FaTE Public Reference Library.

[2]While new properties can be added to objects within existing TRIM.FaTE scenarios, this process is complicated by the existence of links and multiple compartments of given compartment types. Additionally, compartments and composite compartments (as well as new properties) can be added to a scenario from a referenced library via scenario import files as described in Section 1.2 of Module 3, Library and Scenario Data Files.

Section 5.   *Adding Compartment Types*   This section describes how compartment types are added to a library.

Section 6.   *Adding Composite Compartments*   This section describes how composite compartments are added to a library.

In addition, Appendix A provides a detailed description of the formula syntax used in TRIM.FaTE. More detailed descriptions of algorithms, compartments, compartment types, composite compartments, and chemicals are provided in Chapter 3 of Volume I of the TRIM.FaTE Technical Support Document (EPA 2002).

1. ADDING CHEMICALS

New chemicals can be added to a TRIM.FaTE library as needed for different scenarios. To add a new chemical, the user will need to provide information for properties required by TRIM.FaTE (see Table 1), as well as additional information required, directly or indirectly,[3] by algorithms that will describe the transfer or transformation of that chemical in a simulation.[4]

Table 1
Required Chemical Properties

Property Name	Data Type	Description
CAS	Text	The Chemical Abstract Service number for the chemical
category	Category	The category of the chemical (e.g., Organic	Benzo(a)pyrene)
doesTransform	True/False	Indicates whether the chemical transforms to another chemical to be modeled (chemicals modeled as degrading, with the degraded mass tracked in a sink rather than modeled as a transformation product, should be set to "false")
enabled	True/False	Indicates whether the chemical should be used
molecularWeight	Real Number	The molecular weight of the chemical

Prior to adding chemicals to the library, the user will need to review the available algorithms in the library to determine if additional or different algorithms are required to model transport and transformation for this chemical. With this step, the user has responsibility for the appropriateness and scientific defensibility of the algorithms which will be used to transport and/or transform the chemical in their scenario. Any new algorithms required for the new chemical can be added using the steps described in Section 4.

---

[3]Chemical properties may be referenced directly by algorithm properties (e.g., *transferFactor*) or by compartment properties, which are then referenced by algorithm properties.

[4] Note that many chemical-specific properties are actually defined as properties of a compartment, not properties of a chemical; only chemical-specific properties that are constant across all compartments can be defined as chemical properties.

There are three different ways to include a new chemical in a TRIM.FaTE library. The first method involves creating a new chemical "from scratch" (i.e., without using information from an existing chemical). This is the most common method for creating chemicals that are not modeled similarly to any existing chemicals. To include a new chemical using this method, the user should follow these steps.

(1)   Create a new Chemical in the Chemicals portion of the Library.
   A.   Open the TRIM.FaTE library.
   B.   Select "Chemical" from the "Contents" pull-down menu in the object browser.
   C.   Click the "New" button at the bottom of the list of currently included chemicals.
   D.   Specify a name for the new Chemical in the pop-up window and click "OK."
(2)   Using the Property Editor, specify values for the automatically created (required) properties of the new Chemical: *CAS, category, doesTransform* (default is "false"), *enabled* (default is "true"), and *molecularWeight*.
(3)   Add to the newly added Chemical those chemical properties referenced directly or indirectly by the algorithms applied for the new chemical, and edit the value of the properties to reflect characteristics of the new chemical. The steps involved in adding new properties to a Chemical are described in Section 2.
(4)   Add to each Compartment those properties whose values vary by chemical and which are referenced (directly and indirectly) by the algorithms associated with the new chemical. The steps involved in adding new properties to a Compartment are described in Section 2. Given the number of chemical-specific compartment properties typically included in a library, users may choose to use a Library Object Import file (as described in Module 3) for this. However, regardless of the selected method, this can be a very time-consuming process and requires the user to ensure the inclusion of all of the necessary chemical-specific compartment properties.

The second method involves duplicating an existing chemical and editing the duplicate as needed. This method can be used when the new chemical is very similar to an existing chemical. To create a new chemical using this method, the user should follow these steps.

(1)   Duplicate an existing Chemical in the Chemicals portion of the Library.
   A.   Open the TRIM.FaTE library.
   B.   Select "Chemical" from "Contents" pull-down menu in the object browser.
   C.   Select similar chemical from list of currently included chemicals and select "Duplicate" button at bottom of list.
   D.   Specify a name for the newly created Chemical in the pop-up window and click "OK."
(2)   Using the Property Editor, edit, as necessary, the required properties of the new Chemical: *CAS, category, doesTransform* (default is "false"), *enabled* (default is "true"), and *molecularWeight*.

(3)   Using the Property Editor, remove from the new Chemical any properties not needed, directly or indirectly, by algorithms applied for this chemical. This is done by selecting the property from the property table for the new Chemical and clicking on the "Del" button. It is noted that the presence of unused properties will not affect execution of a simulation, but their presence may provide later confusion regarding the chemical characteristics on which the transport and transformation algorithms rely.

(4)   Using the Property Editor, edit the values for the remaining properties of the new Chemical to reflect the characteristics of the new chemical. If the duplicated chemical is very similar to the new chemical, this step may only require editing the values of some of the properties.

(5)   Add to the newly added Chemical those chemical properties referenced directly or indirectly by the algorithms applied for the new chemical and edit the values of these properties to reflect the characteristics of the new chemical. The steps involved in adding new properties to a chemical are described in Section 2. If the duplicated chemical is modeled very similarly to the new chemical, this step may not be required.

(6)   Add to each Compartment those properties whose values vary by chemical and which are referenced (directly and indirectly) by the algorithms associated with the new chemical. The steps involved in adding new properties to a Compartment are described in Section 2. Given the number of chemical-specific compartment properties typically included in a library, users may choose to use a Library Object Import file (as described in Module 3) for this. However, regardless of the selected method, this can be a very time-consuming process and requires the user to ensure the inclusion of all of the necessary chemical-specific compartment properties.

The third method for including new chemicals in a library involves using a Library Object Import file. To use this method, the user should follow these steps.

(1)   Create a text file in the appropriate Library Object Import file format (described in Module 3) containing the new Chemical, the chemical properties referenced (directly or indirectly) by the algorithms applied for the new chemical, and the compartment properties whose values vary by chemical and which are referenced (directly or indirectly) by the algorithms applied for the new chemical.

(2)   Open the TRIM.FaTE library.

(3)   Select "Import" from the "File" menu in the Library Window.

(4)   Choose "Object Importer" in Choose Importer pop-up window and click "OK."

(5)   Select the Library Object Import file using the File Browser and click "Import TRIM Objects."

2. ADDING PROPERTIES

Properties are associated with almost every type of object in TRIM.FaTE.[5] Different types of properties can be used for a number of different purposes, including indicating the chemical to which an algorithm applies or defining the value of a property used in a calculation. There are a number of situations in which a user may find it necessary to add properties to TRIM.FaTE objects. As a relatively simple example, a user may want to add macrophytes to the diet of bald eagles. Provided there is an algorithm describing the ingestion of macrophytes by bald eagles, the user would need to add the property *FractionDietMacrophyte* to the bald eagle compartment. Examples of other situations where a user may need to add properties to existing objects include when adding new compartments or algorithms to a library, or when changing formula properties for an existing object.

Similarly, new properties can be added to objects in a TRIM.FaTE scenario, as well as to objects in a library. This section, however, is focused on the steps associated with adding properties to objects in a library.

It is important to note that in order to add a new property to an object, the user must first confirm that that type of property exists among the property types in the library. If the property is already used by an existing object, the associated property type exists in the library and the property can be added to another object. Users can view the complete list of property types in the library by selecting "Property Types" from the "Contents" pull-down menu in the library object browser. If an appropriate property type is identified in this list, the user can simply add the property to an object as described below. Otherwise, the user must first add a new property type to the library as described in Section 3.

Provided the relevant property type exists in the library, the user can add a new property to a library object by following these steps.

(1)   Open the TRIM.FaTE library.
(2)   Select the object type to which the property will be added from the "Contents" pull-down menu in the object browser.
(3)   Select the object to which the property will be added from the list of existing objects of the selected type and click the "Properties" button below the list. The Property Editor showing the property table for the selected object will appear on the right side of the window.
(4)   Click the "New" button in the Property Editor.
(5)   Select the property type for the new property from the pop-up window and click "OK." If the appropriate property type is not included in the list, add it using the steps described in Section 3. Users can add multiple properties of different types to the object by selecting each property type while holding down the "Cntrl" key.
(6)   Select the chemical to which the property (or properties) will apply and click

---

[5]Objects with which properties are associated include: algorithms, chemicals, compartments, composite compartments, volume elements, links, scenarios, and sources. The only type of TRIM.FaTE objects that do not have associated properties are property types. All types of objects except links, volume elements, and scenarios can be stored in a library.

"OK." If the property is not chemical-specific, select "None." Users can add multiple chemical-specific properties of each selected property type by selecting each chemical while holding down the "Cntrl" key.

Alternatively, new properties can be added to library objects using a Library Object Import file. To use this method, the user should follow these steps.

(1)   Create a text file in the appropriate Library Object Import file format (described in Module 3) containing the new property and its value.
(2)   Open the TRIM.FaTE library.
(3)   Select "Import" from the "File" menu in the Library Window.
(4)   Choose "Object Importer" in Choose Importer pop-up window and click "OK."
(5)   Select the Library Object Import file using the File Browser and click "Import TRIM Objects."

After a property has been added to an object, the user can provide a value for the property. If the property type associated with the property includes a default value, the value for the property will be initially assigned this default. If there is no default value for the property type, the value of the property will appear as "<Unset>." Users can assign new values to properties regardless of whether the property has been assigned a default value. For properties with data types of "Real Number" (see Section 3 for a description of the different data types), the property value can be specified in one of the following four different ways, referred to as "forms."

•   Constants specified directly by users (referred to as "Constant Real Numbers").
•   Sequence of values with associated dates and time specified directly by the user (referred to as "Unevenly Time-stepped Real Numbers").
•   Sequence of values with associated dates and times specified in an external data file (referred to as "Unevenly Time-stepped Real Numbers from File ").
•   Values calculated based on a user-supplied formula (referred to as "Formula Real Numbers"). The syntax for these user-supplied formulas is explained in Appendix A.

To specify a form for a "Real Number," the user should follow these steps.

(1)   Open the TRIM.FaTE library.
(2)   Select the type of object to which the property will be added from the "Contents" pull-down menu in the object browser.
(3)   Select the object to which the property will be added from the list of existing objects and click the "Properties" button below the list.
(4)   Select a property with "Real Number" in the "Data" column of the Property Editor and click the "Form" button in the row of buttons at the top of the Property Editor.
(5)   Select the form of the property value in the "Change Property Form" pop-up window and click "OK." The value of the selected value can then be specified in the selected form.

3. ADDING PROPERTY TYPES

As mentioned in the previous section, before a property can be added to an object (e.g., compartment, chemical) the associated property type must already exist in the library. Property types can be added to a TRIM.FaTE library when a user needs to add a property to an object and none of the currently included property types are appropriate. There are four different ways to add a new property type to a TRIM.FaTE library.

The first method involves creating a new property type "from scratch" (i.e., without using information from an existing property type). This is the most common method and can be used for creating any property type. To create a new property type using this method, the user should follow these steps.

(1)   Open the TRIM.FaTE library.
(2)   Select "Property Types" from the "Contents" pull-down menu in the object browser.
(3)   Click the "New" button at the bottom of the list of currently included property types.
(4)   Specify a name for the property type in the pop-up window and click "OK".
(5)   From the Property Type window, select the "Type of Data" (see Table 2 for descriptions of the different types of data).
(6)   Based on the type of data selected in Step (5), different fields remain for completion in the Property Type window. For "Real Number" and "Integer" types of data, only the units field is required as the minimum documentation associated with a property type. The units field is informational only and is not currently used by TRIM.FaTE. Users are urged to also complete the description field for all types of data to facilitate understanding of the new property type at a later point in time.
(7)   Click "Accept" button.

Table 2
Data Types in TRIM.FaTE

"Type of Data"	Description
Real Number	Properties of this data type can be defined using one of the following formats (or forms, in TRIM.FaTE): *constant real numbers* (e.g., 0.056, 4.5E-10), *unevenly time-stepped real numbers* (a sequence of real numbers with associated dates and times), or *formula real numbers* (mathematical formulas that can include references to other properties). The syntax used for specifying values for formula real number properties is provided in Appendix A.
Integer	Properties of this data type must be defined as integers (e.g., 10, 546)
True or False	Properties of this data type must be defined as either "true" or "false"
Text	Properties of this data type must be defined as alphanumeric text (e.g., Mercury, benchmarks1).

"Type of Data"	Description
Category	Properties of this data type are defined using a hierarchical classification for objects ordered from most general to most specific, with each level separated by a vertical bar ("|"). Each algorithm, chemical, and compartment is assigned a category. Chemical and compartment categories are used in algorithms to specify the types of compartments and chemicals to which the algorithm should apply. Users can specify more generalized categories in order to apply an algorithm to a group of chemicals and/or compartments. For example, an algorithm with the chemicalCategory "Mercury" will apply to chemicals with the categories "Mercury | Divalent Mercury," "Mercury | Elemental Mercury," and "Mercury | MethylMercury." Users can specify that an algorithm applies to all chemicals or compartments in the library using the "All" category.
Date and Time	Properties of this data type are used to specify time during a model simulation (e.g., starting time, ending time, time when an input value changes). Properties of this data type should be specified in the following format: MM/DD/YYYY HH:MM:SS TimeZone For example, 12/10/2002 05:00:00 EST.

The second method involves duplicating an existing property type and editing the duplicate as needed. This method can be used when the new property type has the same data type as an existing property type and may be useful for creating property types that are very similar to an existing property type. To create a new property type using this method, the user should follow these steps.

(1)   Open the TRIM.FaTE library.
(2)   Select "Property Types" from the "Contents" pull-down menu in the object browser.
(3)   Select similar property type (must have the same "Type of Data" as the new property type) from list of currently included property types.
(4)   Click the "Duplicate" button at bottom of list of currently included property types.
(5)   Specify a name for the new property type in the pop-up window and click "OK."
(6)   Edit the values of the attributes (i.e., units, description, minimum value, maximum value, and default value) in the Property Type window as needed.
(7)   Click "Accept" button in the Property Type window.

The third method for including new property types in a library involves using a Library Object Import file. To use this method, the user should follow these steps.

(1)   Create a text file in the appropriate Library Object Import file format (described in Module 3) containing the new property type and its associated attributes.
(2)   Open the TRIM.FaTE library.
(3)   Select "Import" from the "File" menu in the Library Window.
(4)   Choose "Object Importer" in Choose Importer pop-up window and click "OK."
(5)   Select the Library Object Import file using the File Browser and click "Import TRIM Objects.

The fourth method for including new property types in a library involves using a Library Property Type Spreadsheet Import file. To use this method, the user should follow these steps.

(1)   Create a text file in the appropriate Library Property Type Spreadsheet Import file format (described in Module 3) containing the new property type and its associated attributes.

(2)   Open the TRIM.FaTE library.

(3)   Select "Import" from the "File" menu in the Library Window.

(4)   Choose "Property Type Spreadsheet Importer" in Choose Importer pop-up window and click "OK."

(5)   Select the Library Object Import file using the File Browser and click "Import."

4.   ADDING ALGORITHMS

Algorithms are used to mathematically represent the transfer of chemical mass between compartments and the transformation of chemical mass within compartments.[6] Algorithms are, in essence, the formulas used by the model to describe these transfer and transformation processes. Users can add new algorithms to the library in order to model processes not included in the existing algorithms or to provide alternative methods for modeling a process described by an existing algorithm.

In order for newly added algorithms to be used in a TRIM.FaTE scenario, all properties required by algorithms must exist in objects included in the scenario. These properties can be properties of the algorithm or of the associated compartments, chemicals, volume elements, sources, links, or scenario. When adding new algorithms to a library, any newly required properties of existing objects (i.e., compartments, chemicals, and sources) will also need to be added to the library. The following text box describes considerations regarding the type of object (e.g., chemical, compartment) to which the properties required by an algorithm should be added.

---

[6]Note that other equations used in the model, such as those used in formula properties, are not referred to as algorithms in TRIM.FaTE. Rather, Algorithms refers to those TRIM.FaTE objects which include among their properties a transfer factor which is the value (usually in formula form) in units of inverse time that describes the instantaneous flux of modeled chemical per amount of the chemical in the sending compartment.

WHERE SHOULD PROPERTIES REFERENCED BY ALGORITHMS BE ADDED?

Properties referenced by algorithms can be included as properties of the algorithm, or as properties of other TRIM.FaTE objects such as compartments or chemicals. The following are guidelines for determining where to add these properties.

- Properties referenced by an algorithm that are associated with a particular type or types of compartment (e.g., water clarity) should be added to those compartment types.

- Properties referenced by an algorithm that are associated with a single chemical and do not vary with compartment type (e.g., octanol water partition coefficient) should be added to that chemical.

- Properties that are used to simplify (e.g., provide intermediate calculations for) the algorithm property *transferFactor* should be added to the algorithm. The following example demonstrates how this can be done.

  Original *transferFactor* formula: (Algorithm.ArealErosionRate / SendingCompartment.rho) *(SendingCompartment.Chemical.FractionMass_Sorbed/SendingCompartment.VolumeFraction_ Solid)

  New algorithm property called *SolidArealPhaseVelocity*: Algorithm.ArealErosionRate / SendingCompartment.rho

  Revised *transferFactor* formula: Algorithm.SolidArealPhaseVelocity * (SendingCompartment.Chemical.FractionMass_Sorbed/SendingCompartment.VolumeFraction_ Solid)

Scenario, volume element, source, or link properties can also be referenced by an algorithm property; however, these properties cannot be added in a library because scenario, volume element, source and link objects do not exist in the library. These properties can be added directly to a scenario, or, more commonly, via Scenario Import files. After performing a "Verify," any scenario, volume element, and link properties referenced in an enabled algorithm will be added automatically to the relevant objects in the scenario and assigned "<Unset>" values (see Module 2 for more details about the "Verify" function).

In addition to any properties that may be added to the algorithm to facilitate calculation of the *transferFactor*, there are 12 other algorithm properties required by TRIM.FaTE (see Table 4).[7] Users must specify values for all of these properties, with the exception of *chemicalCategory*, *receivingChemicalName*, and *sendingChemicalName*, for which values need to be specified for only one or two of the three properties (as described below).

- For algorithms involving two chemicals (i.e., transformation algorithms), users must specify values for *sendingChemicalName* and *receivingChemicalName* and leave *chemicalCategory* "<Unset>."

---

[7]Users can optionally add properties to an algorithm to simplify the transferFactor equation (see text box).

- For algorithms involving one chemical (i.e., transfer algorithms), users have the option of specifying values for either *chemicalCategory* or both *sendingChemicalName* and *receivingChemicalName*.

If values are specified for all three properties, the *chemicalCategory* property will override the *sendingChemicalName* and *receivingChemicalName* properties when the algorithm is applied. For instructions on applying a single transfer algorithm to multiple chemicals, refer to the text box below.

APPLYING A TRANSFER ALGORITHM TO MULTIPLE CHEMICALS

There are several ways for the user to use a transfer algorithm for more than one chemical.

- The user can make multiple copies of the algorithm and apply each copy to a specific chemical by specifying values corresponding to that chemical for either *chemicalCategory* or both *sendingChemicalName* and *receivingChemicalName*.

- The user can apply a single algorithm to multiple chemicals using the *chemicalCategory* property. In this case, the algorithm would apply to all chemicals that match the specified *chemicalCategory*.

- When the user wants to apply a single algorithm to several different groups of chemicals, they can employ a combination of the first two options. In this case, the user would make a copy of the algorithm for each group of chemicals to which the algorithm will apply. Then, the *chemicalCategory* property for each algorithm would be set to the *Category* corresponding to a group of chemicals.

There are three different ways to add an algorithm to a TRIM.FaTE library. The first method involves creating a new algorithm "from scratch" (i.e., without using information from an existing algorithm). This is the most common method for creating algorithms to represent processes not currently included in the library. To create a new algorithm using this method, the user should follow these steps.

(1)   Open the TRIM.FaTE library.
(2)   Confirm that the compartment(s) and chemical(s) to which the algorithm will apply are included in the TRIM.FaTE library. If any chemicals or compartments are missing from the library, follow the steps described in Sections 1 and 5, respectively, to add them to the library prior to proceeding with the addition of the algorithm.
(3)   Select "Algorithms" from the "Contents" pull-down menu in the object browser.
(4)   Click the "New" button at the bottom of the list of currently included algorithms.
(5)   Specify a name for the new algorithm in the pop-up window and click "OK."
(6)   Using the Property Editor, specify values for the following algorithm properties: *category*, *chemicalCategory* -or- *sendingChemicalName* and *receivingChemicalName*, *mate* (optional), *receivingCompartmentCategory*, and *sendingCompartmentCategory* (see Table 4).
(7)   Using the Property Editor, edit the default values of the following algorithm properties as necessary: *doesTransportChemical* (default is "true"),

---

doesTransformChemical (default is "false"), *enabled* (default is "true"), and *isDefaultForCategory* (default is "false") (see Table 4).

(8)   Using the Property Editor, specify value for the *transferFactor* property. This property is typically defined by a formula that references properties of other objects (e.g., chemicals, compartments). The syntax for these formulas is described in Appendix A.

(9)   If properties for existing chemicals and compartments needed by *transferFactor* are not included in the library, the user should add these properties to the appropriate objects (this may involve adding new property types, which is described in Section 3).

The second method involves duplicating an existing algorithm and editing the duplicate as needed. This method can be used when the new algorithm is very similar to an existing algorithm. To create a new algorithm using this method, the user should follow these steps.

(1)   Open the TRIM.FaTE library.

(2)   Confirm that the compartment(s) and chemical(s) to which the algorithm will apply are included in the TRIM.FaTE library. If any chemicals or compartments are missing from the library, follow the steps described in Sections 1 and 5, respectively, to add them to the library prior to proceeding with the addition of the algorithm.

(3)   Select "Algorithms" from the "Contents" pull-down menu in the object browser.

(4)   Select a similar algorithm from list of currently included algorithms and click "Duplicate" button at bottom of list

(5)   Specify a name for the new algorithm in the pop-up window and click "OK."

(6)   Using the Property Editor, edit the values of the following algorithm properties as necessary: *category*, *chemicalCategory* -or- *sendingChemicalName* and *receivingChemicalName*, *doesTransportChemical*, *doesTransformChemical*, *enabled*, *isDefaultForCategory, mate* (optional), *receivingCompartmentCategory*, and *sendingCompartmentCategory* (see Table 4).

(7)   Using the Property Editor, edit the value for the *transferFactor* property. This property is typically defined by a formula that references properties of other objects (e.g., chemicals, compartments). The syntax for these formulas is described in Appendix A.

(8)   If properties for existing chemicals and compartments needed by *transferFactor* are not included in the library, the user should add these properties to the appropriate objects (this may involve adding new property types, which is described in Section 3).

The third method for including new algorithms in a library involves using a Library Object Import file. To use this method, the user should follow these steps.

(1)   Create a text file in the appropriate Library Object Import file format (described in Module 3) containing the new Algorithm, the required algorithm properties and associated values, and any additional algorithm properties (and their associated values) required to calculate the *transferFactor*.

(2)   Open the TRIM.FaTE library.

(3)   Select "Import" from the "File" menu in the Library Window.
(4)   Choose "Object Importer" in Choose Importer pop-up window and click "OK."
(5)   Select the Library Object Import file using the File Browser and click "Import TRIM Objects.

Table 4
Algorithm Properties

Property Name	Type of Data	Required/ Optional	Description/Notes
category	Category	Required	The category of the algorithm, which is used to identify the type of process represented by the algorithm (e.g., Transformation, Advection | Horizontal). If a new algorithm represents a process not included in the existing algorithms, the user can set it to a new category name. If the new algorithm represents an alternative to an existing algorithm, the user should use the category of the existing algorithm.
chemicalCategory	Category	Optional[a]	The chemical category to which the algorithm applies. This property can be used for transport algorithms to apply the algorithm to all chemicals of the specified category, thus reducing the number of algorithms included in the library. The user must specify either a *chemicalCategory* or a *receivingChemicalName* and *sendingChemicalName* for each algorithm. If *chemicalCategory* is specified, the model ignores the values for the *receivingChemicalName* and *sendingChemicalName* properties. For example, an algorithm with the *chemicalCategory* of "Mercury" would apply to chemicals with *Category* values of "Mercury | Divalent Mercury" and "Mercury | Elemental Mercury" but not to chemicals with *Category* values of "Organic | Benzo(a)pyrene."
doesTransformChemical	True/False	Required	Indicates whether the algorithm represents a transformation process. If this property is true, *doesTransportChemical* must be false, and vice versa.
doesTransportChemical	True/False	Required	Indicates whether the algorithm represents a transport process. If this property is true, *doesTransformChemical* must be false, and vice versa.

Property Name	Type of Data	Required/ Optional	Description/Notes
enabled	True/False	Required	Indicates whether the algorithm should be used. This property is typically used to indicate which algorithm should be used when there is more than one algorithm for a given process, or if there is a desire to shut off a particular algorithm for any reason (e.g., testing).
isDefaultForCategory	True/False	Required	Indicates whether the algorithm is the default for the algorithm category when there is more than one algorithm for the category.
mate	Text	Optional	The name of the algorithm that should be used on reciprocal links.
receivingChemicalName	Text	Optional[a]	The name of the receiving chemical. For transformation algorithms, this should be set to the product of the transformation. For transport algorithms, this should be set to the same chemical as the *sendingChemicalName*. This property must be specified if the *chemicalCategory* is not specified.
receivingCompartmentCategory	Category	Required	The compartment category (e.g., Abiotic | Surface water, Bird | Mallard) of the compartment that is receiving chemical mass. For transformation algorithms, this should be the same as the *sendingCompartmentCategory*.
sendingChemicalName	Text	Optional[a]	The name of the sending chemical. For transformation algorithms, this should be set to the chemical that is being transformed into another chemical. For transport algorithms, this should be set to the same chemical as the *receivingChemicalName*. This property must be specified if the *chemicalCategory* is not specified.
sendingCompartmentCategory	Category	Required	The compartment category of the compartment that is sending chemical mass. For transformation algorithms, this should be the same as the *receivingCompartmentCategory*.
transferFactor	Real Number	Required	The value used to mathematically represent the transfer or transformation described by the algorithm. The *transferFactor* is typically defined as a formula that references properties of other objects (e.g., compartments, chemicals)

[a] The user must provide values for either *chemicalCategory* or *sendingChemicalName* and *receivingChemicalName*. If values are provided for all three properties, the algorithm will use the *chemicalCategory* value and ignores the values for *sendingChemicalName* and *receivingChemicalName*.

5. ADDING COMPARTMENT TYPES

Compartment types may be added to a TRIM.FaTE library to allow a user to model abiotic media and biotic species not included in the existing library. For example, a user may want to model a river surface water compartment type in addition to the lake surface water compartment type in the existing library. When a compartment type is added to a library, there are five properties (described in Table 5) that must be included among the properties associated with the compartment type. Although the additional properties associated with a given compartment type vary, these five properties are associated with every compartment type.

Table 5
Required Compartment Type Properties

Property Name	Data Type	Description
acceptableAbiotic	Category	Only specified for biotic compartment types (should remain as "<Unset>" for abiotic compartment types). Identifies the category of primary abiotic compartment type (e.g., Abiotic | Soil | Surface Soil) associated with a volume element to which this biotic compartment type can be added.
category	Category	The category of the compartment type (e.g., Fish | Water Column Omnivore, Abiotic | Air | Air - Default, Abiotic | Soil | Root Zone | Root Zone - Default).
concentrationOutputFactor	Real Number	A factor that is multiplied by the concentration in this compartment in the base units in order to create outputs in the units desired by the user. The use of this property, including the base units for each compartment type, is explained in detail in Module 14, Simulation Results and Analyses.
concentrationOutputUnits	Text	The output units created based on the *concentrationOutputFactor* property. The use of this property is explained in detail in Module 14, Simulation Results and Analyses.
isBiotic	True/False	Indicates whether the compartment type is biotic.

Prior to adding new compartment types to the library, the user will need to review the available algorithms to determine if additional or different algorithms are required to model transport and transformation for this compartment type. With this step, the user has responsibility for the appropriateness and scientific defensibility of the algorithms which will be applied to transport and/or transform chemicals in the new compartment type in their scenario. Any new algorithms required for the new compartment type can be added using the steps described in Section 4.

There are three different ways to add a compartment type to a TRIM.FaTE library. The first method involves creating a new compartment type "from scratch" (i.e., without using information from an existing compartment type). This is the most common method for creating

compartment types that are not similar to any existing compartment types. To create a new compartment type using this method, the user should follow these steps.

(1) Open the TRIM.FaTE library.
(2) Select "Compartments" from the "Contents" pull-down menu in the object browser.
(3) Click the "New" button at the bottom of the list of currently included compartments.
(4) Specify a name for the new compartment in the pop-up window and click "OK."
(5) Using the Property Editor, specify values for the following compartment properties: *acceptableAbiotic* (leave "<Unset>" for abiotic compartments), *category*, *concentrationOutputFactor*, *concentrationOutputUnits*, and *isBiotic* (default is "false").
(6) Using the Property Editor, add to the newly added Compartment those properties required (directly or indirectly) by the algorithms applied for the new compartment type, and edit the values of the properties to reflect the characteristics of the new compartment.

The second method involves duplicating an existing compartment type and editing the duplicate as needed. This method can be used when the new compartment type is very similar to an existing compartment type. To create a new compartment type using this method, the user should follow these steps.

(1) Open the TRIM.FaTE library.
(2) Select "Compartments" from the "Contents" pull-down menu in the object browser.
(3) Select similar compartment from list of currently included compartments and select "Duplicate" button at bottom of list.
(4) Specify a name for the new compartment in the pop-up window and click "OK."
(5) Using the Property Editor, edit values as needed for the following compartment properties: *acceptableAbiotic* (leave "<Unset>" for abiotic compartments), *category*, *concentrationOutputFactor*, *concentrationOutputUnits*, and *isBiotic*.
(6) Using the Property Editor, remove from the new Compartment any properties not needed (directly or indirectly) by algorithms applied for this compartment type. This is done by selecting the

---

TOP-LEVEL CATEGORY NAMES FOR COMPARTMENTS

For the property *category* that is required for all compartment types, multiple "levels" can be defined. The first level designates the "top level" of the category (i.e., basic type), with subsequent level(s) indicating a more specific compartment. Different levels are separated by a pipe (i.e., |). Abiotic compartment types (e.g., air, soil) all have a top-level category of "abiotic." Biotic compartments must be assigned to one of the following top-level category names:

- Bird
- Fish
- In sect
- Mammal
- Aquatic Plant
- Terrestrial Plant
- Earthworm
- Amphibian
- Reptile

For example, the full value for category property for the common loon is "Bird | Common Loon."

property from the property table for the new Compartment and clicking on the "Del" button. It is noted that the presence of unused properties will not affect execution of a simulation, but their presence may provide later confusion regarding the compartment characteristics on which the transport and transformation algorithms rely.

(7)   Using the Property Editor, edit existing properties as needed to reflect the characteristics of the new compartment type. If the duplicated compartment type is very similar to the new compartment type, this step may only require editing the values of some of the properties.

(8)   Using the Property Editor, add to the new Compartment any additional properties required (directly or indirectly) by the algorithms applied for the new compartment type, and edit the values of the properties to reflect the characteristics of the new compartment type.

The third method for including new compartment types in a library involves using a Library Object Import file. To use this method, the user should follow these steps.

(1)   Create a text file in the appropriate Library Object Import file format (described in Module 3) containing the new Compartment and the compartment properties referenced (directly or indirectly) by the algorithms applied for the new compartment type.

(2)   Open the TRIM.FaTE library.

(3)   Select "Import" from the "File" menu in the Library Window.

(4)   Choose "Object Importer" in Choose Importer pop-up window and click "OK."

(5)   Select the Library Object Import file using the File Browser and click "Import TRIM Objects."

6.   ADDING COMPOSITE COMPARTMENTS

Composite compartments are used to group related compartment types so that one of the included compartment types can easily reference properties of the other included compartment types in a formula. Note that composite compartments are only a concept that can simplify the setup of libraries and scenarios. It is the individual compartments in which chemical mass is stored and transformed, and among which chemical mass is transferred.

The composite compartment type existing in the current TRIM.FaTE library groups together the compartment types that comprise the different parts (e.g., leaf, root, stem) of several general types of plants (e.g., deciduous, coniferous). To model a different or more specific type of vegetation not included in the existing library (e.g., oak forest), a new composite compartment type could be added. Prior to creating the new composite compartment type, however, the individual compartment types that will be grouped must be created (as described in Section 5).

A composite compartment is created for a set of existing compartments via the following steps.

(1)   Open the TRIM.FaTE library.

(2)   Select "Composite Compartments" from the "Contents" pull-down menu in the object browser.

(3)   Click the "New" button at the bottom of the list of currently included composite compartments.

(4)   Specify a name for the new composite compartment in the pop-up window and click "OK."

(5)   Confirm that the compartments to be included in the composite compartment exist in the library.

(6)   Select the compartments to be included in the composite compartment from the "Available Compartments" column and click the "<<Add" button to associate them with the composite compartment.

Alternatively, new composite compartment types can be added to library objects using a Library Object Import file. To use this method, the user should follow these steps.

(1)   Create a text file in the appropriate Library Object Import file format (described in Module 3) containing the new composite compartment type and specifying the compartment types to be associated with it.

(2)   Open the TRIM.FaTE library.

(3)   Select "Import" from the "File" menu in the Library Window.

(4)   Choose "Object Importer" in Choose Importer pop-up window and click "OK."

(5)   Select the Library Object Import file using the File Browser and click "Import TRIM Objects."

7.   REFERENCES

U.S. Environmental Protection Agency. 2002. TRIM.FaTE Technical Support Document. Volume I: Description of Module. EPA-453/R-02-011a. Office of Air Quality Planning and Standards.

APPENDIX A
FORMULA SYNTAX

The value of a numeric property (i.e., properties with data types of "real number") can be represented by a formula. Formulas can be specified by selecting a property in the Property Editor, clicking the "Form" button, and selecting "Formula Real Number" in the pop-up window. After the "Form" of the property has been specified as a "Formula Real Number," the Value Editor appears below the Property Editor. The user can specify a formula as the value of the property in the Value Editor by typing the formula in the correct format and clicking the "Store" button. The format for writing formulas is described in this appendix, which is divided into the following five sections.

A.1	*General Guidelines*	This section provides the basic rules and syntax used in writing formulas in the TRIM.FaTE syntax.
A.2	*Built-in Constants*	This section provides a list and description of each of the constants available for writing formulas.
A.3	*Built-in Functions*	This section provides a list and description of each of the functions available for writing formulas.
A.4	*Referencing Object Properties*	This section provides the guidelines and syntax for referencing object properties in a formula.
A.5	*Formula Examples*	This section provides examples of formulas using the various options presented in this appendix.

A.1   GENERAL GUIDELINES

- All values are interpreted as floating point values.

- For logical operations (i.e., true/false), 0.0 is considered to be "false" and any nonzero value is considered to be "true."

- Upper- and lower-case letters can be used interchangeably.

- Non-integer values can be specified in decimal notation (e.g., 2303, 0.0056) or scientific notation (e.g., 2.303E03, 5.6E-03).

- All standard arithmetic operations are supported, including:

Operation	Symbol
Addition	+
Subtraction	-
Multiplication	*
Division	/
Exponential	**

- Comparison operations include:

Operation	Symbol
Less than	<
Less than or equal to	<=
Greater than	>
Greater than or equal to	>=
Equal	==
Not equal	!=

- Operators used in evaluating conditional statements include "&&" (and) and "||" (or).

- A conditional expression is written as:

        \<condition\> ? \<if-true expression\> : \<if-false expression\>

    For example, "hour < 6 ? 0.0 : 5.0" will have a value of 0.0 when the hour is less than 6 and 5.0 otherwise. Conditional expressions can be nested.

- The formulas are evaluated by following the standard order of operations. Parentheses or brackets can be used to group related calculations or to change the order of evaluation.

A.2   BUILT-IN CONSTANTS

There are several constants included in TRIM.FaTE to simplify and reduce possible errors in formulas. These physical, chemical, and unit conversion constants can be specified by name in formulas. Table A-1 provides a list of all of the built-in constants in TRIM.FaTE.

Table A-1
TRIM.FaTE Built-in Constants

NAME	VALUE (units, if applicable)	DESCRIPTION
constants.g_per_kg	1000	Conversion factor between kg and g
constants.idealGasConstant	8.314 (Pa-m3/mol-K)	Ideal Gas Constant
constants.kg_per_g	0.001	Conversion factor between g and kg
constants.kg_per_m3_Water	1000 (kg/m3)	Density of water
constants.kg_per_ug	0	Conversion factor between μg and kg
constants.L_per_m3	1000	Conversion factor between m^3 and L
constants.m3_per_kg_Water	1e-3 (m3/kg)	Inverse of density of water
constants.m3_per_L	0.001	Conversion factor between L and m^3
constants.m3_per_um3	0	Conversion factor between μm^3 and m^3
constants.pi	3.14159265359	π
constants.ug_per_kg	1e+09	Conversion factor between kg and μg
constants.um3_per_m3	1e+18	Conversion factor between m^3 and μm^3
constants.vonKarmensConstant	0.74	von Karmens Constant
constants.waterMeltingPoint	273.15 (K)	Melting point of water

A.3   BUILT-IN FUNCTIONS

There are a number of built-in functions in TRIM.FaTE used to simplify formulas and standardize the approach for particular calculations. These functions are divided into three different categories: mathematical functions, spatial functions, and atmospheric transport functions. Table A-2 provides a list and description of each of the available functions by category.

Table A-2
Built-in Functions

FUNCTION	DESCRIPTION
Mathematical Functions	
abs(a1)	Returns the absolute value of a1
exp(a1)	Returns the exponential of a1
ln(a1)	Returns the natural logarithm of a1
log10(a1)	Returns the log base 10 of a1
max(a1, a2, a3, ...)	Returns the maximum value among a1, a2, a3,...
min(a1, a2, a3, ...)	Returns the minimum value among a1, a2, a3,...
mod(a1, a2)	Returns the modulo of a1 relative to a2
sqrt(a1)	Returns the square root of a1
Spatial Functions	
area()	Returns the horizontal area (in m^2) of a volume element. This function can only be used in formulas referencing volume elements. By default, each volume element has an "area" property that invokes this function.
interfacialArea()	Returns the interfacial area (in m^2) between two volume elements. This function can only be used in properties for links between compartments in adjacent volume elements. This function cannot be applied to links between compartments within a volume element. By default, each link has an "interfacialArea" property that invokes this function.
volume()	Returns the volume (in m^3) of a volume element. This function can only be used in formulas referencing volume elements. Other types of objects, such as compartments, can have "volume" properties, but those properties cannot invoke this function. By default, volume elements have a "volume" property that invokes this function.
Atmospheric Transport Functions	
advectiveTransport (horizontalWindSpeed, windDirection, verticalWindSpeed, stabilityClass)	Returns a factor for advective transport (m/d) which, when multiplied by the interfacial area of the link and divided by the volume of the sending compartment, gives a standard transfer factor (1/d). This function requires four arguments: horizontal wind speed (m/s), horizontal wind direction (degrees clockwise from north from which the wind is blowing), vertical wind speed (m/s), and stability class. The stability class argument is a number 1-6 corresponding to the stability classes A-F. This function can only be used in properties that belong to algorithms.
boundaryContribution (boundaryConcentration, horizontalWindSpeed, windDirection)	Returns the advective contribution (in moles) to an air compartment from an external boundary of the volume element containing the compartment. Concentration must be specified in g/m3, speed in m/s, and direction in degrees clockwise from north from which the wind is blowing. This function can only be used in properties attached to air compartments contained within volume elements bordering the boundary of the modeling region.

A.4   REFERENCING OBJECT PROPERTIES

Properties of objects in the simulation are referenced by the general syntax:

<object specification>.<property name>

The methods of specifying objects are described below.

A.4.1 General Guidelines

- Property names can consist of letters, digits, "$", and "_" with the restriction that the first character cannot be a digit.

- To reference the object that contains the property, use the appropriate keyword from the following list.

compartment
compositeCompartment
volumeElement
link
algorithm
source
chemical
project
scenario

For example, the user could reference the "DustDensity" property of an air compartment in the formula used to calculate the value of another property in the compartment using "compartment.DustDensity."

A.4.2 Referencing Chemical-specific Properties

Some properties, such as oxidation rate in a compartment, are chemical-specific. Appending ".chemical" after an object specification references a chemical-specific property. How this is used depends on the type of object that is specified. For example, if an algorithm acts on the chemical "Hg", "sendingCompartment.chemical.oxidationRate" will reference the "oxidationRate" property for Hg in the compartment on the sending side of a link.

Users should not confuse properties of chemical objects with chemical-specific properties. Chemical objects have their own set of properties that are referenced within the chemical object as "chemical.<property name>". The properties of the chemical objects are referenced within algorithms using either "sendingChemical.<property name>" or

"receivingChemical.<property name>". These properties are referenced in chemical-specific properties within all other object types as "currentChemical.<property name>". Note that properties of chemical objects cannot be referenced in non-chemical specific properties for object types other than "chemical" and "algorithm". For example, the property "r" in a compartment called "Soil" that was specific to Hg could refer to the "Soil" property "q" that was specific to Hg as "compartment.chemical.q". In addition, the property "s" in "Soil" that is specific to Hg could refer to the "t" property of Hg as "currentChemical.t".

A.4.2   Referencing Properties in Other Objects

Table A-3 shows the different ways properties associated with objects other than the one containing the formula can be specified. The table indicates the types of objects within which these properties can be referenced and whether the reference to the property requires a compartment category.

The object specifications for the properties with "Yes" for "Requires Compartment Category" must be followed by a compartment category in square brackets ("[ ]"). For example,

receivingVolumeElementSumOf[Plant | Leaf].DryDepInterceptionFraction

The compartment category is treated as additional information to identify which compartment is being referenced. A specified category will include all more specific categories. For example, category "air" could include category "air | boundary layer" and "air | surface layer". Thus, the user must ensure that only one compartment satisfies the specified category. The column "Requires Compartment Category" below indicates which object specifications require a compartment category.

Table A-3
Properties Referenced from Other Objects

Keyword	Requires Compartment Category	Formula Can Belong To	Notes
containingProject	N	algorithm, compartment, composite compartment, link, scenario, source, volume element	The project that contains the formula's owner.
containingScenario	N	algorithm, chemical, compartment, composite compartment, link, source, volume element	The scenario that contains the formula's owner.

Keyword	Requires Compartment Category	Formula Can Belong To	Notes
containingVolumeElement	N	compartment, composite compartment	The volume element that contains the formula's owner.
containingVolumeElementSumOf	Y	compartment, composite compartment	This finds all compartments in the same volume element as the formula's owner that have the specified category and computes the sum of the specified property for each compartment found. If no compartments are found, the result is 0.0.
currentChemical	N	All	The chemical that matches the property's qualifier.
linkedCompartment	Y	compartment	The compartment linked to the formula's compartment that belongs to the specified category.
receivingChemical	N	algorithm	The chemical produced or transported by the algorithm.
receivingCompartment	N	algorithm, link	The compartment on the receiving side of the link.
receivingLinkedCompartment	Y	algorithm, link	The compartment with the given category that is connected to the link's receiving compartment.
receivingParcel	Y	algorithm, link	The compartment with the given category that is within a volume element that has the same parcel as the link's receiving volume element.
receivingPrimaryAbioticCompartment	N	algorithm, link	The primary abiotic compartment of this link's receiving volume element.
receivingVolumeElement	N	algorithm, link	This link's receiving volume element.
receivingVolumeElementSumOf	Y	algorithm, link	This finds all compartments in the receiving volume element that have the specified category and computes the sum of the specified property for each compartment found. If no compartments are found, the result is 0.0.

Keyword	Requires Compartment Category	Formula Can Belong To	Notes
receivingWithinCompositeCompartment	Y	algorithm, link	The compartment with the specified category that is within the receiving compartment's parent composite compartment.
sendingChemical	N	algorithm	The chemical consumed or transported by the algorithm.
sendingCompartment	N	algorithm, link	The compartment on the sending side of the link.
sendingLinkedCompartment	Y	algorithm, link	The compartment with the given category that is connected to this link's sending compartment.
sendingParcel	Y	algorithm, link	The compartment with the given category that is within a volume element that has the same parcel as the link's receiving volume element.
sendingPrimaryAbioticCompartment	N	algorithm, link	The primary abiotic compartment of this link's receiving volume element.
sendingVolumeElement	N	algorithm, link	This link's sending volume element.
sendingVolumeElementSumOf	Y	algorithm, link	This finds all compartments in the sending volume element that have the specified category and computes the sum of the specified property for each compartment found. If no compartments are found, the result is 0.0.
sendingWithinCompositeCompartment	Y	algorithm, link	The compartment with the specified category that is within the sending compartment's parent composite compartment.
theLink	N	algorithm	The link to which the algorithm is attached.
thePrimaryAbioticCompartment	N	compartment, composite compartment	The primary abiotic compartment of the volume element that contains this compartment or composite compartment.

Keyword	Requires Compartment Category	Formula Can Belong To	Notes
withinCompositeCompartment	Y	compartment	The compartment with the specified category that is within this compartment's parent composite compartment.
withinContainingVolumeElement	Y	compartment	The compartment with the specified category that is within this compartment's volume element.

A.5   FORMULA EXAMPLES

This section provides a number of different examples of formulas using the syntax described above. An explanation of each example is provided to aid users in understanding how to apply the syntax.

Example 1.

Object Type: Compartment
Object Name: Surface Soil
Property Name: Height
Chemical (if chemical-specific): N/A

Formula: containingVolumeElement.Height

Description: The value of this property is defined as the value of the "Height" property of the volume element containing this Surface Soil compartment.

Example 2.

Object Type: Compartment
Object Name: Air
Property Name: FractionMass_Sorbed
Chemical (if chemical-specific): Benzo(A)Pyrene

Formula: 1 - 1/(1+compartment.Chemical.ParticleGasPartitionCoefficient *
                 compartment.DustLoad* Constants.ug_per_kg)

Description: The value of this property is a function of the "ParticleGasPartitionCoefficient" property for Benzo(A)Pyrene for this Air Compartment, the "DustLoad" property for this Air compartment, and the built-in constant "ug_per_kg."

Example 3.

   Object Type: Compartment
   Object Name: Leaf
   Property Name: isDay
   Chemical (if chemical-specific): N/A

   Formula: containingScenario.simulateSteadyState == 1 ?
                   compartment.isDay_SteadyState : containingScenario.isDay_Dynamic

   Description: The value of this property is equal to the value of the "isDay_SteadyState"
   property for this Leaf compartment if the scenario property "simulateSteadyState" is true,
   and the "isDay_Dynamic" property of the scenario if the scenario property
   "simulateSteadyState" is false.

Example 4.

   Object Type: Algorithm
   Object Name: Advection from Air to Air
   Property Name: transferFactor
   Chemical (if chemical-specific): N/A

   Formula: (advectiveTransport(containingScenario.horizontalWindSpeed,
                   containingScenario.windDirection,
                   containingScenario.verticalWindSpeed,
                   containingScenario.stabilityClass)) * TheLink.InterfacialArea /
                   SendingCompartment.Volume

   Description: The value of this property is equal to the value of the function
   "advectiveTransport" (evaluated using the properties "horizontalWindSpeed,"
   "windDirection," "verticalWindSpeed," and "stabilityClass" for the scenario) times the
   value of the property "InterfacialArea" for the link associated with the algorithm divided
   by the "Volume" property of the sending air compartment.