SimBiology® Release Notes

MATLAB®

MathWorks®

How to Contact MathWorks

Latest news: www.mathworks.com

Sales and services: www.mathworks.com/sales_and_services

User community: www.mathworks.com/matlabcentral

Technical support: www.mathworks.com/support/contact_us

Phone: 508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Contents

R2020a

R2019b

R2019a

Bug Fixes

R2018b

R2018a

R2017b

R2015b

R2015a

R2014b

R2014a

R2013b

R2013a

R2012b

R2012a

R2011b

R2011a

R2010b

R2010a

R2009b

R2009a

R2008b

R2008a

R2007b+

R2007b

No New Features or Changes

R2007a

No New Features or Changes

R2006b+

R2006b

Bug Fixes

R2006a+

R2006a

R14SP3+

R2021a

Version: 6.1

New Features

Bug Fixes

Compatibility Considerations

Undo and redo model configurations

While a model is open in one (or both) of the SimBiology apps, you can undo and redo model building and diagram configuration actions taken in the apps or at the command line. Click the **Undo** or **Redo** button on the **Home** tab of the **SimBiology Model Builder** app. For details, see "Undo and Redo Model Changes in SimBiology".

Configure SimBiology diagram block and line properties programmatically

You can adjust the properties of blocks and lines shown in the diagram of the **SimBiology Model Builder** app programmatically. For instance, you can change various block and line properties such as block position, block shape, block text, line color, and line width at the command line. You can also join and split species blocks. For details, see the following functions:

- `simbio.diagram.getBlock`
- `simbio.diagram.setBlock`
- `simbio.diagram.getLine`
- `simbio.diagram.setLine`
- `simbio.diagram.splitBlock`
- `simbio.diagram.joinBlock`

Integration-based profile likelihood confidence intervals

In addition to the existing optimization-based profile likelihood confidence interval method, `sbioparameterci` lets you compute the profile likelihood confidence intervals using a new, integration-based approach. This new approach is based on integrating differential equations derived from the Lagrange equations. To use the integration-based method, set "`'Type'`" to `'profileLikelihood'` and "`'UseIntegration'`" to `true`.

SimBiology Model Analyzer: Improved plotting performance

The **Model Analyzer** app now generates plots faster than before. The improved performance has been verified in various cases including:

- plotting time courses of numerous states from a single simulation
- plotting a single response from a scan over numerous parameter values
- plotting a single response from a scan over multiple parameters with different categorization options
- plotting numerous groups of experimental and simulation data

Functionality being removed or changed

SimBiology Model Builder: Variants and doses are included in exported HTML file
Behavior change

When you export a SimBiology model as an HTML file using the **SimBiology Model Builder** app, the app lets you optionally include the variants and doses information in the exported HTML file.

SimBiology Model Builder: Selectively edit the properties of an object after multiselecting several objects in the diagram
Behavior change

If you have selected multiple objects of different types in the diagram, such as species, reactions, and lines, the **Property Editor** lets you pick the object that you want to edit the properties for.
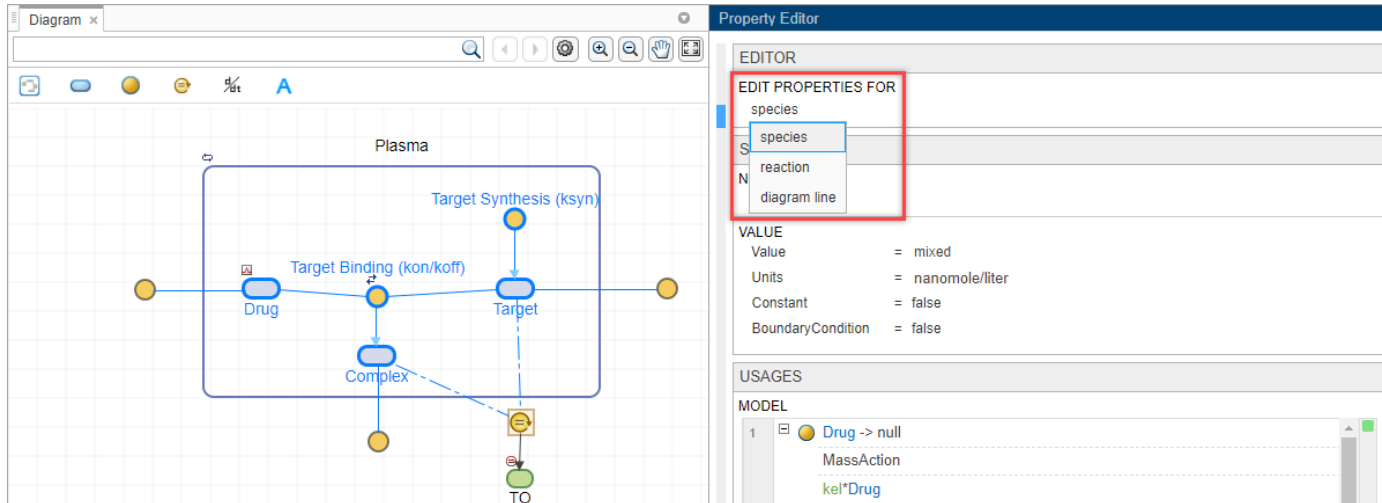
SimBiology Model Builder: Diagram alignment tools are available
Behavior change

You can perform block alignments by using the diagram alignment tools. To access the tools, select **Diagram Tools** > `Diagram Alignment Tools` on the **Home** tab of the app.
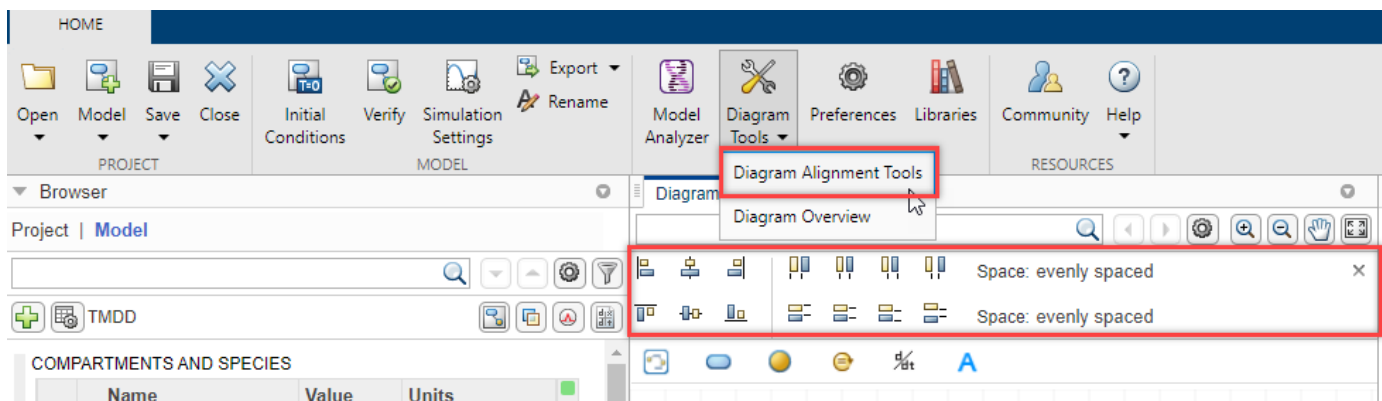
SimBiology Model Analyzer: Automatic creation of datasheet after program run
Behavior change

After you run the **Fit Data** program or **Non-Compartmental Analysis** program, the app automatically creates and opens a datasheet with the results.

SimBiology Model Analyzer: Enable and disable program warnings
Behavior change

The app shows all enabled warnings in the **Messages** section of a program. To disable a warning, first right-click the warning and select **Copy Command to Turn Off Warning**. Then go back to the MATLAB command line and paste the command.

For example, the copied command could look like:

```
warning('off', 'SimBiology:DimAnalysisNotDone_MatlabFcn_Dimensionless')
```

To enable the same warning, enter the following.

```
warning('on', 'SimBiology:DimAnalysisNotDone_MatlabFcn_Dimensionless')
```

SimBiology Model Analyzer: Project View is the new default
Behavior change

The app now uses the **Project View** by default. This view groups the datasheets and plots with the program that generated them. You can change the view from the action menu of the **Browser** panel as shown in the following screenshot.
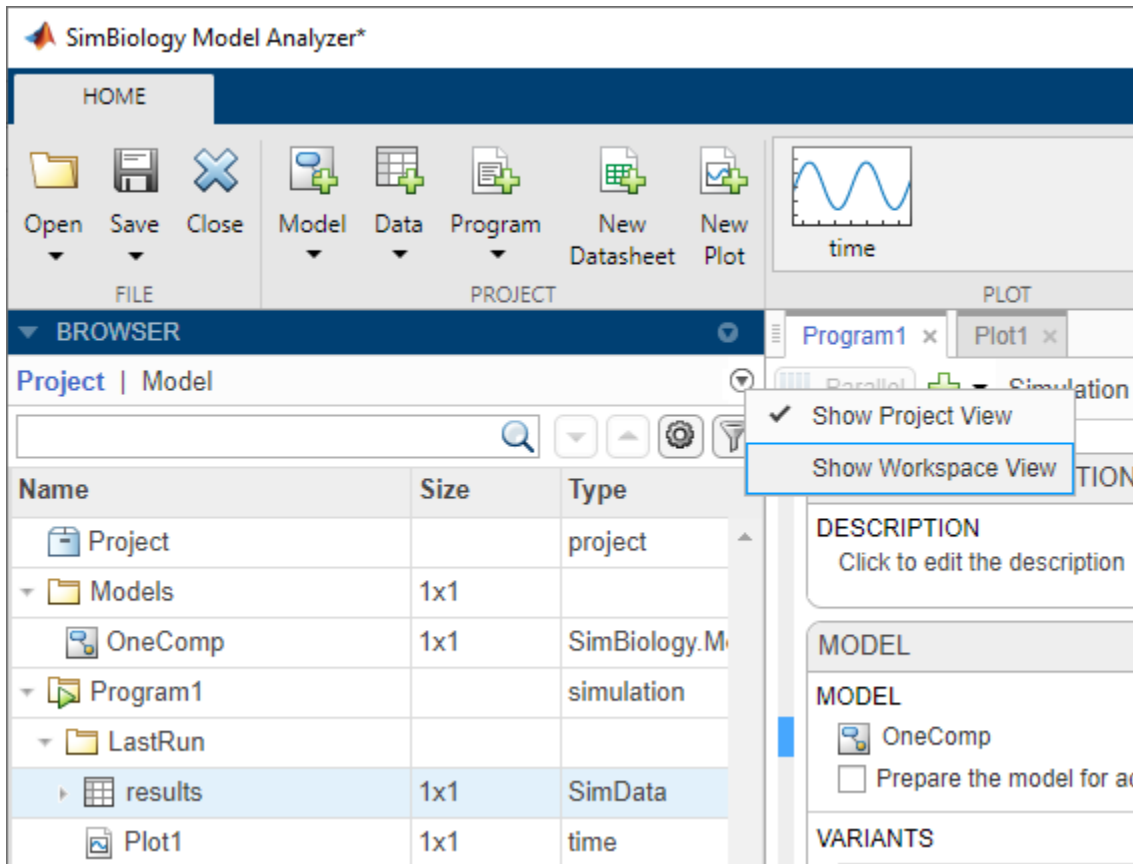
SimBiology Model Analyzer: NCA step is available for Simulation and Scan Programs
Behavior change

You can now perform noncompartmental analysis (NCA) on data from simulation and scan programs.

Annotation property has been removed
Errors

The `Annotation` property of the following objects has been removed. Use the Notes property instead.

- `AbstractKineticLaw`
- `Compartment`
- `ConfigSet`
- `Event`
- `KineticLaw`
- `Model`
- `Observable`
- `Parameter`
- `Reaction`
- `RepeatDose`
- `Rule`
- `ScheduleDose`
- `Species`
- `Unit`
- `UnitPrefix`
- `Variant`

Offset property has been removed
Errors

The `Offset` property of a `Unit` object has been removed. Use an absolute unit that does not require an offset. For example, use kelvin instead of Celsius.

sbioshowunits no longer returns offset information
Behavior change

When you specify four output arguments when calling `sbioshowunits`, the function no longer returns the offset information as the fourth output. The fourth output is now a logical vector that is true for each built-in unit.

You can no longer specify five output arguments when you call `sbioshowunits`.

sbiounit no longer accepts offset information as an input
Behavior change

You can no longer specify an offset as an input argument when you call `sbiounit`. Use an absolute unit that does not require an offset, such as kelvin instead of Celsius.

Celsius and Fahrenheit units have been removed
Errors

`Celsius`, `celsius`, and `fahrenheit` units have been removed from the built-in units library. Use `kelvin` units instead.

If you have a model containing `Celsius`, `celsius`, or `fahrenheit` units, change the units to `kelvin`.

If you have a script that sets the Units property of a parameter to `'Celsius'`, `'celsius'`, or `'fahrenheit'`, set this property to `'kelvin'` and adjust the `Value` property accordingly. Similarly, if you have compound units, make appropriate changes. For example, change `joule/Celsius` to `joule/kevin`.

sbmlimport: support for URL schemes other than HTTP and HTTPS will be removed
Still runs

Support for passing in a URL that is not HTTP or HTTPS as an input to `sbmlimport` will be removed in a future release. Download the file locally first and then use the file name as an input instead.

R2020b

Version: 6.0

New Features

Bug Fixes

Compatibility Considerations

SimBiology Model Builder: Build QSP and mechanistic systems biology models

The **SimBiology Model Builder** app lets you interactively build models of dynamic systems, such as quantitative systems pharmacology (QSP) models, physiologically based pharmacokinetic (PBPK) models, and pharmacokinetic/pharmacodynamic (PK/PD) models. It provides a block diagram editor with integrated model component tables and property editor to build the model reaction schematic.

To open the app, type `simBiologyModelBuilder` at the command line or click the **SimBiology Model Builder** button on the **Apps** tab under **Computational Biology**.

To analyze models, use the **SimBiology Model Analyzer** app.

Move SimBiology object to new parent or owner

You can now move a species or parameter object to a new parent and a compartment object to a new owner using the `move` function. For details about the function for a species or parameter object, see here. For compartments, see here.

Functionality being removed or changed

Loading a project or model from the command line when the SimBiology Model Builder or SimBiology Model Analyzer app is open
Behavior change

If the **SimBiology Model Builder** or **SimBiology Model Analyzer** app is open, you cannot load a project or model from the command line using the `simBiologyModelBuilder`, `simBiologyModelAnalyzer`, or `simbiology` functions. Load the project or model from the app directly.

Changes to 'cubic' method of resample
Behavior change

The `resample` function of a `SimData` object provides different interpolation methods. `'cubic'`, one of these interpolation methods, now performs cubic convolution. Previously, `'cubic'` was the same as `'pchip'`. For details, see `interp1` (MATLAB).

Reaction names cannot be set to empty
Behavior change

You cannot set a reaction name to an empty character vector (`''`) or empty string (`""`). For details on naming model components, see Guidelines for Naming Model Components.

New backup file name and backup behavior in SimBiology apps
Behavior change

When you open a SimBiology project that was created in R2020a or earlier in the **SimBiology Model Builder** or **SimBiology Model Analyzer** app, the app creates a backup file. For example, the app creates a backup file *filename*`_R2019a.sbproj.backup` for a project *filename* created using R2019a. If such a backup file already exists, the app overwrites it and does not save the backup to a different file name as in previous versions.

sbiodesktop will be removed
Still runs

sbiodesktop will be removed in a future release. Use simBiologyModelBuilder instead.

R2020a

Version: 5.10

New Features

Bug Fixes

Compatibility Considerations

sbiosobol and sbiompgsa: Perform global sensitivity analysis (GSA)

You can now perform global sensitivity analysis (GSA) on a SimBiology model to explore the effects of variations in model parameters, species, or compartments on the model response. GSA can help you detect which parameters are influential on the model output. Use the following functions to perform GSA.

- `sbiosobol` — Compute the first and total order Sobol indices. The Sobol method is based on the decomposition of the variance of the model output with respect to contributions from varying model quantities. For more details about the function and results object, see `sbiosobol` and `SimBiology.gsa.Sobol`.

- `sbiompgsa` — Perform multiparametric GSA for decision-making based on the sensitivity of a model response with respect to variations in model quantities. For more details about the function and results object, `sbiompgsa` and `SimBiology.gsa.MPGSA`.

Observables: Perform post-simulation calculations

You can perform post-simulation calculations by defining and evaluating custom expressions—for instance, to calculate the fraction of ligand that is bound to a receptor at each time step. These expressions are called *observables*. You can add observable expressions to a SimBiology model or `SimData` object.

Add to model

When you add an observable expression to a model object using `addobservable(model)`, the function returns a new object called `Observable` and adds the object to the model. The expression is evaluated after every model simulation and the results are stored in the returned `SimData` object. You can access the model observable property by typing *m1*.`Observables` at the command line, where *m1* is a SimBiology model.

The new `observable` object has the following properties.

- `Name` — Object name.
- `Expression` — Observable expression.
- `Active` — Flag indicating whether to evaluate the observable expression during model simulation. By default, this property is set to `true`.
- `Units` — Observable unit.
- `Notes` — Description of the object.
- `Parent` — Parent object for the observable object.
- `Tag` — Label to associate with the object.
- `Type` — Object type.
- `UserData` — Data associated with the object.

Add to SimData object

When you add an observable expression to a `SimData` object using `addobservable(SimData)`, the function returns a new `SimData` object with the calculated values for the added observable expressions. You can also use the following functions when you work with observables: `updateobservable` and `renameobservable`.

The `SimData` object has the following new and updated properties to support observables. These updates also apply to the `SimData` object returned by `sbiosimulate` or `SimFunction` after you simulate a model that has observable expressions.

- `Data` — The property contains the calculated values for observable expressions as columns in the data matrix. Scalar observables are repeated so that the same value appears in every row.
- `DataCount` — The property includes observables in the count. This is equal to the number of columns in the `Data` property or the number of elements in the `DataInfo` property.
- `DataInfo` — The property includes entries for observables. Each entry is a cell array containing a structure with these fields:
  - `Scalar` — Flag to indicate if the observable expression is scalar-valued.
  - `Expression` — Observable expression
  - `Type` — `'observable'`
  - `Name` — Name of the observable expression
  - `Units` — Observable expression unit
- `DataNames` — The property includes the names of added observable expressions in the same order as the columns of the `Data` property.
- `ScalarObservables` — This new property contains a table of all scalar-valued observables. The table variable names show the names of scalar observables in the same order as they appear in the `Data` property (and the `DataNames` property).
- `VectorObservables` — This new property contains a table of all vector-valued observables. The table variable names show the names of vector observables in the same order as they appear in the `Data` property (and the `DataNames` property).

The following `SimData` object functions have also been updated to support the observable expression results.

- `getdata` — The second output (`x`) contains the calculated results from observables. It is identical to `SimData.Data`. The third output (`names`) is identical to `SimData.DataNames`. If you specify the `'nummetadata'` format, the output `names` is identical to `SimData.DataInfo`.
- `resample` — The function interpolates the observable results in the same way as it interpolates other results.
- `select` — The function supports selection by using the type `observable`.
- `selectbyname` — The function supports selecting observables by name.

Specify Observables as responses

You can specify observables as responses (model outputs) when you simulate the model using a `SimFunction` object, fit the model using `sbiofit` or `sbiofitmixed`, or perform global sensitivity analysis using `sbiosobol` or `sbiompgsa`.

Compatibility Considerations

The **SimBiology Model Analyzer** app now lets you add and evaluate observable expressions. For details, see "Changes to support observable expressions in SimBiology Model Analyzer app" on page 3-4.

Functionality being removed or changed

Changes to support observable expressions in SimBiology Model Analyzer app
Behavior change

To allow adding, editing, and evaluating observable expressions, **SimBiology Model Analyzer** has been updated as follows.

Changes to Postprocessing: Calculate Statistics Step

- The step has been renamed **Postprocessing: Calculate Observables**.
- The **Observables** (previously known as **Statistics**) table has a new **Units** column. You can pick an existing observable from the model or add a new observable expression to the table. If you add a new observable expression to the table, the expression is also added to the model as a new observable object automatically.
- The name of an observable must be unique across the model, that is, the name must be different from other model component names. An observable name cannot be an empty text, `null`, `time`, or contain the characters `->`, `<->`, `[`, or `]`.
- When **Unit Conversion** is on, any observable expression added to the step (or datasheet) must have units defined. Otherwise, the program throws an error and the results are returned as `NaN` values for the observables.
- The **Use** check box controls which observable expressions are evaluated when you run the program. The check box does not change the `Active` property of the observable object on the model.
- Deleting a row from the table (by either using the delete key or the context menu option) does not delete the corresponding observable object from the model. To delete an observable object from the model, right-click on the row and select **Delete Observable from Model**.
- The **Run this program step** button and the default plot generation button have been removed from the **Postprocessing: Calculate Observables** step. To evaluate the observable expressions, run the whole program or run the simulation step of the program. Use the plot generation button of the simulation step to plot the observable results by default or not.

Changes to Calculate Statistics Program

- The program has been renamed **Calculate Observables**. This program lets you evaluate a series of observable expressions on an existing simulation data and plot the results.
- The program returns results in a `SimData` (simulation data) format instead of a table format.

Changes to Datasheet

- If you add an observable expression to a datasheet (containing simulation data **SimData**) from the **LastRun** folder, the expression is added to the model as an observable object automatically. Similarly, if you make changes to expressions of existing observables in a datasheet in **LastRun**, the changes are saved on the corresponding observable objects in the model automatically.
- If you add an observable expression to a datasheet in a *saved* folder instead of the **LastRun** folder, the expression is added to the corresponding simulation data `SimData` object, but not to the model.
- When a scalar observable object (shown as `1x1` in the **Size** column in the **Browser**) is dragged onto a datasheet with other vector-valued observables, time, or state columns, the scalar observable column is filled with the same value across all the rows (scalar expanded).

Changes to Observables Folder in Worskspace

- In the **Workspace** of the **Browser**, the **Observables** folder is now removed.
- The vector-valued observables are listed together with other `SimData` states under **results**. However, the scalar-valued observables are saved under **results > scalars**.

Changes When Loading Previous Projects

- **R2019a** or earlier projects — Any existing expressions from the **Calculate Statistics** section of a task are added as observable objects to the model. If the expression names have conflicts, they are renamed by appending _*N*, where *N* is a positive integer.
- **R2019b** projects — Any existing expressions from the **Postprocessing: Calculate Statistics** step of a program and any expression in the `SimData` datasheets from the **LastRun** folder are added as observable objects to the model. Any results from **LastRun** or saved folders of the **Postprocessing: Calculate Statistics** program are converted from the table format to the `SimData` (simulation data) format.

Changes to Plotting

- You can drag the vector-valued observables onto a **time** plot to see their time courses. You can drag a scalar-valued observable to a time plot to see a constant (or threshold) line across the plot.
- Dragging the **scalars** folder onto a **Plot Matrix** plot creates a plot of the scalar observables versus sample values (or just the distribution of the scalar observables if there are no samples in the program).
- The **Calculate Observables** step in a program generates a plot matrix plot only if there are scalar observables.
- During the plot updates, any simulation data, such as states data and observable results, that have been deleted or are no longer generated by the program are removed from the corresponding plots, such as **time** and **xy** plots. Plots that are now empty due to the lack of supporting data are deleted automatically.
- When you disable the default plot generation of a program step and run the step, any existing plots showing data from this step are removed.

sbioplot Figure Changes
Behavior change

Legends are statically displayed in the figure generated by `sbioplot`. The **All Runs** checkbox has been removed.

Changes to the summary function of LeastSquaresResults object
Behavior change

The `summary` function now returns a structure array, instead of a figure with fit statistics. Each structure has the `Name` and `Table` fields which contain the corresponding fit statistics.

Scenarios objects can no longer be imported into SimBiology Model Analyzer app
Behavior change

It is recommended that you use the **Generate Samples** program or step to generate simulation scenarios in the **SimBiology Model Analyzer** app. You can no longer import `SimBiology.Scenarios` objects created at the command line into the app. To inspect samples of the object at the command line, use the `generate` function.

R2019b

Version: 5.9

New Features

Bug Fixes

Compatibility Considerations

SimBiology.Scenarios: Perform Monte Carlo simulations using various modeling and dosing conditions

The `SimBiology.Scenarios` object lets you create value sets to explore a model behavior for varying species, parameters, and compartments. Values can be drawn from various probability distributions, including lognormal distribution, normal distribution, and uniform distribution. You can combine these values with different dosing conditions and simulate to investigate the model behavior and what-if hypotheses. For more information, type `help SimBiology.Scenarios` at the command line or visit `SimBiology.Scenarios`.

SimBiology Model Analyzer: Analyze QSP, PBPK, and PK/PD models

The **SimBiology Model Analyzer** app lets you analyze models of dynamic systems such as quantitative systems pharmacology (QSP), metabolic network, physiologically based pharmacokinetic (PBPK) models, and pharmacokinetic/pharmacodynamic (PK/PD) models. The app provides several analysis methods for your models and various plots to visualize the results. For details and examples, see **SimBiology Model Analyzer**.

To open the app, type `simBiologyModelAnalyzer` at the command line or click **SimBiology Model Analyzer** on the **Apps** tab under **Computational Biology**.

To build or edit models, use the **SimBiology** app. Type `simbiology` at the command line or click **SimBiology** on the **Apps** tab.

The **SimBiology Model Analyzer** app replaces the **Task Editor** of the **SimBiology** app. For more information, see "SimBiology app no longer supports the Task Editor" on page 4-3.

Simulate and fit a SimBiology model inside a parfor loop

You can now simulate and fit a SimBiology model inside a `parfor` loop. You no longer need to call `parallel.pool.Constant` prior to the `parfor` loop. See the following code example.

```
%%
% Load the project.
sbioloadproject lotka
sdVec = [];
%%
% Simulate the model inside parfor.
parfor i = 1:4
    p = sbioselect(m1,'Name','c1');
    p.Value = i;
    sdtemp = sbiosimulate(m1);
    sdVec = [sdVec;sdtemp];
end
```

New Property Aliases for Species, Compartment, and Parameter objects

`Species`, `Compartment`, and `Parameter` objects now have the following new property aliases for some of the existing properties.

New Property Alias	Equivalent Existing Property
`Constant` (new property alias for `Species`, `Compartment`, and `Parameter` objects)	`ConstantAmount`, `ConstantCapacity`, `ConstantValue`
`Value` (new property alias for `Species` and `Compartment` objects)	`InitialAmount`, `Capacity`
`Units` (new property alias for `Species`, `Compartment`, and `Parameter` objects)	`InitialAmountUnits`, `CapacityUnits`, `ValueUnits`

There are no changes to the existing properties and they continue to work as before. The new property aliases provides a convenient way to modify these existing properties. For example, to set the amount of one of these objects as constant, you can now use:

```
obj.Constant = true;
```

You no longer have to use different constant properties (`ConstantAmount`, `ConstantCapacity`, and `ConstantValue`) for different objects.

Use rename with any SimBiology model component

You can now use the `rename` function with any SimBiology model component. The model components are `Species`, `Parameter`, `Compartment`, `AbstractKineticLaw`, `KineticLaw`, `RepeatDose`, `ScheduleDose`, `Variant`, `Event`, `Reaction`, and `Rule`.

Functionality being removed or changed

SimBiology app no longer supports the Task Editor
Behavior change

The **SimBiology** app no longer supports the **Task Editor**. Use the **SimBiology Model Analyzer** app instead.

If you load a project created in R2019a or earlier using the **SimBiology Model Analyzer** or **SimBiology** app, the project is updated to support the new features. The changes follow.

- **SimBiology Model Analyzer** provides built-in analyses called programs. Programs let you chain together a sequence of analysis steps and run each step separately or in a sequence. Programs replace tasks.
- If you load a project created previously, the existing tasks are converted to equivalent programs. However, **SimBiology Model Analyzer** does not support the following tasks: `Calculate conserved cycles`, `Search model(s)`, and `Generate report`.
- The `Create custom analysis` task has been renamed `Custom Code`.
- Any group simulation task is converted to a scan program.
- The `taskInfo` structure for a custom analysis task is not loaded when converted to a custom program.
- The **Fit Data** program does not support `nlinfit`. If Optimization Toolbox™ is available, the program uses `lsqnonlin`. If not, the program uses the `fminsearch` function. Note that the command line function `sbiofit` still supports `nlinfit` as one of the estimation methods.
- The `Sum`, `Fit Summary`, and `Sensitivity Matrix Subplot Max` plots are not supported. Custom plots are also not supported.

- The `Sensitivity Matrix Subplot` plot generated by the `Run scan with sensitivities` task is not supported.

- **SimBiology Model Analyzer** does not support the **Live Plots** functionality of the **Task Editor**. However, if **Live Plots** in the **Task Editor** is turned on and **Live Plots** shows a time plot of simulation results, then an equivalent time plot is added on import to the **SimBiology Model Analyzer** app. The time plot is updated and displayed after the program finishes (not while it is running). If there are multiple axes within **Live Plots**, the app generates one time plot per axes.

  If **Live Plots** shows a *time* plot from a math expression (also known as a math line or math data), the math data is added as variables in the *observables* table in the *results* folder of the corresponding program. You can drag and drop the variables to a new plot to visualize the data.

  If **Live Plots** shows a *scan* plot containing custom expressions, neither the data nor the plot is imported. You can add the **Postprocessing: Calculate Statistics** step to your program to enter the custom expressions and view them using a new **Plot Matrix** plot.

- The app does not allow schedule doses added to the **Explorer** section to have dose sliders. You can add repeat doses and adjust them using sliders.

- If you save a project at the command line (using `sbiosaveproject`) and open the project in the app, only models in the project are loaded. No data is loaded.

- When you open an existing project created in R2019a or earlier, the app creates a backup of the original file. The backup file has the release information suffixed to its name, `filename_release.sbproj`, where *release* is the MATLAB® release you used to create the project. For example, if you open an R2019a project named `foo.sbproj` in the app, the app creates a backup file called `foo_R2019a.sbproj`.

  - If `foo_R2019a.sbproj` already exists, the app appends the release information again and saves the backup as `foo_R2019a_R2019a.sbproj`.

  - If `foo_R2019a_R2019a.sbproj` already exists, the app appends _*N* to the file name, where *N* is a positive integer and increases monotonically as needed.

When you save a project, the app creates a backup file of the version prior to the saved version of the project. The backup file has the `.bak` extension (for example, `foo.sbproj.bak`). Every time you save the project, the BAK file is overwritten.

Object properties are column vectors
Behavior change

All SimBiology object properties are standardized to column vectors when possible. For example, the `ScheduleDose object` properties `Amount`, `Rate`, and `Time` are column vectors instead of row vectors.

getstoichmatrix and getadjacencymatrix returns species in a new order
Behavior change

The order of species in the output arguments (`M` and `objSpecies`) of `getstoichmatrix` matches the order of species returned by `modelObj.Species`.

Similarly, the order of species in the output arguments (`M` and `Headings`) of `getadjacencymatrix` matches the order of species returned by `modelObj.Species`.

Many functions now return the input object as an output
Behavior change

The following functions now return the input object as an output.

- `rename`
- `reorder`
- `removeconfigset`
- `verify`
- `addcontent`
- `rmcontent`
- `setTable`

Reactions are named automatically

Behavior change

If you create a new reaction at the command line and do not specify its name, the reaction gets an automatic name `Reaction_N`, where N is a positive integer. N increases monotonically as you add more reactions without specifying their names.

If you copy a reaction using `copyobj`, the name of the copied reaction is generated by appending `_N`, where N is the smallest integer not in use by that prefix. For example, `GivenName_1` gets copied to `GivenName_1_1`.

R2019a

Version: 5.8.2

Bug Fixes

6

R2018b

Version: 5.8.1

Bug Fixes

Compatibility Considerations

Functionality being removed or changed

Support for numeric vector and matrix inputs for fixed and random effects for sbiosampleparameters will be removed
Warns

Support for specifying a numeric vector for the fixed effects (thetas) or a matrix for the covariance matrix of random effects (omega) will be removed in a future release. Use a table instead. For details, see sbiosampleparameters.

R2018a

Version: 5.8

New Features

Bug Fixes

Compatibility Considerations

Parameterized Dosing: Specify dosing properties using model parameters for applications such as scaling dosing by body weight

You can specify the following RepeatDose object properties using model parameters: `Amount`, `Rate`, `Interval`, `StartTime`, and `RepeatCount`. Parameterized doses give you more flexibility to model different dosing scenarios such as scaling the dose amount by body weight. For details, see Parameterized and Adaptive Doses.

Compatibility Considerations

If a parameter is referenced by the LagParameterName and DurationParameterName property of a RepeatDose object, the parameter must be constant to be an input for sensitivity analysis.

If a parameter is referenced by any other RepeatDose object properties, namely, `Amount`, `Rate`, `Interval`, `StartTime`, and `RepeatCount`, you cannot use the parameter as an input for sensitivity analysis.

Adaptive Dosing: Change the behavior of the dose during a simulation in response to the values of model states or parameters

You can change the dosing behavior during a simulation in response to events, such as increasing the dose amount when the drug concentration drops below some threshold.

Some doses are not modeled as instantaneous changes. For example, an IV infusion involves drug being added at a fixed rate over a fixed duration. There are two options to define what happens if an event modifies a dose parameter while a dose is in progress. The first option lets you stop the active dose event if any relevant parameter values change (by setting the EventMode property of the dose object to `'stop'`). The second option lets you continue the active dose event to completion, and updated parameter values only affect subsequent dose events (by setting `EventMode` to `'continue'`). For an example, see Change Dose Behavior In Response to Changes in Model Parameters. For details, see Parameterized and Adaptive Doses.

R2017b

Version: 5.7

New Features

Bug Fixes

Compatibility Considerations

Noncompartmental Analysis: Calculate NCA parameters from grouped data for mixed dosing schedules and mixed routes of administration

You can calculate NCA parameters from the concentration-time data that have both single dosing schedules (one dose per individual) and multiple dosing schedules (multiple doses per individual). Three types of drug administration routes are supported: IV bolus, IV infusion, and EV (extravascular) infusion.

From the command line, use `sbionca` to calculate NCA parameters. From the SimBiology desktop, select **Open > NCA** on the **Define Plot** tab after you import the data.

Compatibility Considerations

In the SimBiology desktop, all relevant NCA parameters are computed and reported. You can no longer select the parameters to compute.

Confidence Intervals: Calculate parameter and prediction confidence intervals using Gaussian, profile likelihood, and bootstrap approximation

You can calculate confidence intervals for parameters and model predictions estimated by `sbiofit`. For the confidence intervals of estimated parameters, use `sbioparameterci`. For the confidence intervals of estimated model predictions (that is, simulation results based on estimated parameters), use `sbiopredictionci`. Three different approximations for the confidence intervals are supported: Gaussian, profile likelihood, and bootstrap. Use the `plot` function of the corresponding confidence interval object to visualize the computed confidence intervals. Get a table of parameter confidence intervals by using the `ci2table` function.

Steady State Calculations: Compute model steady states using an algebraic method

Use an algebraic method to compute steady states of a SimBiology model directly from its reactions, rate rules, and algebraic equations. When you run `sbiosteadystate`, use the name-value pair `'Method'` to specify the `'Algebraic'` or `'Simulation'` approach. The simulation approach simulates the model to find steady states. The algebraic approach attempts to find steady states by root finding, solving systems of linear equations, or using optimization.

Compatibility Considerations

The function now uses `'Algebraic'` as the default method first. If unsuccessful, it then uses `'Simulation'`. The steady state returned by the algebraic method is not guaranteed to be the same as the one found by the simulation method.

Diagram View: Annotate models using text blocks

You can drag and drop text blocks on to the model diagram to annotate the model.

Assignment Expressions: Enforce both initial and repeat assignment expressions at simulation start time

SimBiology reorders and evaluates all rules (including initial assignments) as a set of simultaneous constraints. Therefore, you cannot create a circular set of assignments, such as `a = b + 1` and `b = a + 1`. The order in which the rules appear in the model has no effect on the simulation results.

Compatibility Considerations

- You might observe different simulation results with respect to initial assignments for previous releases of SimBiology (R2017a or earlier). To recover the same simulation results at time = 0, as in R2017a or earlier, use the `updateInitialAssignments` function in the command line. If you are using the SimBiology desktop, right-click the model from the **Project Workspace** and select **Remove Order Dependencies**. For details see Evaluation Order of Rules.

- In previous releases (R2017a or earlier), if a repeated assignment changed a compartment volume, SimBiology used the compartment capacity to determine the initial amount and conserved it when the compartment volume changed at time = 0. In R2017b or later, SimBiology uses the `InitialAmount` property of the species as the initial condition at time = 0. For details, see Conservation of Amounts During Simulation.

- `sbmlimport` adds an initial assignment rule when importing SBML models with the following conditions.

  - If an SBML model has a species *s* initialized to *X* using `initialAmount` and has the attribute setting `hasOnlySubstanceUnits = false`:

    - SimBiology sets the initial amount of *s* to *X*.

    - If the model does not already have an initial assignment or repeated assignment rule for *s*, SimBiology adds an initial assignment rule `s = X / V`, where *V* is the compartment volume (capacity). This rule ensures that the initial amount of *s* is a concentration unit.

    - If the model already has an initial assignment or repeated assignment rule for *s*, then SimBiology does not use the value *X*. Instead, SimBiology evaluates the rule and sets the appropriate initial amount.

  - If an SBML model has a species *s* initialized to *X* using `initialConcentration` and has the attribute setting `hasOnlySubstanceUnits = true`:

    - SimBiology sets the initial amount of *s* to *X*.

    - If the model does not already have an initial assignment or repeated assignment rule for *s*, SimBiology adds an initial assignment rule `s = X * V`, where *V* is the compartment volume (capacity). This rule ensures that the initial amount of *s* is an amount unit.

    - If the model already has an initial assignment or repeated assignment rule for *s*, then SimBiology does not use the value *X*. Instead, SimBiology evaluates the rule and sets the appropriate initial amount.

- If a species *s* in an SBML model has the attribute setting `hasOnlySubstanceUnits = true` without any units defined, SimBiology issues a warning and sets the species amount unit to a default unit (mole) to ensure it is interpreted as an amount, not a concentration. The imported SimBiology model has the `DimensionalAnalysis` property set to false to prevent dimensional analysis errors.

R2017a

Version: 5.6

New Features

Bug Fixes

Parameter Estimation: Estimate parameters using scatter search global optimization

SimBiology provides a global optimization algorithm called scattersearch to overcome some of the challenges of parameter estimation problems in dynamic models such as convergence to local minima. Specify the estimation function name as `'scattersearch'` when you call `sbiofit`.

SimFunction with Variants: Specify parameters for simulation using variants

You can now use variants as inputs when creating or simulating a `SimFunction object`. Pass in a variant or vector of variants as the last input argument when you call `createSimFunction`, and the variant values are then applied to the model as baseline values. Similarly, when you execute a `SimFunction object`, you can pass in a matrix or cell array of variant objects, and these variant values are used as initial conditions during simulation.

SBML Support: Load SBML models with function definitions, and models that use reaction IDs to reference reaction rates in mathematical expressions

SimBiology lets you load SBML models with function definitions. You can also load SBML models that use reaction IDs to reference reaction rates in mathematical expressions such as assignment rules. For details, see SBML Support.

Performance: Provide faster compilation of SimBiology models and faster simulation of models with large numbers of dosing events

SimBiology takes less time to accelerate models with large numbers of rules or reactions using `sbioaccelerate` or the `accelerate` method of a `SimFunction object` or an exported model (SimBiology.export.Model). The simulation of models with large numbers of dosing events has been improved, and it now takes less time to simulate them.

Diagram View: Navigate between clone blocks, change reactions graphically, and generate variants from estimation and simulation results

In the diagram view of the SimBiology desktop, you can navigate between clone blocks by using the new keyboard shortcuts **N** or **P** keys. You can also update reactions interactively by dragging and dropping existing reaction lines to another species.

You can generate variants directly from the **Simulation** task or **Fit Data** task after it finishes running. The **Variants to Generate** section of the task provides options to generate variants with the final simulated or estimated value of each quantity.

Functionality being changed or removed

Functionality	Result	Use This Instead	Compatibility Considerations
`'UseParallel'` in `sbiofit`	Still runs	-	`sbiofit` now supports several levels of parallelization provided that Parallel Computing Toolbox™ is available. For details, see Multiple Parameter Estimations in Parallel.

R2016b

Version: 5.5

New Features

Bug Fixes

Compatibility Considerations

Noncompartmental Analysis: Handle sparse sampling in addition to serial sampling

The SimBiology desktop lets you perform noncompartmental analysis (NCA) on the measured time-concentration profile data to calculate NCA parameters such as AUC, terminal half-life, and clearance. In cases where longitudinal data on a single subject is not available, the desktop now supports sparse sampling where data collected from multiple subjects are pooled to obtain NCA parameter values. To view NCA parameters, select **Open > NCA** on the **Define Plot** tab after you import the data.

Diagram View: Configure block properties in place with graphical context menus

You can configure block properties using graphical context menus in the diagram view of the SimBiology desktop. Select a block in the diagram, and then hover over the graphical icons for the context menu options.
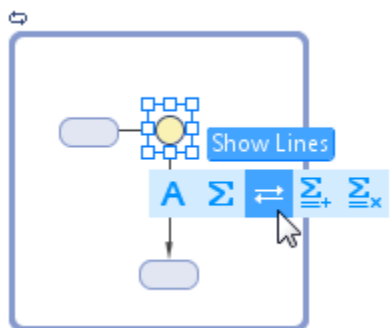
Diagram View: Access model information directly from the graphical view

You can get more information about a model and its components directly from the diagram view. For instance, the diagram shows usages of a quantity by connecting lines from the quantity block to all the expressions that use it. Select **Show Expression Lines** from the context menu of a quantity block. Similarly for the expression blocks, the diagram shows the relationships between quantities and expressions graphically. For example, you can select **Show Expression and Lines** from the graphical context menu of an expression block to see which quantities are being used in the expression. In addition, you can remove unused quantities from the model by selecting **Remove Unused** from the **Model** tab.

Model Usages: Determine model component usage from the command line using findUnusedComponents and findUsages functions

Use the `findUnusedComponents` method of a SimBiology model to find unused species, parameters, and compartments of the model. Use the `findUsages` method of a quantity to look for model components that are using the quantity.

Functionality being removed or changed

Functionality	Result	Use This Instead	Compatibility Considerations
The `Active` property of `configset object`, `variant object`, `ScheduleDose object`, and `RepeatDose object`	Still runs	Explicitly specify the corresponding object as an input argument when you run `sbiosimulate`.	This property of these objects will be removed in a future release.
Load Model Components from File in the SimBiology desktop	Still run	-	If a model component being imported already exists in the model, you can overwrite the property values of the existing component by selecting the **Overwrite current property values** check box. The desktop will not add a duplicate component if one already exists and will only overwrite the property values if the check box is selected.

R2016a

Version: 5.4

New Features

Bug Fixes

Compatibility Considerations

Progress Plot for Fitting: Visualize parameter estimation progress in real time for both desktop and cluster simulations

You can view the progress of parameter estimation while running the Fit task in the SimBiology desktop or using `sbiofit` in the command line. For the desktop, the plots are on by default. For the command line, set the `'ProgressPlot'` option of `sbiofit` to true when you run the function. The plots also keep track of the progress when you perform estimation in parallel using remote clusters. The plots show the log-likelihood, termination criteria, estimated parameters for each iteration, and first-order optimality measure for applicable optimization function.

Model Selection Information: Report parameter estimation quality measures (log-likelihood, AIC, BIC) consistently across all estimation algorithms, allowing direct comparisons between model parameterizations

You can compare different models and select the best fit using the model information criteria such as log-likelihood, Akaike Information Criterion (AIC), and Bayesian Information Criterion (BIC). These values are stored in the `LeastSquaresResults object` and `NLMEResults object` returned by `sbiofit` and `sbiofitmixed`, respectively. This example compares estimation results from different error models. If you are using the desktop, open the **Task Data Comparison** tool to compare these values.

Steady States: Compute model steady states using the sbiosteadystate function, for example to equilibrate a model prior to dosing

You can compute the steady states of a model by using the `sbiosteadystate` function.

SBML Support: Load and save the most recent SBML file format (level 3 version 1)

SimBiology now supports the most recent specification of SBML (level 3 version 1).

Model Examples: Open SimBiology model examples hosted on the MATLAB Central File Exchange from within the SimBiology desktop

To open the hosted SimBiology example projects from within the desktop, select **Content** > **Recent Files**. The projects are listed under the **Example Projects on MATLAB Central File Exchange** section.

Block Property Editor: Display and edit block properties with a single click

To open the **Block Property Editor** in the Diagram view of the SimBiology desktop, double-click a block or select **Home** > **Layout** > **Block Property Editor**. You can then select different blocks with a single click, and the editor updates to display the corresponding block properties.

Functionality being removed or changed

Functionality	Result	Use This Instead	Compatibility Considerations
`ErrorModel` property of `LeastSquaresResults object`, `NLINResults object`, `OptimResults object` and `NLMEResults object`	Errors	`ErrorModelInfo.ErrorModel`	This property has been removed. This property is now a variable of `ErrorModelInfo` table, which is a new property of the results object. `ErrorModelInfo.ErrorModel` is categorical.
`ErrorParameters` property of `LeastSquaresResults object`, `NLINResults object`, `OptimResults object` and `NLMEResults object`	Errors	`ErrorModelInfo.a` and `ErrorModelInfo.b`	This property has been removed. Use `ErrorModelInfo.a` and `ErrorModelInfo.b` instead.
Default output format of `sbionmimport`	Still runs	—	The default output format is now the `groupedData` format. If you need to use a `dataset` instead of a `groupedData` object, convert the output *data* to a dataset using `table2dataset(groupedData2table(data))`.
`'OutputFormat'` name-value pair argument of `sbionmimport`	Errors	If you need to use a `dataset` instead of a `groupedData` object, convert the output *data* to a `dataset` using `table2dataset(groupedData2table(data))`.	The `'OutputFormat'` name-value pair argument has been removed.

R2015b

Version: 5.3

New Features

Bug Fixes

Compatibility Considerations

Application builder for model distribution and simulation

You can create standalone applications for model distribution and simulation from the SimBiology desktop. The application can be an executable that runs outside of MATLAB or a MATLAB app. To deploy a standalone application, you need MATLAB Compiler™.

To build an executable for model simulation, first add a simulation task and open it. Then in the **Task Editor**, on the **Editor** tab, select **Create App > Deploy**. For a MATLAB app, select **Create App > Package as MATLAB App**.

You can include model exploration tools in the application, such as sliders to adjust quantity values and dose schedules, as well as calculated statistics. To test the user interface and see the final deployed application or MATLAB app, select **Create App > Launch**.

Equations view of models

You can view the differential equations and other expressions of a model in the SimBiology desktop. On the **Model** tab, select **Open > Equations**.

Viewing and editing reaction rates in the diagram view

You can view and edit the reaction rate of a reaction in the **Diagram** view of a model. To see the reaction rate, press **Alt** + Click (or **Cmd** + Click for a Macintosh) on a reaction block. You can edit the reaction rate and kinetic law in the expressions view of the **Browser**. To open this view, select **View > By Expressions** on the **Model** tab.

Bioavailability example

This new example shows how to calculate the bioavailability of the drug Ondansetron. For details, see Estimating the Bioavailability of a Drug.

International character support

SimBiology supports non-English characters according to the locale settings of a target machine. For example, a user can enter species names using Japanese characters on a Japanese locale machine without corrupting or losing character data.

DPI-aware SimBiology desktop graphics for Apple Retina displays and high-resolution displays on Windows

The SimBiology desktop looks sharp and properly scaled on all Windows® and Macintosh systems. These improvements are noticeable on the following high-DPI (dots-per-inch) systems:

- Macintosh systems with Apple Retina displays.
- Windows systems in which the DPI (dots-per-inch) value is higher than 96.

Automatic selection of amount and mass units for simulation

SimBiology automatically selects an amount or mass unit that is used internally during simulation when `UnitConversion` is on. This automatic selection is enabled by setting `AmountUnits` and

MassUnits property of an active `configset object` to a string `'<automatic>'`, which is the default.

It is recommended that you use the default unit (`<automatic>`) or choose units for states such that the simulated values are neither too large (greater than 10^6) or too small (less than 10^{-6}).

If you need to recover the simulation behavior from releases prior to R2015b:

- Set the `AmountUnits` to `mole`. However, if the model has quantity units in `molecule`, set the unit to `molecule` instead.
- Set the `MassUnits` to `kilogram`.

Functionality being removed or changed

Functionality	Result	Use This Instead	Compatibility Considerations
`'pso'` as an input argument for `sbiofit` and `sbioparamestim`	Errors	`'particleswarm'`	Replace instances of `'pso'` with `'particleswarm'`.
`ErrorModel` property of `LeastSquaresResults object`, `NLINResults object`, `OptimResults object` and `NLMEResults object`	Warns	`ErrorModelInfo.ErrorModel`	This property will be removed in a future release. This property is now a variable of `ErrorModelInfo` table, which is a new property of the results object. `ErrorModelInfo.ErrorModel` is categorical.
`ErrorParameters` property of `LeastSquaresResults object`, `NLINResults object`, `OptimResults object` and `NLMEResults object`	Warns	`ErrorModelInfo.a` and `ErrorModelInfo.b`	This property will be removed in a future release. Use `ErrorModelInfo.a` and `ErrorModelInfo.b` instead.
Default output format of `sbionmimport`	Still runs	—	The default output format is now the `groupedData` format. If you need to use a `dataset` instead of a `groupedData` object, convert the output *data* to a dataset using `table2dataset(groupedData2table(data))`.

Functionality	Result	Use This Instead	Compatibility Considerations
`'OutputFormat'` name-value pair argument of `sbionmimport`	Warns	If you need to use a `dataset` instead of a `groupedData` object, convert the output *data* to a `dataset` using `table2dataset(groupedData2table(data))`.	The `'OutputFormat'` name-value pair argument will be removed in a future release.

R2015a

Version: 5.2

New Features

Bug Fixes

Compatibility Considerations

Graphical representation of all modeling constructs

The SimBiology desktop includes the **Browser**, a table-like display of model quantities and expressions, to supplement the **Diagram** view of a model. The **Browser** lets you access every quantity and expression of the model and edit them easily.

The block library now has parameter, initial assignment, repeated assignment, rate rule, algebraic rule, event, repeat dose, schedule dose, and variant blocks.

High-accuracy parameter sensitivities enabling more robust parameter optimization

`sbiofit` uses high accuracy sensitivities based on complex step differentiation to generate gradients of objective functions. High accuracy gradients improve parameter estimation with `lsqnonlin`, `lsqcurvefit`, `fmincon`, and `fminunc`.

Response-specific error models

You can specify a separate error model for each response in your model while fitting using `sbiofit` (except for with the `nlinfit` estimation method). For instance, if you have two responses in your model and would like to specify an exponential error model for the first response and a proportional model for the second, use:

```
fitResults = sbiofit(...,'ErrorModel',{'exponential','proportional'};
```

Sensitivity analysis using the SimFunction object

The `SimFunction` object lets you compute sensitivities. Specify the name-value pair arguments `'SensitivityInputs'` and `'SensitivityOutputs'` when you run `createSimFunction` to create the object. Then execute the function object to compute sensitivities.

```
f = createSimFunction(...,'SensitivityInputs','all','SensitivityOutputs','all');
[time,y,sensMatrix] = f(...);
```

Group simulation task enhancements

The **Group Simulation** task has a table for mapping between data and model components in the **Map Between Data and Model** section. You can also overlay results in the **Live Plots** and export individual axis to a figure. The context menu of the **Live Plots** has additional options.

Compartments as input factors for sensitivity analysis

You can specify constant compartments as input factors for sensitivity calculations in addition to species and parameters. Use the `Inputs` property to specify the compartments with respect to which you want to compute the sensitivities.

Functionality being removed or changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
Default output format of `sbionmimport`	Still runs	—	The default output format is now the `groupedData` format. Use the `OutputFormat` name-value pair argument to change the format to the `dataset` format.
`'pso'` as an input argument for `sbiofit` and `sbioparamestim`	Warns	`'particleswarm'`	Replace instances of `'pso'` with `'particleswarm'`.
`ErrorModel` property of `LeastSquaresResults object`, `NLINResults object`, `OptimResults object` and `NLMEResults object`	Still runs	`ErrorModelInfo.ErrorModel`	This property will be removed in a future release. This property is now a variable of `ErrorModelInfo` table, which is a new property of the results object. `ErrorModelInfo.ErrorModel` is categorical.
`ErrorParameters` property of `LeastSquaresResults object`, `NLINResults object`, `OptimResults object` and `NLMEResults object`	Still runs	`ErrorModelInfo.a` and `ErrorModelInfo.b`	This property will be removed in a future release. Use `ErrorModelInfo.a` and `ErrorModelInfo.b` instead.
Sensitivity calculation for models with initial assignment rules using `sbiosimulate`	Still runs	—	Sensitivity calculation with respect to quantities participating in initial assignment rules is now updated to provide correct sensitivities.

R2014b

Version: 5.1

New Features

Bug Fixes

Compatibility Considerations

Estimation of category-specific parameters using the sbiofit function

You can estimate parameters specific to subpopulations which are defined by categories such as sex, age, or height. You can also simultaneously estimate population-wide parameters and individual-specific parameters.

The `CategoryVariableName` property of an `estimatedInfo object` defines the category of a parameter. The default category name is `'<GroupVariableName>'` which means the parameter has a group-specific estimate for each group defined in the data. If `CategoryVariableName` is set to `'<none>'`, then the parameter has a population-wide estimate meaning all groups in the data use the same parameter estimate. The property can also be set to the name of a variable in your data that you intend to categorize by. For example, to estimate sex-specific values of parameter `k1` use:

```
estimatedParam = estimatedInfo('k1','CategoryVariableName','Sex');
results = sbiofit(model,data,...,estimatedParam);
```

where `Sex` is the name of the variable in `data` that categorizes individuals by sex.

Parameter bounds in the sbiofit function

You can specify lower and upper bounds for estimated parameters during fitting using `sbiofit`. Set the `Bounds` property of an `estimatedInfo object` to define the bound constraints. For example, to estimate a parameter `k1` with bound constraints:

```
estimatedParam = estimatedInfo('k1','Bounds',[1 3]);
results = sbiofit(model,data,...,estimatedParam);
```

Desktop enhancements for estimation methods

The fit task of SimBiology desktop supports these estimation methods:

- `'nlinfit'` (Statistics Toolbox™ is required.)
- `'fminsearch'`
- `'lsqcurvefit'` (Optimization Toolbox is required.)
- `'lsqnonlin'` (Optimization Toolbox is required.)
- `'fminunc'` (Optimization Toolbox is required.)
- `'fmincon'` (Optimization Toolbox is required.)
- `'patternserch'` (Global Optimization Toolbox is required.)
- `'ga'` (Global Optimization Toolbox is required.)
- `'particleswarm'` (Global Optimization Toolbox is required.)
- `'nlmefit'` (Statistics Toolbox is required.)
- `'nlmefitsa'` (Statistics Toolbox is required.)

Additionally, you can specify advanced algorithm settings for each estimation method and bound constraints if they are supported by the selected method.

Species, parameter, and compartment names accepted by the StatesToLog property

You can specify a string or cell array of strings containing species, parameter, or compartment names to set the `StatesToLog` property. For example, to log species `x` and `y` simulation data of a SimBiology model `m`:

```
cs = getconfigset(m);
cs.RuntimeOptions.StatesToLog = {'x','y'};
```

Functionality being removed or changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
Default output format of `sbionmimport`	Still runs	—	The default output format will change from the `dataset` to `groupedData` format. Use the `OutputFormat` name-value pair argument to ensure that a `dataset` is returned in future releases.
`'pso'` as an input argument for `sbiofit` and `sbioparamestim`.	Still runs	`'particleswarm'`	Replace instances of `'pso'` with `'particleswarm'`.

R2014a

Version: 5.0

New Features

Bug Fixes

Compatibility Considerations

Desktop enhancements for model and task viewing, exploration, and editing

The SimBiology desktop includes the following enhancements:

- You can open different model pages such as model quantities, expressions, and modifiers at the same time. Each page opens as a new tab.
- A **View** tab allows tiling for better visualization and comparison of different tasks and model pages.
- The **Task Editor** opens in a new window allowing you to view different model pages and task pages at the same time.
- Each task page has three sections, namely **Task**, **Explorer Tools**, and **Live Plots**. The **Task** section describes the details of the task, the **Explorer Tools** section allows you to explore the model's quantities or doses without changing the model, and the **Live Plots** section shows simulation results.
- You can customize generated plots after each task runs as well as **Live Plots** . For example, you can change *x*- and *y*-axes to log scale. To change to log scale on a generated plot, go to the **Plots to Generate** section of a task, and click [icon] under **axesStyle**. To customize **Live Plots**, go to the context menu and select **Properties > Axes Properties > Log scale** for each axis.
- You can scan repeat dose amounts, rates, intervals, and number of doses.
- You can get additional details and tips about the model, tasks, and Explorer Tools using context-sensitive help. To view the content of context-sensitive help, hover your mouse over the information bubble [icon] .

sbiofit and sbiofitmixed functions for parameter estimation

SimBiology provides two additional functions for parameter estimation:

- `sbiofit` for estimating model parameters using nonlinear regression. `sbiofit` can be used with optimization methods from MATLAB, Optimization Toolbox, Global Optimization Toolbox, or Statistics Toolbox. It can be used in place of `sbioparamestim` and `sbionlinfit` to estimate grouped data or pooled data. `sbiofit` allows you to fit all groups using the same parameters (set `'Pooled'` option to `true`) or fit each group separately using group-specific parameters (set `'Pooled'` option to `false`).
- `sbiofitmixed` for estimating model parameters using nonlinear mixed-effects estimation. `sbiofitmixed` can be used in place of `sbionlmefit` and `sbionlmefitsa`, and requires Statistics Toolbox. `sbiofitmixed` supports covariate models.

Both functions support parameter transformations and specification of group-specific dosing. In addition, results objects returned by these functions provide additional plots such as the boxplot of parameter estimates for each parameter, plots of observations vs. predictions, and residual distributions.

Enhanced command-line calling of models for parameter scanning

The SimBiology model object now has a method called `createSimFunction` that creates a `SimFunction object` . In addition, SimBiology introduces two functions `sbiosampleparameters` and `sbiosampleerror`. Together, these enhancements provide:

- An interface for parameter scans from the command line
- A way to sample parameters based on the statistical model and then simulate the model for each parameter sample
- An interface similar to a function handle that can be used with other toolboxes including Curve Fitting Toolbox™ and Control System Toolbox™
- Parallel execution if Parallel Computing Toolbox is available

Name-value pair argument for sbionmimport

`sbionmimport` has an additional name-value pair argument called `'OutputFormat'` that controls the output types from the function. The values are `'dataset'` (default) and `'groupedData'`.

Functionality Being Removed or Changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
sbioparamestim	Still runs	sbiofit	Replace all instances of sbioparamestim with sbiofit.
sbionlinfit	Still runs	sbiofit	Replace all instances of sbionlinfit with sbiofit.
sbionlmefit	Still runs	sbiofitmixed	Replace all instances of sbionlmefit with sbiofitmixed.
sbionlmefitsa	Still runs	sbiofitmixed	Replace all instances of sbionlmefitsa with sbiofitmixed.
PKData object	Still runs	groupedData object	Replace all instances of PKData object with groupedData object.
PKModelMap object	Still runs	estimatedInfo object, CovariateModel object, and sbiodose	Replace all instances of PKModelMap object by using a combination of estimatedInfo object, CovariateModel object, cell array of strings, and sbiodose. See sbiofit and sbiofitmixed for illustrated examples.

R2013b

Version: 4.3.1

New Features

Bug Fixes

Compatibility Considerations

Reordering variants and doses

You can now reorder variants and doses using the `reorder` method.

Renaming reactions

You can now rename any reaction in a model using the `rename` method, and SimBiology updates accordingly any references to reaction-scoped parameters in the model's variants.

Improved support for dimensional analysis of exponents or power expressions

SimBiology now allows exponentiation of any dimensionless quantity to any dimensionless power. For example, you can write the following expression if both `x` and `a` are dimensionless:

`(x + 3)^(a + 0.5)`

Changes to exported SimBiology model dose objects

You can no longer change the `Rate` property of exported dose objects such as `SimBiology.export.RepeatDose` and `SimBiology.export.ScheduleDose` if *all* of these conditions are true:

- The `UnitConversion` property of the model is already set to `true`.
- The `Rate` property is empty or set to 0.
- The `RateUnits` property is set to empty.

These conditions ensure that the `Rate` property has a correct unit for dimensional analysis. To change the `Rate` property, either:

- Use the `Set` function to set the `UnitConversion` property of the original model to `false`. Then use the `export` function to export the model again.
- Set the `RateUnits` property appropriately.

Updates to simulation data after individual or population fitting

After fitting individual or population data, SimBiology now returns simulation data that contains fitted responses evaluated not only at experiment time points, but also at other time points (time steps chosen by an ODE solver).

Functionality being removed or changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
`resample(...,'cubic')` method of the `SimData` object	Warns	`resample(...,'pchip')`	Replace all instances of `resample(...,'cubic')` with `resample(...,'pchip')`.

R2013a

Version: 4.3

New Features

Bug Fixes

Particle swarm optimization for parameter estimation

SimBiology supports the particle swarm optimization (PSO) algorithm for parameter estimation tasks.

You can use this algorithm with the `sbioparamestim` function if you have Global Optimization Toolbox.

Dose and variant control within the Simulation Viewer

You can add, select, or modify doses and variants within the Simulation Viewer. You can create and explore different dose schedules. Sliders for adjusting the desired dose amount, time, and rate are now available.

You can also adjust quantity values of parameters, species, and compartments in a model using the Simulation Viewer without changing the original values in the model.

Speedup of sbionlinfit, sbionlmefit, sbionlmefitsa, and sbioparamestim functions (using Parallel Computing Toolbox)

You can run `sbionlinfit`, `sbionlmefit`, `sbionlmefitsa`, and `sbioparamestim` functions using Parallel Computing Toolbox for faster execution of data fitting tasks.

- When using the command line, you can enable parallel computing by:

  - Opening a MATLAB worker pool: `matlabpool open`
  - Then setting the name-value pair argument `'UseParallel'` of a fitting function to `true`.

- When using the desktop interface, you can enable parallel computing by selecting **Run in Parallel** check box on the **Task** tab for these tasks:

  - **Fit data**
  - **Run scan**
  - **Run scan with sensitivities**

Desktop enhancements for user ordering and grouping of models, tasks, and data

You can reorder tasks, models, and data in the Project page of the SimBiology desktop. For better organization and grouping, you can add folders to **Tasks** and **Data** sections of the project.

General desktop usability improvements

You can sort the order of variants being applied to a task in the SimBiology desktop. You can add a custom stop time for a task by editing **Task Stop Time** in the SimBiology desktop.

18

R2012b

Version: 4.2

New Features

Bug Fixes

Compatibility Considerations

Speedup of simulation of models with multicore processors and computer clusters using Parallel Computing Toolbox

You can now export SimBiology models and simulate the exported model using Parallel Computing Toolbox. To export a model, use the new `export` method.

For an example of using an exported model with Parallel Computing Toolbox, see PK/PD Modeling and Simulation to Guide Dosing Strategy for Antibiotics.

Text display of SimBiology models as a system of differential equations

You can now view a text display of the differential algebraic equation system that SimBiology creates when you build a model using reactions, rules, events, variants, and doses. Use either:

- Command line – Use the `getequations` method of a `Model` object.

- SimBiology desktop – Select **Export Model as Equations** from the ⬇ button when a model is open.

Fit Data Tool and Simulation Viewer to interactively visualize and explore data and models

The SimBiology desktop has a new tool and Simulation Viewer enhancements to improve interactions with data and models:

- **Fit Data Tool** — New tool to explore imported data against multiple models, fit models to data, and compare fit results of multiple models. You can select this tool from the DEFINE PLOT or EXPLORE DATA tabs, which are available after importing external data.

- **Simulation Viewer Enhancements** — Controls for this viewer are now available on a SIMULATION VIEWER tab in the desktop. You can use the controls on this tab to interact with the Simulation Viewer, including viewing sensitivities, adding and removing plots, and removing selected plot lines. Two new plots are available in the Simulation Viewer:

  - **Sensitivity Matrix Plot** — Displays the results of a Sensitivity Analysis task by plotting the sensitivity matrix using the time integral for the sensitivities for the inputs and outputs.

  - **Scan Plot** — Displays the results of a Scan task by plotting the result of a user-specified expression or sensitivity for each simulation run in the scan.

Noncompartmental analysis for characterizing pharmacokinetic data

When importing data using the SimBiology desktop, the desktop now calculates and displays noncompartmental analysis (NCA) parameters in an **NCA** tab, assuming your data includes columns labeled independent variable, dependent variable, and dose. In addition to NCA parameters, statistics that were formerly on the **Statistics** tab (except for mean) now appear on the **NCA** tab.

NCA parameters include AUC, MRT, terminal half life, and clearance, and are useful to obtain initial estimates for parameters used in a **Fit parameters** task. For a full list of NCA parameters that SimBiology calculates, import data, right-click the table in the **NCA** tab, and then select **NCA Parameters to Calculate**.

Deployment of SimBiology models using MATLAB Compiler

You can now export SimBiology models and deploy the exported model using MATLAB Compiler. To export a model, use the new `export` method.

For an example of using an exported model with MATLAB Compiler, see Deploy a SimBiology Model.

SimBiology desktop enhancements

The SimBiology desktop includes the following enhancements:

- **Example Projects** — The desktop Quick Start page now links to projects that are included with the software.
- **EXPLORE DATA tab** — When viewing data in the desktop, options previously available on the EXCLUSION and ADD DATA tabs have been combined and included in the EXPLORE DATA tab.
- **Drag and Drop** — The desktop has additional drag-and-drop functionality:

  - When editing a task, such as fitting or calculating sensitivities, you can drag compartments, species, parameters, and variants from a table in the MODEL tab to a task window.
  - When using the Simulation Viewer, you can drag compartments, species, parameters, and variants from a table in the MODEL tab to a plot in the Simulation Viewer.

Import SBML from URL

You can now import an SBML project into SimBiology from a URL as well as a local file. You must have the Java® programming language to import from a URL. For details, see `sbmlimport`.

Scaled absolute tolerances

For SimBiology models containing species with varying scales or units, you can now improve simulation performance and convergence by enabling absolute tolerance scaling. When enabled, SimBiology scales the absolute error tolerances individually for state values during a simulation. Use either:

- Command line — Use the new `AbsoluteToleranceScaling` property, and, if necessary, the `AbsoluteToleranceStepSize` property of the `SolverOptions` property of a `Configset` object.
- SimBiology desktop — Use the **AbsoluteToleranceScaling** check box, and optionally, the **AbsoluteToleranceStepSize** text box in the Simulation Settings dialog box.

Compatibility Considerations

If you load a model created in SimBiology R2012a or earlier, be aware that:

- The configuration settings now include the `AbsoluteToleranceScaling` property, which is enabled by default.
- For models generated using a `PKModelDesign` object or the PK Wizard, the default value for the `AbsoluteTolerance` property is now `1e-6`, instead of `1e-15`, and the `AbsoluteToleranceScaling` property is enabled.

Enhancements to sensitivity analysis

Performance is improved for model simulation using `sbiosimulate` and model acceleration using `sbioaccelerate`.

Compatibility Considerations

For sensitivity analysis, `sbiosimulate` ignores deterministic `SolverType` and always uses `sundials`.

Log compartment data during simulations

You can now log compartment data during simulations, from the desktop (using the Simulation Settings of a task) or the command line (using the `StatesToLog` property).

Enhancements to simulation results reporting

When simulating a model with multiple reaction-scoped parameters, the reaction names generated by the simulation now include an autonumbered suffix to differentiate between unnamed reactions. For example, UnnamedReaction_1.p and UnnamedReaction_2.p.

Compatibility Considerations

If your model contains multiple reaction-scoped parameters, when you simulate the model in the R2012b software, the reaction names in the simulation results will differ from those returned by simulations run before R2012b. The different reaction names appear in the following outputs:

- *names* output of the `sbiosimulate` function
- `DataInfo` property of the `SimData object`
- *names* output of the `getdata` method
- *Outputs* and *InputFactors* outputs of the `getsensmatrix` method
- *n* output of the `selectbyname` method

Additionally, if your model, created before R2012b, contains multiple reaction-scoped parameters, when you use the `selectbyname` method of the `SimData` object, the *NameValue* input might need to be updated, using the autonumbered reaction names.

Enhancement to writing repeated assignment rules

When writing repeated assignment rules, you no longer need to manually order the rules such that dependent rules follow rules that define their variables. You can list the individual rules in any order, and the software will determine the correct execution order automatically.

Removal of sbiohelp function

Compatibility Considerations

The sbiohelp function has been removed and now errors. Do not use this function.

Functionality being removed

Functionality Being Removed or Changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
sbiohelp function	Errors	—	See the Compatibility Considerations subheading in "Removal of sbiohelp function" on page 18-5.

R2012a

Version: 4.1
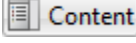
New Features

Bug Fixes

Compatibility Considerations

SimBiology Desktop Enhancements

The SimBiology desktop includes the following enhancements:

- and (back and forward) buttons improve navigation.
- The Simulation Viewer is available to view simulation results during the simulation. It also lets you view results of selected states, tweak values, resimulate, overlay results, and visually compare results to experimental data. Open the Simulation Viewer from the **MODEL** tab or the **Tools** menu in the **HOME** tab.
- Diagram and Custom views now include a **BLOCK** tab and a **TOOLS** tab containing all tools used to build and edit a model diagrammatically, including pin, split, hide, and alignment tools. The Diagram Table View and the Alignment Tool are no longer available from the **Tools** menu in the **HOME** tab.
- The Block Library is now part of the Diagram and Custom views. There is no longer a separate Block Library Browser available from the **Tools** menu in the **HOME** tab.
- You can edit block properties in the Diagram view or Custom view (available from the **MODEL** tab). In either view, click the **Edit Properties** button on the **BLOCK** tab, or right-click a block and select **Properties**. There is no longer a separate Block Property Editor available from the **Tools** menu in the **HOME** tab.
- Filter items in the Component Palette. For example, you can filter parameters to those set by repeated assignment or initial assignment rules.
- Export simulation results to a Microsoft® Excel® file (Windows only).

SBML Level 2 Version 4 Support

SimBiology now supports SBML Level 2 Version 4. For more information, see SimBiology and SBML.

Annotation Property Being Removed

Compatibility Considerations

The Annotation property of the following objects will be removed in a future release. If you try to set the Annotation property, you receive a warning. Set and use the Notes property instead.

- AbstractKineticLaw
- Compartment
- ConfigSet
- Event
- KineticLaw
- Model
- Parameter
- Reaction
- RepeatDose
- Rule
- ScheduleDose

- Species
- Unit
- UnitPrefix
- Variant

Celsius and Fahrenheit Units Being Removed

`Celsius`, `celsius`, and `fahrenheit` units will be removed from the built-in units library in a future release. Use `kelvin` units instead.

Compatibility Considerations

If you have a model containing `Celsius`, `celsius`, or `fahrenheit` units, change the units to `kelvin`.

If you have a script that sets the `ValueUnits` property of a parameter to `'Celsius'`, `'celsius'`, or `'fahrenheit'`, set this property to `'kelvin'` and adjust the `Value` property accordingly.

Enhanced Solver Support

The default for the `SolverType` property has changed from `sundials` to `ode15s`. Possible values for the `SolverType` property have changed and are limited to:

- ode15s
- ode23t
- ode45
- sundials
- ssa
- expltau
- impltau

As a result of the above changes, when simulating models containing events or doses, you are no longer limited to using the SUNDIALS solver:

- When simulating models containing events, you can specify any deterministic (ODE or SUNDIALS) solver or the stochastic `ssa` solver for the `SolverType` property of the `Configset` object.
- When simulating models containing doses, you can specify any deterministic (ODE or SUNDIALS) solver for the `SolverType` property of the `Configset` object.

As a result of the above changes, when performing a parameter estimation or data fitting, you are no longer limited to using the SUNDIALS solver. You can specify any deterministic (ODE or SUNDIALS) solver for the `SolverType` property of the active `Configset` object.

Compatibility Considerations

The following solvers are no longer valid:

- ode23

- ode23s
- ode23tb
- ode113

If you load a model whose `Configset` object specifies one of the above solvers for the `SolverType` property, SimBiology changes the `SolverType` property to `ode15s` (default solver).

Enhanced Simulation Data Logging Support

`OutputTimes` is a new property of the `SolverOptions` property of a `Configset` object. It lets you specify the times during a deterministic (ODE) simulation to log data.

Changes to Simulation Stop Criteria

The `StopTimeType` property of a `Configset` object will be removed in a future release. It is being replaced by the `StopTime` property, which now always specifies simulation time, and two new properties: `MaximumNumberOfLogs` and `MaximumWallClock`. A simulation stops when it meets any of the criteria set by `StopTime`, `MaximumNumberOfLogs`, or `MaximumWallClock`.

Additionally, `OutputTimes` is a new property of the `SolverOptions` property of a `Configset` object. If you set the `OutputTimes` property, its values override the `StopTime` and `MaximumNumberOfLogs` criteria.

Compatibility Considerations

The `StopTimeType` property of a `Configset` object will be removed in a future release. The behavior of the `StopTime` and `StopTimeType` properties has changed. If you load a model whose `Configset` object has a `StopTimeType` property set to `approxWallTime` or `numberOfLogs`, it will change it to `simulationTime` and use the value specified by the `StopTime` property.

Changes to the AbsoluteTolerance Property

The `AbsoluteTolerance` property of the `SolverOptions` property of a `Configset` object has changed. It no longer has an upper limit of 1.

Parameter Estimation and Fitting Enhancements

There are improvements to the parameter estimation and fitting features in SimBiology. As a result:

- Some defaults for the *method* input for the `sbioparamestim` function have changed. Specifically, when using `lsqcurvefit`, `lsqnonlin`, or `fmincon` for the *method*, the default for the `TypicalX` field is `1e-6`*(initial values of components to be estimated).
- The default for the `FunValCheck` name-value pair or the `FunValCheck` field of the `optionStruct` input argument for the `sbionlinfit` function has changed. The default is now `off`. This change allows the optimization to recover from trial parameter values that result in a simulation error.
- The default for the `Options` name-value pair or the `Options` field of the `optionStruct` input argument for the `sbionlmefit` and `sbionlmefitsa` functions has changed. The default of the `FunValCheck` field of the `Options` structure is now `off`. This change allows the optimization to recover from trial parameter values that result in a simulation error.

Compatibility Considerations

In SimBiology Version 4.1, when using the following functions, the defaults of some input arguments have changed as explained above.

- sbioparamestim
- sbionlinfit
- sbionlmefit
- sbionlmefitsa

Weighted Least-Squares Fitting

When performing nonlinear least-squares regression, you can now fit data with nonconstant residual variance by specifying an error model or weights. You can do this by using either the:

- Command line — Use sbionlinfit with either the ErrorModel or Weights name-value pair argument, or specify a value for the ErrorModel or Weights field in the optionStruct input argument.
- SimBiology desktop — Add an individual fit task and use the **Response and Error Model Information** or **Residual Weights** settings.

Simultaneously Fit Data from Multiple Dose Groups

You can now use either nonlinear least-squares regression or nonlinear mixed-effects modeling techniques to estimate a single set of parameters for a data set where only the dosing varies across individuals.

To simultaneously fit data from multiple dose groups using nonlinear regression, use either the:

- Command line — Use sbionlinfit and set the Pooled field or name-value pair input argument to true.
- SimBiology desktop — Add an **individual fit** task and select **Pool Data** in the **Algorithm Settings** section.

To simultaneously fit data from multiple dose groups using nonlinear mixed-effects, use either the:

- Command line — Do either:
  - Use sbionlmefit with a CovariateModel object input argument and omit the random effect (eta) from the expressions in the CovariateModel object.
  - Use sbionlmefit with an InitEstimates input argument and set the REParamsSelect field or name-value pair input argument to a 1-by-n logical vector, with all entries set to false, where n equals the number of fixed effects.
- SimBiology desktop — Add a **population fit** task and omit the random effect (eta) from the expressions in the **Estimated Parameters** section.

Dosing Enhancements

Multiple dose objects in a model can now target the same species.

Covariate Class, Properties, and Methods

The `CovariateModel` class and related properties and methods replace the `PKCovariateModel` class and related properties and methods, to perform covariate analysis for continuous covariates. The `CovariateModel` class lets you specify initial estimates as parameters in the covariate expression, instead of embedding them in the covariate expression. For details, see Specifying a Covariate Model and the demo, `Modeling the Population Pharmacokinetics of Phenobarbital in Neonates`.

The `construct` method of the `PKModelDesign` class returns a `CovariateModel` object with default expression information.

The `PKData` class includes a method, `getCovariateData`, which returns only the covariate data from a data set.

Compatibility Considerations

The `PKCovariateModel` class and its related properties and methods now error. Use the `CovariateModel` class and related properties and methods instead.

Population Fitting Better Supports Covariate Analysis

The `sbionlmefit` and `sbionlmefitsa` functions accept the `CovariateModel` object.

The `sbionlmefit` and `sbionlmefitsa` functions return results that are better annotated. The `results` output structure includes fields with more descriptive names. For details on the new fields, see the `sbionlmefit` and `sbionlmefitsa` reference pages.

Compatibility Considerations

The following syntaxes that accept a `PKCovariateModel` object now error:

results = sbionlmefit(*modelObj*, *pkModelMapObject*, *pkDataObject*, *pkCovModel*)

results = sbionlmefitsa(*modelObj*, *pkModelMapObject*, *pkDataObject*, *pkCovModel*)

Instead, use these syntaxes that accept a `CovariateModel` object:

results = sbionlmefit(*modelObj*, *pkModelMapObject*, *pkDataObject*, *CovModel*)

results = sbionlmefitsa(*modelObj*, *pkModelMapObject*, *pkDataObject*, *CovModel*)

The following fields in the `results` output structure returned by `sbionlmefit` and `sbionlmefitsa` have been removed. Use the new fields instead:

Removed Field	New Field
estimate	FixedEffects
phiP	PopulationParameterEstimates
phiI	IndividualParameterEstimates
beta	FixedEffects

Removed Field	New Field
psi	RandomEffectCovarianceMatrix
b	RandomEffects

Fitting Functions

Compatibility Considerations

The first output argument of `sbionlinfit` is an array of objects instead of an array of structures. The properties of the object are the same as the fields in the previous structure, with these exceptions:

- The `estimate` field has been removed. Instead, use the `ParameterEstimates` property, which includes fitted coefficients and their standard errors.
- There is a new property, `CovarianceMatrix`, which contains the estimated covariance matrix for the fitted coefficients.

Changes to Sensitivity Analysis

You can use parameters as outputs when performing sensitivity analysis.

Compatibility Considerations

The `ParameterInputFactors` and `SpeciesInputFactors` properties of the `SensitivityAnalysisOptions` property of the `Configset` object error. Instead, use the new `Inputs` property.

Also, the `SpeciesOutput` property of the `SensitivityAnalysisOptions` property of the `Configset` object error. Instead, use the new `Outputs` property.

SimData Object Stores Simulation Data

The SimBiology `SimData` object stores the data returned from any simulation.

Compatibility Considerations

The `SimData` object is the container for simulation and analysis task data. Previously, simulation and analysis data were stored as time series objects. Functions that used to return time series objects now return `SimData` objects. Functions that used to take a time series object as an input argument now take a `SimData` object.

Support for time series objects in SimBiology functions has been removed. Although you can load a project containing a time series object, you can no longer use a time series object as an input argument. You must resimulate to create a `SimData` object.

The `sbiogetsensmatrix` and `sbiogetnamedstate` functions are now replaced by the `SimData` object methods `getsensmatrix` and `selectbyname` respectively.

The `sbioupdate` function is now removed.

Submodel Support Removed

Support for submodels has been removed.

Compatibility Considerations

While you can still load a project or model containing submodels, only the top-level model opens. You must manually add the submodel to the top-level model using compartments.

As a result of removing submodels, the following functionality is no longer relevant and has been removed:

- `addmodel` method
- `Models` property of the model object
- `sbioupdate` function

Functionality Being Removed or Changed

Functionality Being Removed or Changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
Annotation property	Warns	Notes property	Set and use the Notes property.
Setting the ValueUnits property of a parameter to 'Celsius', 'celsius', or 'fahrenheit'	Warns	'kelvin'	Set the ValueUnits property to 'kelvin', and adjust the Value property accordingly.
ode23, ode23s, ode23tb, and ode113 values for the SolverType property of the Configset object	Warns	ode15s, ode23t, ode45, or sundials	Change SolverType property to a valid deterministic solver.
simulationTime value for the StopTimeType property of the Configset object	Warns	StopTime property	Set StopTime property of the Configset object.
numberOfLogs and approxWallTime values for the StopTimeType property of the Configset object	Errors	MaximumNumberOfLogs and MaximumWallClock properties	Set MaximumNumberOfLogs and MaximumWallClock properties of the of the Configset object.

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
PKCovariateModel object and related properties and methods	Errors	CovariateModel class and related properties and methods	See the Compatibility Considerations subheading in "Covariate Class, Properties, and Methods" on page 19-6.
ParameterInputFactors and SpeciesInputFactors properties of the SensitivityAnalysisOptions property of the Configset object	Errors	Inputs property of the SensitivityAnalysisOptions property of the Configset object	Replace all instances of ParameterInputFactors and SpeciesInputFactors with Inputs.
SpeciesOutput property of the SensitivityAnalysisOptions property of the Configset object	Errors	Outputs property of the SensitivityAnalysisOptions property of the Configset object	Replace all instances of SpeciesOutput with Outputs.
estimate and beta fields in the results structure returned by sbionlmefit or sbionlmefitsa	Errors	FixedEffects field	See the Compatibility Considerations subheading in "Population Fitting Better Supports Covariate Analysis" on page 19-6.

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
phiP field in the results structure returned by sbionlmefit or sbionlmefitsa	Errors	PopulationParameter Estimates field	See the Compatibility Considerations subheading in "Population Fitting Better Supports Covariate Analysis" on page 19-6.
phiI field in the results structure returned by sbionlmefit or sbionlmefitsa	Errors	IndividualParameter Estimates field	See the Compatibility Considerations subheading in "Population Fitting Better Supports Covariate Analysis" on page 19-6.
psi field in the results structure returned by sbionlmefit or sbionlmefitsa	Errors	RandomEffectCovaria nceMatrix field	See the Compatibility Considerations subheading in "Population Fitting Better Supports Covariate Analysis" on page 19-6.

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
b field in the results structure returned by sbionlmefit or sbionlmefitsa	Errors	RandomEffects field	See the Compatibility Considerations subheading in "Population Fitting Better Supports Covariate Analysis" on page 19-6.
estimate field in the results structure returned by sbionlinfit	Errors	ParameterEstimates property of the results object returned by sbionlinfit	See the Compatibility Considerations subheading in "Fitting Functions" on page 19-7.

R2011b

Version: 4.0

New Features

Bug Fixes

Compatibility Considerations

Redesigned and Enhanced Desktop

The SimBiology desktop is redesigned to better support streamlined, iterative workflows. It no longer includes a Project Explorer to navigate between models, tasks, and data. Now, use the following management and navigation aids for projects and libraries:

• **Toolstrip** — Displays the **HOME** tab and other tabs depending on what you select in the address bar.

• **HOME tab** — Open and manage projects, including adding models, analysis tasks, and data to a project.

• **Address bar** — Select a project, model, task, or data to view and edit in the desktop window. Navigate between models, tasks, and data. Select and view libraries.

• **Actions button** ⊙ — Use to select actions appropriate for the item (project, model, task, data, or library) selected and displayed in the desktop.

Tip Selecting Project in the address bar displays all models, tasks, and data included in the project.

Use HOME tab to add models, tasks, and data to project

Toolstrip

Use address bar to select a project, model, task, data, or library

Use Actions button to select actions for project, model, task, data, or library
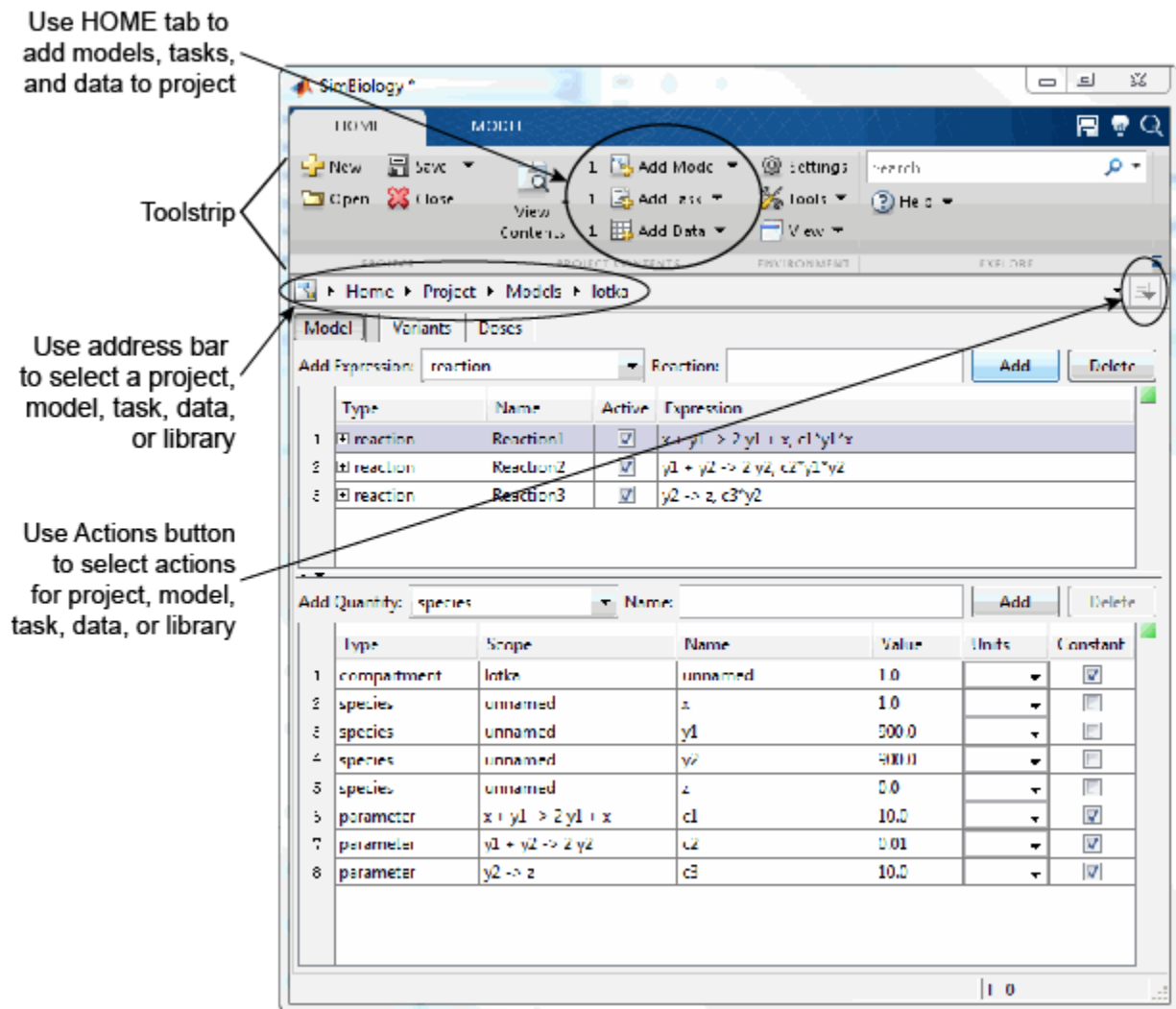
The **MODEL**, **TASK**, and **DATA** tabs better support the following iterative workflows:

- **Model-centric workflow** — Refine a model by performing iterative edits on the model. Without leaving the model, run multiple tasks and view results to help guide you with your model edits.

- **Task-centric workflow** — Refine an analysis task by performing iterative runs and edits on the task. Without leaving the task, select different models to run the task on. For example, select different models for a parameter fit task to find the model that best fits the data.

- **Data-centric workflow** — Plot and explore data, all from the DATA tabs.

For more information, see Getting Started Using the SimBiology Desktop.

Open Projects in the Desktop from the Command Line

The `simbiology` function is updated with a new syntax that lets you open projects (sbproj files) in the SimBiology desktop from the command line.

New Covariate Class, Properties, and Methods

There is a new `CovariateModel` class and related properties and methods that replace the existing `PKCovariateModel` class and related properties and methods, to perform covariate analysis for continuous covariates. The new `CovariateModel` class lets you specify initial estimates as parameters in the covariate expression, instead of embedding them in the covariate expression. For details, see Specifying a Covariate Model and the demo, `Modeling the Population Pharmacokinetics of Phenobarbital in Neonates`.

The `construct` method of the `PKModelDesign` class now returns a `CovariateModel` object with default expression information.

The `PKData` class includes a new method, `getCovariateData`, which returns only the covariate data from a data set.

Compatibility Considerations

The `PKCovariateModel` class and its related properties and methods now return warnings and will be removed in a future release. Use the `CovariateModel` class and related properties and methods instead.

Population Fitting Better Supports Covariate Analysis

The `sbionlmefit` and `sbionlmefitsa` functions accept the new `CovariateModel` object.

The `sbionlmefit` and `sbionlmefitsa` functions now return results that are better annotated. The `results` output structure now includes new fields with more descriptive names. For details on the new fields, see the `sbionlmefit` and `sbionlmefitsa` reference pages.

Compatibility Considerations

The following syntaxes that accept a `PKCovariateModel` object will be removed in a future release:

results = sbionlmefit(*modelObj*, *pkModelMapObject*, *pkDataObject*, *pkCovModel*)

results = sbionlmefitsa(*modelObj*, *pkModelMapObject*, *pkDataObject*, *pkCovModel*)

Instead, use these syntaxes that accept a `CovariateModel` object:

results = sbionlmefit(*modelObj*, *pkModelMapObject*, *pkDataObject*, *CovModel*)

results = sbionlmefitsa(*modelObj*, *pkModelMapObject*, *pkDataObject*, *CovModel*)

The following fields in the `results` output structure returned by `sbionlmefit` and `sbionlmefitsa` will be removed in a future release. Use the new fields instead:

Removed Field	New Field
estimate	FixedEffects
phiP	PopulationParameterEstimates
phiI	IndividualParameterEstimates

Removed Field	New Field
beta	FixedEffects
psi	RandomEffectCovarianceMatrix
b	RandomEffects

Support for the following name-value pair arguments used by `sbionlmefit` and `sbionlmefitsa` has changed:

- `FEGroupDesign` — You can no longer set this argument. It is computed from the covariate model.
- `ParamTransform` — If using the syntax that accepts a `CovariateModel` object, you can no longer set this argument. It is computed from the covariate model.
- `REParamsSelect` — If using the syntax that accepts a `CovariateModel` object, you can no longer set this argument. It is computed from the covariate model.

Fitting Functions Enhancements

The following fitting functions now return standard errors. These functions also now run faster when estimating parameters of most common PK models:

- `sbionlinfit` — Perform nonlinear least-squares regression using SimBiology models
- `sbionlmefit` — Estimate nonlinear mixed effects using SimBiology models
- `sbionlmefitsa` — Estimate nonlinear mixed effects with stochastic EM algorithm

Compatibility Considerations

The first output argument of `sbionlinfit` is now an array of objects instead of an array of structures. The properties of the object are the same as the fields in the previous structure, with the following exceptions:

- The `estimate` field will be removed in a future release. Instead, use the `ParameterEstimates` property, which includes fitted coefficients and their standard errors.
- There is a new property, `CovarianceMatrix`, which contains the estimated covariance matrix for the fitted coefficients.

Parameter Estimation Enhancements

The `sbioparamestim` function is more robust and accurate due to the addition of the `FinDiffRelStep` field in the *method* input argument.

Compatibility Considerations

The *method* input for the `sbioparamestim` function has changed. Most important, when *method* uses the `lscurvefit`, `lsqnonlin`, or `fmincon` function, there is an additional field, `FinDiffRelStep`. For details on this new field, see the `sbioparamestim` reference page.

Also, when *method* uses the `lscurvefit` or `lsqnonlin` function, the `DiffMinChange` field now uses the same defaults as `lscurvefit` or `lsqnonlin`. Likewise, when *method* uses the `fmincon` function, the `TypicalX` field now uses the same default as `fmincon`.

Enhancements to Sensitivity Analysis

Models containing repeated assignment rules, rate rules, or doses now support sensitivity analysis. Also, you can now use parameters as outputs when performing sensitivity analysis.

Compatibility Considerations

The `ParameterInputFactors` and `SpeciesInputFactors` properties of the `SensitivityAnalysisOptions` property of the `Configset` object will be removed in a future release. Instead, use the new `Inputs` property.

Also, the `SpeciesOutput` property of the `SensitivityAnalysisOptions` property of the `Configset` object will be removed in a future release. Instead, use the new `Outputs` property.

Additional Syntax Checking of Reaction Rates, Rule Assignments, Event Triggers, and Event Functions

Model verification and simulation now perform additional syntax checks on reaction rates, rule assignments, event triggers, and event functions. These checks ensure all expressions simulate properly.

Compatibility Considerations

If an expression for a reaction rate, rule assignment, event trigger, or event function ends in any of the following, it now returns an error during verification or simulation:

- Semicolon
- Comma
- Comment text preceded by %
- Line continuations indicated by ...

Annotation Property Being Removed

Compatibility Considerations

The `Annotation` property of the following objects will be removed in a future version:

- `AbstractKineticLaw`
- `Compartment`
- `ConfigSet`
- `Event`
- `KineticLaw`
- `Model`
- `Parameter`
- `Reaction`
- `RepeatDose`

- Rule
- ScheduleDose
- Species
- Unit

Function Elements Being Removed

Function Elements Being Removed

Function Element Name	What Happens When You Use This Function Element	Use This Instead	Compatibility Considerations
PKCovariateModel object and related properties and methods	Warns	CovariateModel class and related properties and methods	See the Compatibility Considerations subheading in "New Covariate Class, Properties, and Methods" on page 20-4.
estimate and beta fields in the results structure returned by sbionlmefit or sbionlmefitsa	Warns	FixedEffects field	See the Compatibility Considerations subheading in "Population Fitting Better Supports Covariate Analysis" on page 20-4.
phiP field in the results structure returned by sbionlmefit or sbionlmefitsa	Warns	PopulationParameterEstimates field	See the Compatibility Considerations subheading in "Population Fitting Better Supports Covariate Analysis" on page 20-4.
phiI field in the results structure returned by sbionlmefit or sbionlmefitsa	Warns	IndividualParameterEstimates field	See the Compatibility Considerations subheading in "Population Fitting Better Supports Covariate Analysis" on page 20-4.

Function Element Name	What Happens When You Use This Function Element	Use This Instead	Compatibility Considerations
`psi` field in the `results` structure returned by `sbionlmefit` or `sbionlmefitsa`	Warns	`RandomEffectCovarianceMatrix` field	See the Compatibility Considerations subheading in "Population Fitting Better Supports Covariate Analysis" on page 20-4.
`b` field in the `results` structure returned by `sbionlmefit` or `sbionlmefitsa`	Warns	`RandomEffects` field	See the Compatibility Considerations subheading in "Population Fitting Better Supports Covariate Analysis" on page 20-4.
`estimate` field in the `results` structure returned by `sbionlinfit`	Warns	`ParameterEstimates` property of the `results` object returned by `sbionlinfit`	See the Compatibility Considerations subheading in "Fitting Functions Enhancements" on page 20-5.
Annotation property of SimBiology objects	Warns	—	See the Compatibility Considerations subheading in "Annotation Property Being Removed" on page 20-6.

Function Element Name	What Happens When You Use This Function Element	Use This Instead	Compatibility Considerations
ParameterInputFactors and SpeciesInputFactors properties of the SensitivityAnalysisOptions property of the Configset object	Warns	Inputs property of the SensitivityAnalysisOptions property of the Configset object	See the Compatibility Considerations subheading in "Enhancements to Sensitivity Analysis" on page 20-6.
SpeciesOutput property of the SensitivityAnalysisOptions property of the Configset object	Warns	Outputs property of the SensitivityAnalysisOptions property of the Configset object	See the Compatibility Considerations subheading in "Enhancements to Sensitivity Analysis" on page 20-6.

R2011a

Version: 3.4

New Features

Bug Fixes

Compatibility Considerations

Redesigned and Enhanced Desktop

The SimBiology desktop is redesigned to support:

- Streamlined model-building workflow
- Iterative workflows
- Comparison of analysis results

The **Project Explorer** (outlined in *red*) is organized by Models, Tasks, and Data. After selecting a model, task, or data in the **Project Explorer**, then view and set related information in the **Work Area** (outlined in *blue*).
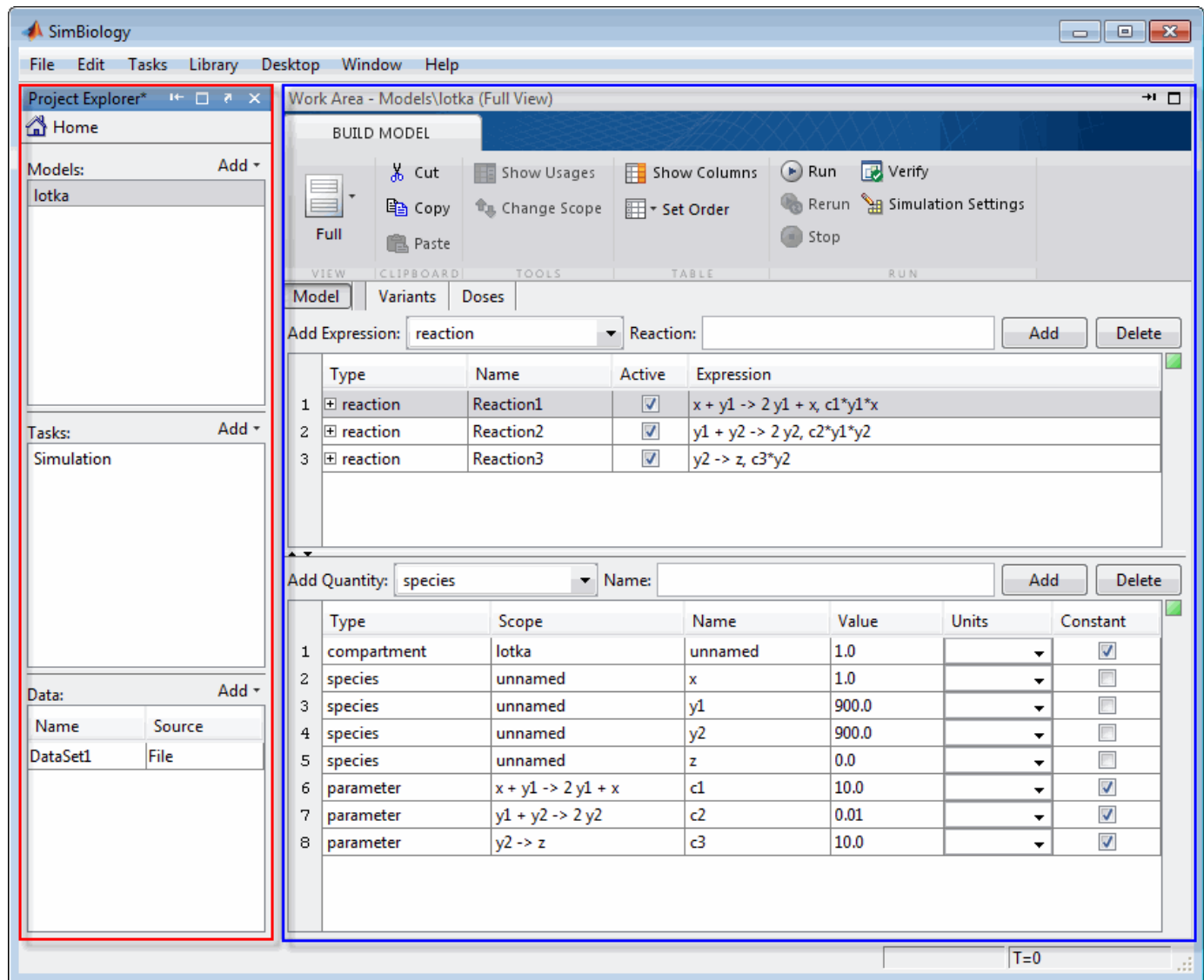
The desktop now lets you:

- Use Full and Custom views for model building, in addition to the Table and Diagram views.

- Create a task once, then run it on multiple models.
- View and compare a summary of results from running like analysis tasks on one or more models.

Note Because the SimBiology desktop's more intuitive design better supports the model building and analyzing workflow, the context-sensitive help was removed from the desktop.

Support of Arbitrary Time Units

SimBiology now supports arbitrary time units for simulation time, and is no longer limited to seconds. The default value of the `TimeUnits` property of a `Configset` object is `second` for a `Model` object created using `sbiomodel`, and `hour` for a `Model` object created from a `PKModelDesign` object. For details, see the `TimeUnits` property reference page.

Parameter Estimation and Population Fitting Enhancements

Following is a new function:
`sbiofittool` — Open SimBiology desktop for population fitting.

There are improvements to the parameter estimation and population fitting features in SimBiology:

- When performing a parameter estimation, you can now specify nonconstant parameter quantities (in addition to species amounts) to be matched during estimation. For details, see the description of *observed_array* in the `sbioparamestim` reference page.
- When performing a parameter estimation, you can now specify compartment capacities and species amounts (in addition to parameter quantities) to be estimated. For details, see the description of *estimated_array* in the `sbioparamestim` reference page.
- Some defaults for the *method* input for the `sbioparamestim` function have changed. Most defaults are the same as the defaults associated with the function selected as *method*. However, some defaults differ. For details, see the `sbioparamestim` reference page.
- The defaults for the *optionStruct* input for the `sbionlmefit` and `sbionlmefitsa` functions have changed. The defaults for *optionStruct* are the same as the defaults for the name-value pair arguments accepted by the `nlmefit` and `nlmefitsa` functions, except:

  - The `OptimFun` argument's default is `fminunc`, if you have Optimization Toolbox installed. Otherwise, the default is still `fminsearch`.
  - The `DerivStep` field of the `Options` name-value pair argument. The default of the `DerivStep` field, when used by `sbionlmefit` or `sbionlmefitsa`, is the lesser of `1e-4`, or the value of the `SolverOptions.RelativeTolerance` property of the configuration set associated with *modelObj*, with a minimum of `eps^(1/3)`.

- The default for the `DerivStep` field for the `sbionlinfit` function has changed. The default is now is the lesser of `1e-4`, or the value of the `SolverOptions.RelativeTolerance` property of the configuration set associated with *modelObj*, with a minimum of `eps^(1/3)`.
- When using `sbionlinfit` to perform an individual fitting, if the fitting for an individual fails, the function returns a warning indicating the individual that failed, and then continues with the fitting, instead of returning an error and stopping.

Compatibility Considerations

In SimBiology Version 3.4, when using the following functions, the defaults of some input arguments have changed as explained above.

- sbioparamestim
- sbionlmefit
- sbionlmefitsa
- sbionlinfit

Calculation of Weighted Residuals for Population Fitting Tasks

SimBiology now calculates individual weighted residuals, population weighted residuals, and conditional weighted residuals when performing population fitting.

Increased Performance When Using SUNDIALS Solvers

Simulations of large models using SUNDIALS solvers now run faster.

Evaluation of Simultaneous Events

When multiple events are triggered simultaneously, the event functions execute sequentially, in the order the events are listed in the model.

Compatibility Considerations

In SimBiology Version 3.3 and earlier, when multiple events were triggered simultaneously, each event function executed independently, at the same time, using the current state of the model. Then the results of the independently executed events merged in the order the events were listed in the model.

In SimBiology Version 3.4, when multiple events are triggered simultaneously, the event functions execute sequentially, in the order the events are listed in the model. This change impacts event functions that change model components in a dependent fashion. For details, see Evaluation of Simultaneous Events.

New and Updated Demos

Following are new demos:

- Simulating the Glucose-Insulin Response
- Monte Carlo Simulation of a PK/PD Model for an Antibacterial Agent

The Parameter Scanning, Parameter Estimation, and Sensitivity Analysis in the Yeast Heterotrimeric G Protein Cycle demo is updated to take advantage of some of the parameter estimation and population fitting enhancements on page 21-3.

R2010b

Version: 3.3

New Features

Bug Fixes

Support for Error Models Using sbionlmefit

Parameter fitting functionality now supports the following error models:

- `constant`
- `proportional`
- `combined`
- `exponential`

You can specify an error term when performing population fitting using the `sbionlmefit` or `sbionlmefitsa` function.

Support for Covariate Analysis

With the introduction of the `PKCovariateModel` object and related methods, you can now perform covariate analysis for continuous covariates.

If you use the command line, see the following section for more information:

- Specifying a Covariate Model

If you use the SimBiology desktop, see the following section for more information:

- Specifying a Covariate Model in the SimBiology Desktop

Support for Multiple Response Fitting

SimBiology now supports multiple responses when performing a population or individual fit task.

If you use the command line, see the following sections for more information:

- Creating a Pharmacokinetic Model Using the Command Line
- Performing Data Fitting with PKPD Models

If you use the SimBiology desktop, see the following sections for more information:

- Creating PK Models in the SimBiology Desktop Using a Wizard
- Fitting Pharmacokinetic Model Parameters in the SimBiology Desktop

Support for Time Lags

SimBiology now supports time lags for doses used in a simulation or fit (population or individual) task.

If you use the command line, see the following sections for more information:

- Creating a Pharmacokinetic Model Using the Command Line
- About Dosing Types
- Performing Data Fitting with PKPD Models

If you use the SimBiology desktop, see the following sections for more information:

- Creating PK Models in the SimBiology Desktop Using a Wizard
- About Dosing Types
- Fitting Pharmacokinetic Model Parameters in the SimBiology Desktop

Support for Dimensionless Units

SimBiology now supports dimensionless units when performing dimensional analysis and unit conversion. You specify dimensionless units either via the command line (by setting the `ValueUnits` property of a parameter object to `'dimensionless'`) or in the desktop (by selecting `dimensionless` for the **ValueUnits** in the Parameters pane).

Modeling, Simulation, and Analysis Tools

The following function is updated:

- `sbiotrellis` — Plot simulation results in trellis plot. Updated to accept multiple columns from the data set to plot on the *y*-axis.

Parameter Scan Subplots Display Parameter Information

After performing a parameter scan, the plot window now displays the species, parameters, and compartments associated with each subplot. Hover the mouse cursor over a subplot to display this information at the bottom of the window.
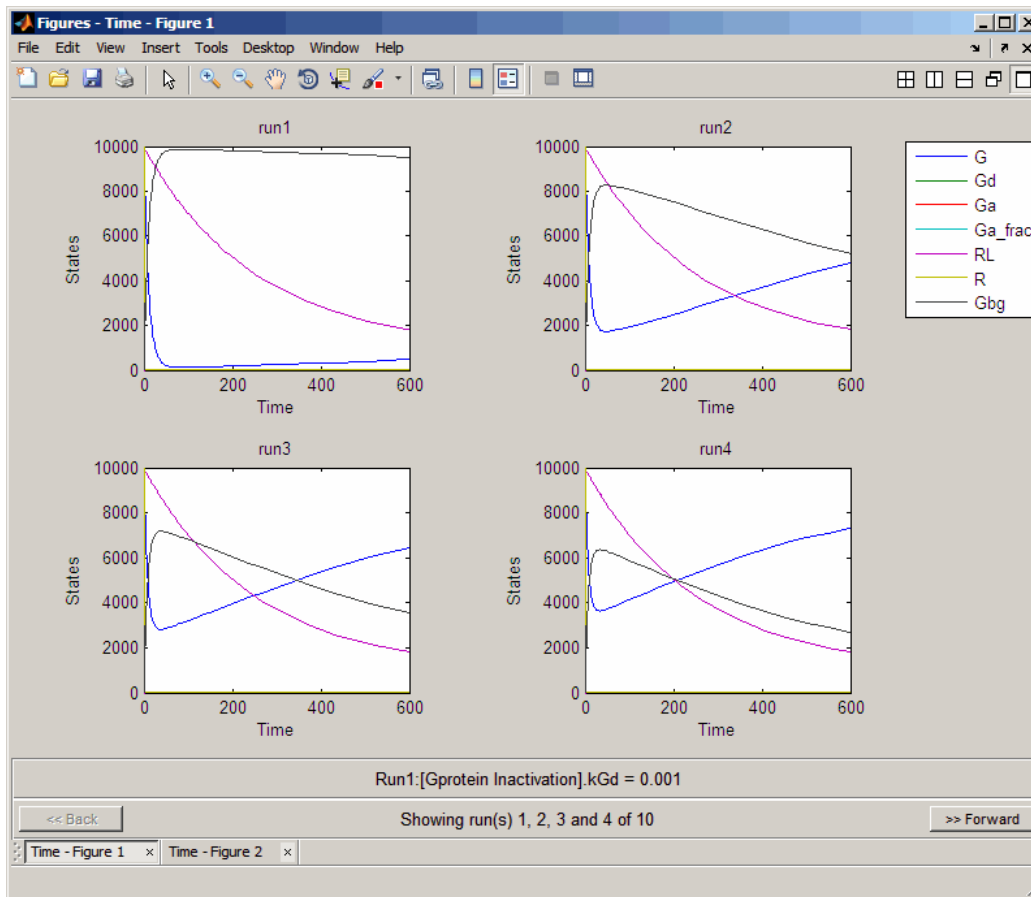
Removal of the Export Tab

The **Export** tab is removed from all Model Task panes. By default, data from these tasks automatically exports to the MATLAB workspace. To disable data exportation, select **File > Preferences > Model Tasks**, then clear the **Export task data when task completes executing** check box. To export when exporting is disabled, or to export using settings other than the default, in the Project Explorer pane, select the **Data** node under a specific Model Task, then in the **Data** pane,

click ![Export] to open the Export Data dialog box.

Demos for Modeling

Following is a new modeling demo:
`Deterministic Simulation of a Model Containing a Discontinuity`

R2010a

Version: 3.2

New Features

Bug Fixes

Compatibility Considerations

Stochastic Approximation Expectation-Maximization (SAEM) Algorithm for Fitting Population Data

Now you can choose the SAEM algorithm when fitting population data. This functionality requires Statistics Toolbox (Version 7.3 or later).

The new stochastic algorithm for fitting NLME models is more robust with respect to starting values, enables parameter transformations, and relaxes assumption of constant error variance.

For more information, see:

- `sbionlmefitsa`
- `nlmefitsa`
- Pharmacokinetic Modeling Functionality

Enhanced Support for Importing NONMEM Formatted Files

Import data files with NONMEM® interpretation of column headers. SimBiology interprets the data file during import and creates the data set to use during fitting. For more information see Importing Data — Supported Files and Data Types. After import you can also create dose schedules using the information in the imported data.

New Mode for Accelerating Simulations

SimBiology enables you to prepare your models for accelerated simulations. Use this functionality to run many simulations with different initial conditions, or to run very long simulations (for example, simulations that take a minute or longer to run). Before you can use this feature you must install a C compiler, and run `mex -setup` before you can use this feature. For more information see Accelerating Model Simulations and Analyses.

Enhanced Support for Applying Dosing to a Model and Dosing Multiple Compartments

Create and apply dosing using `RepeatDose Object`, `ScheduleDose Object` and the `adddose` method at the command line or the **Doses** pane in the desktop.
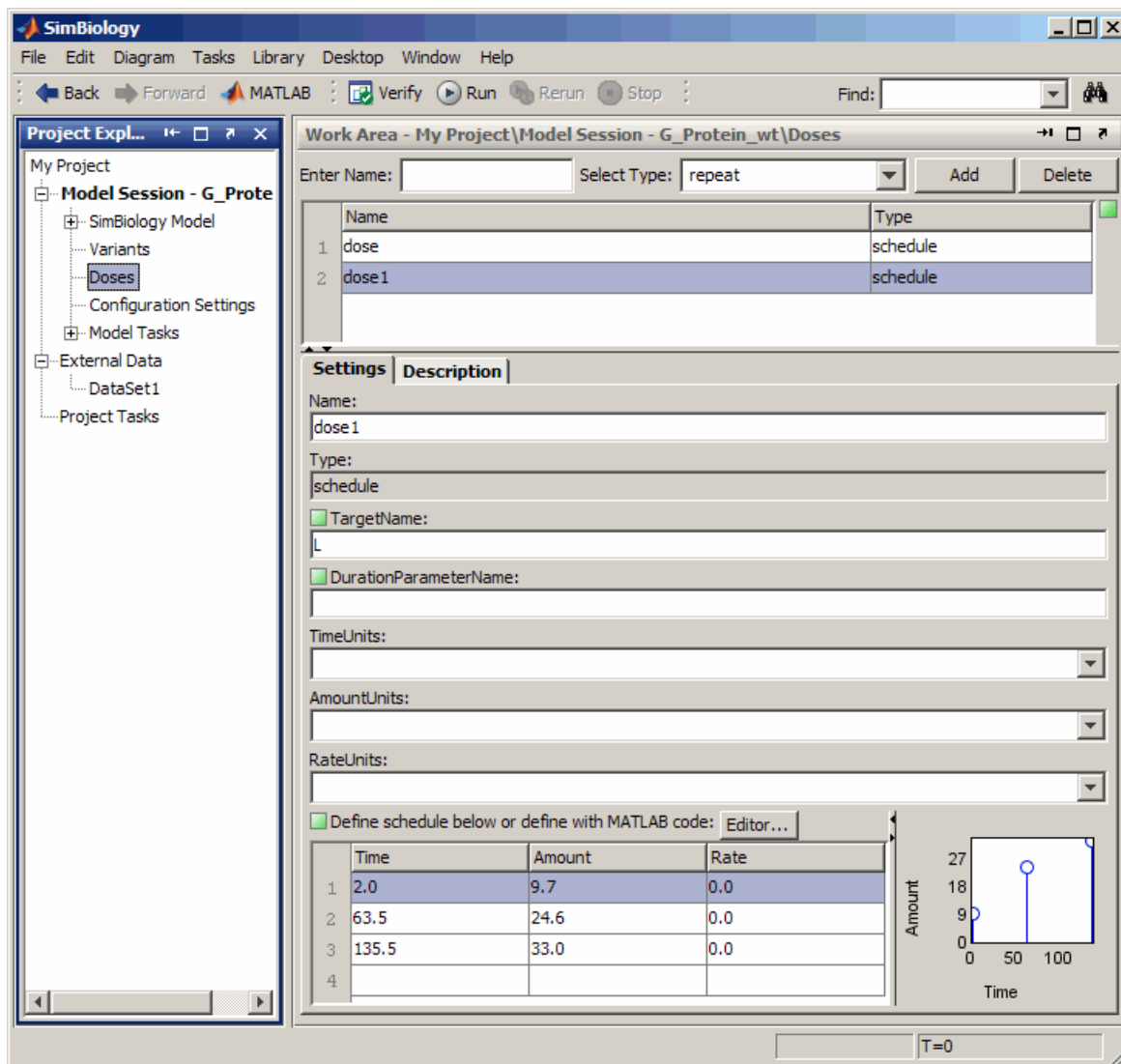
Compatibility Considerations

- Previously, simulating models with dosing information required the `sbiosetdosingprofile` function. Using `sbiosetdosingprofile` now errors and you must change how you apply dosing. For related information on dosing in pharmacokinetic models see About Dosing Types.

- Previously, you could specify that a parameter is dosed. Now only species can accept a dose.

- Previously, the PK models you created using the **New Project Wizard** or the `construct` method varied depending on the dose chosen. Now you get the same model, which allows you to change between dosing types.

Support for Parameter Transformations

During parameter fitting, you now can specify parameter transformations. The following parameter transformations are now supported:

- none
- log
- probit
- logit

You can specify parameter transformations in individual (`sbionlinfit`) and population fitting (`sbionlmefit` or `sbionlmefitsa`) functions .

Compatibility Considerations

Previously, `sbionlinfit` and `sbionlmefit` returned the log-transformed estimates for the fixed effects. Now `sbionlinfit`, `sbionlmefit` (and `sbionlmefitsa`) return untransformed and transformed estimates for the fixed effects.

Support for Error Models

Parameter fitting functionality now supports the following error models:

- constant
- proportional
- combined
- exponential

You can specify an error term in conjunction with a population fitting (`sbionlmefitsa`) function.

Functions and Properties Being Removed

Function or Property Name	What Happens When You Use Function or Property?	Use This Instead	Compatibility Considerations
sbiosetdosingprofile	Errors	RepeatDose Object, ScheduleDose Object, adddose	See the Compatibility Considerations subheading in"Enhanced Support for Applying Dosing to a Model and Dosing Multiple Compartments" on page 23-2.

R2009b

Version: 3.1

New Features

Bug Fixes

Compatibility Considerations

Increased Performance When Repeatedly Simulating a Model

Many analysis tasks that involve repeatedly simulating a model now run faster. These tasks include parameter fits and scans, as well as repeatedly simulating the same model using different variants or setting different values for the `InitialAmount` of species, the `Capacity` of compartments, and the `Value` of parameters. Under these conditions, repeatedly simulating the model generates any applicable warnings only the first time. To display warnings again, use the verification methods described in Verifying That a Model Has No Warnings or Errors in the SimBiology User's Guide.

Enhanced Desktop Support for Scanning Using Monte Carlo Methods

Scanning analysis now includes additional support for Monte Carlo methods. You can specify sampling using the options `multivariate normal distribution` or `latin hypercube sample with a normal distribution`. The Statistics Toolbox is required for this functionality. For more information, see Scanning Analysis in the SimBiology User's Guide documentation.

Desktop Support for Copy and Paste

The SimBiology desktop now supports copying and pasting. Use the typical keyboard shortcuts or the context menus to execute these commands.

View Status of Parameter Fitting Task During Run

You can track the status of a parameter fit when using nonlinear mixed effects with the `sbiofitstatusplot` function at the command line or by selecting an option in the SimBiology desktop.

Improved Usability for Model Building and Debugging

The SimBiology desktop now supports:

- Dynamically updated error indicators for variants, configuring plots, and defining scans — Shows correctly defined, incorrectly defined, and warning indicators (green, red, and yellow) for additional help with model debugging.
- M-Lint indicators — Wherever code appears in the desktop, the indicators used by M-Lint also appear.
- Help for user-identified MATLAB code — Select and use the context menu to find help on functions where code appears in the desktop.

Unit Conversion Compatibility Considerations

Previously a model with no units specified could have unit conversion on or off. Now, a model with no units specified that has `UnitConversion` on shows an error. Set `UnitConversion` off for models without units.

Functions and Properties Being Removed

Function or Property Name	What Happens When You Use Function or Property?	Use This Instead	Compatibility Considerations
`-flat` as an option for `getstoichmatrix` and `getadjacencymatrix`	Errors	Not applicable	Since support for submodels has been removed on page 28-2, this option no longer applies.

R2009a

Version: 3.0

New Features

Bug Fixes

New Feature to Import, Visualize, and Statistically Analyze Clinical and Experimental Data

You can import tabular data into the SimBiology desktop or the MATLAB Workspace. The supported file types are `.xls`, `.csv`, and `.txt`.

At the command line, you can process and visualize the data using command-line functions. In the SimBiology desktop, you can filter the raw data to suppress outliers, visualize data using MATLAB plots, and calculate statistics to analyze the data. You can further choose to plot the imported data with any analysis task.

See Importing Data — Supported Files and Data Types in the SimBiology User's Guide for more information.

New Functionality to Create Pharmacokinetic Models

You can automatically generate pharmacokinetic (PK) models by specifying number of compartments, dosing type, and method of elimination. If you plan to use the MATLAB command line, see Creating a Pharmacokinetic Model Using the Command Line in the SimBiology User's Guide for more information.

If you plan to use the SimBiology desktop, the new Add Model wizard lets you automatically generate PK models in the desktop. See Creating PK Models in the SimBiology Desktop Using a Wizard in the SimBiology User's Guide for more information.

In addition, in the SimBiology desktop you can start by creating a new project using the new Project Wizard which also lets you add data, create models, and add analysis tasks in a SimBiology project. See "New Project Wizard to Add Data, Create Models, and Specify Tasks" on page 25-3 for additional information.

New Functionality to Fit Data and Estimate Parameters Using Nonlinear Mixed Effects

You can perform both individual and population fits to grouped longitudinal data.

- Individual fit — Fit data using nonlinear least squares method, estimate parameters, and calculate residuals and the estimated coefficient covariance matrix.
- Population fit — Estimate the fixed effects and the random sources of variation on parameters, using nonlinear mixed-effects models.

  You can use the following methods to estimate the fixed effects:

  - `LME` — Linear mixed-effects approximation
  - `RELME` — Restricted LME approximation
  - `FO` — First-order estimate
  - `FOCE` — First-order conditional estimate

For more information, see Pharmacokinetic Modeling Functionality in the SimBiology User's Guide.

New Diagnostic Plots for Individual and Population Fitting Results

In the SimBiology desktop, after fitting the data, the analysis generates diagnostic plots that show:

- The predicted time courses and observations for an individual or the population
- Observed versus predicted values
- Residuals versus time, group, or predictions
- Distribution of the residuals
- A box-plot for random effects or parameter estimates from individual fitting.

For more information, see Visualizing Parameter Fitting Results and Generating Diagnostic Plots in the SimBiology User's Guide.

New Project Wizard to Add Data, Create Models, and Specify Tasks

The newly added Project Wizard in the SimBiology desktop lets you:

- Add data from text files, spreadsheets, or the MATLAB Workspace.
- Create models, including automatically generate pharmacokinetic models by specifying number of compartments, dosing type, and method of elimination.
- Specify analysis tasks to add to the project.

For an example of how to use the Project Wizard, see Modeling Using the SimBiology Graphical User Interface in the SimBiology Getting Started Guide.

New simbiology Command to Open the SimBiology Desktop

A new function, `simbiology`, has been added for enhanced usability in opening the SimBiology desktop. `simbiology` is equivalent to the `sbiodesktop` function, which is also supported.

Enhanced Usability Features in the SimBiology Desktop

The following enhancements to the SimBiology desktop are included in this release:

- **Back** and **Forward** buttons to help with navigation between desktop panes
- In **Preferences**, the ability to choose default model tasks to add to a model when loading an SBML file or importing a model from the MATLAB Workspace

New Demo for Pharmacokinetic Modeling

There is a new demo showing pharmacokinetic modeling functionality (`Modeling the Population Pharmacokinetics of Phenobarbital in Neonates`). To see all demos, click `SimBiology demos` or type `demo('MATLAB', 'SimBiology')` at the command prompt.

R2008b

Version: 2.4

New Features

Bug Fixes

Compatibility Considerations

Enhanced Usability with the Redesigned Reaction Pane

In the SimBiology desktop, the redesigned reaction pane consolidates the procedure to configure kinetic law, rate parameters, rate species, and reaction rates on one screen.

Additional Support for Showing Usages and Generating Reports in the SimBiology Desktop

Additional Support for Showing Usages

The **Compartments** and **Species** panes now have added support for showing usages of compartments and species in a model. To show usages of a component, right-click (Windows), or **Ctrl** +click (Macintosh) the compartment or species table, and select **Show Usages**. Support has also been added for showing usages from the **Diagram View**.

Additional Support for Generating Reports

You can now automatically populate report contents with one click in the **Report** pane. In the **Project Explorer**, right-click (Windows), or **Ctrl**+click (Macintosh) **Project Tasks** and select **Add Task > Generate report** to create a report. Select **Help > SimBiology Desktop Help** to see the context-

sensitive help for information on how to generate reports. Click **AutoBuild** to populate the contents of your report automatically.
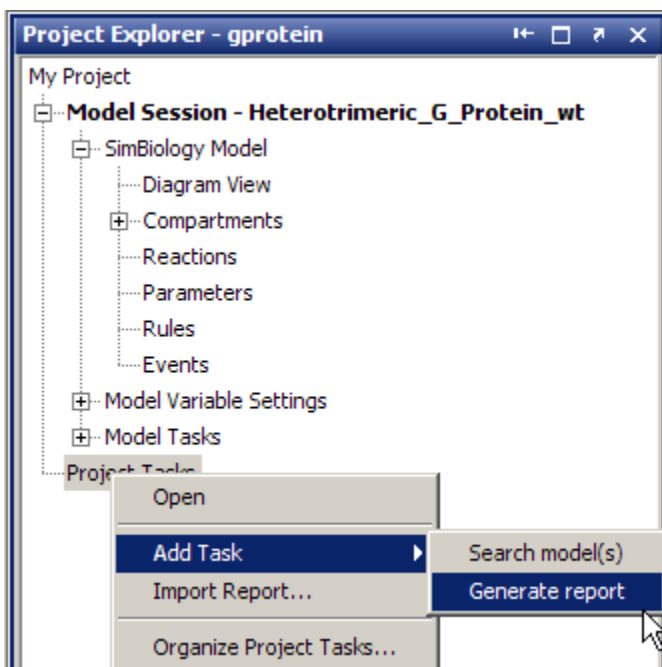
Support for Specifying Additional Inputs in Custom Plot Types

You can now specify additional inputs for plot types and define their types, default values, and ranges where applicable. For example, you can use this feature to extend the `Time` plot with an additional input to specify markers.

Compatibility Considerations

This consideration applies when you previously created custom plot types. Before, you could define the number of species or parameters that could be specified for `x` or `y` arguments. This functionality has been removed. You should modify your plot code to enforce this constraint. For an example of enforcing the constraint, see the code for the built-in plot type XY.

To access this code:

1   In the SimBiology desktop, select **Desktop > Library Explorer**. The **Library Explorer** opens.

2   Select **Plot Types**.

3   In the plot types table, select XY. The code section updates to show XY plot type code.

Edit Graphical Models Using the New Block Property Editor

The **Block Property Editor** is a tool that facilitates model building using the **Diagram View** pane by docking next to the diagram and allowing you to change properties of the selected block in the diagram. If you select multiple blocks, you can edit the properties that are common between each block. To open the **Block Property Editor**, select a block in the **Diagram View** pane, and then select **Diagram > Tools > Block Property Editor**.
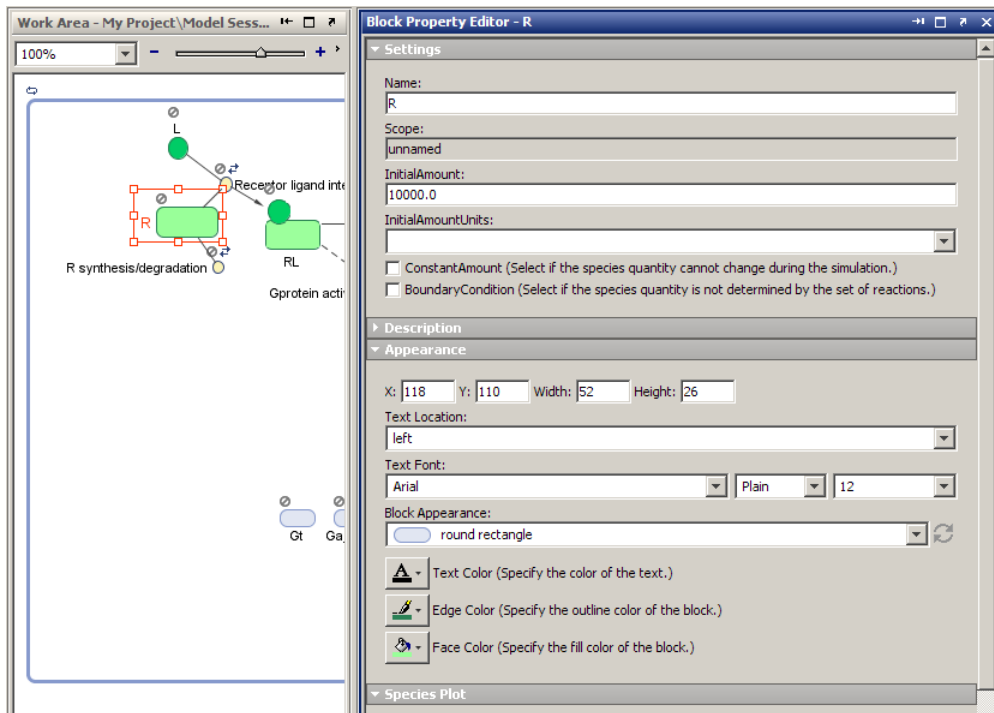
Manage and Share Libraries Using the New Library Explorer

The **Library Explorer** adds the ability to add, modify, and share the contents of kinetic law, plot type, unit, unit prefix, and block libraries. To open the **Library Explorer**, in the SimBiology desktop, select **Desktop > Show Library Explorer**.
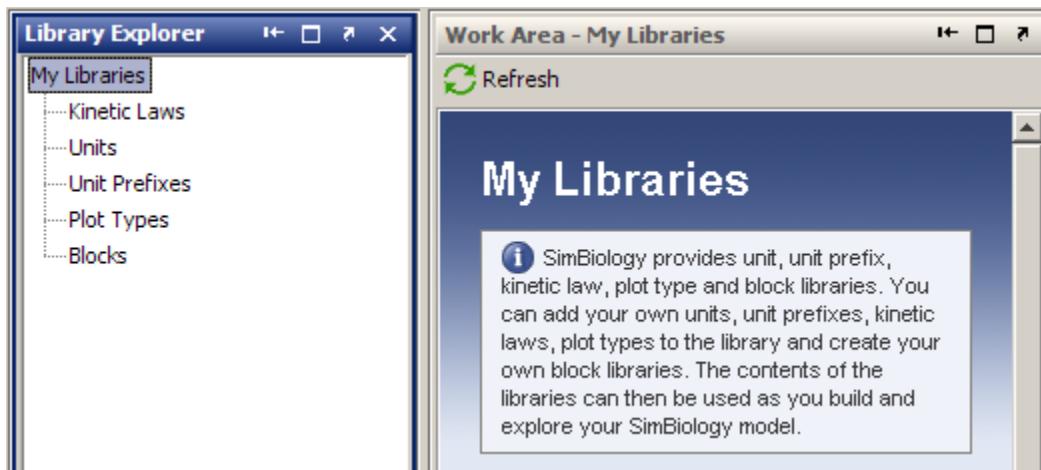
The **Library Explorer** shows all built-in and user-defined components, namely kinetic laws, units, unit prefixes, plot types, and blocks.

- **Kinetic Laws** — Contains kinetic laws that you can use as templates while creating a reaction rate expression.
- **Units** — Contains units that you can specify for compartment capacity, species amounts and parameter values, to do dimensional analysis and unit conversion during simulation.

- **Unit Prefixes** — Contains all unit prefixes that you can specify in combination with a valid unit for compartment capacity, species amounts and parameter values, to do dimensional analysis and unit conversion during simulation.
- **Plot Types** — Contains different types of plots that you can use with **Model Tasks** to visualize your results.
- **Blocks** — Contains blocks that you can use in the **Diagram View**.

For help, open the **Library Explorer** and select **Help > SimBiology Desktop Help** to see the context-sensitive help. Select a library in the **Library Explorer** to view information specific to the library.

Additional Options for Renaming Compartments, Species, and Parameters

New Method for Renaming at the Command Line

The newly added `rename` method allows you to change the name of a compartment, species, or parameter, and update the name in expressions that refer to the component. Use the `rename` method instead of the `set` method to enable the name change and expression update.

New Options for Renaming in the SimBiology Desktop

You can specify how compartments, species, and parameter names should be updated in expressions. When you rename a compartment, species, or parameter, by default the names are set to be updated in all expressions that refer to the component being renamed. During renaming, if the component is used in one or more expressions, you will see a dialog box that tells you which expressions will be updated. For example, the following dialog box appears when you try to rename species G to Gnew.
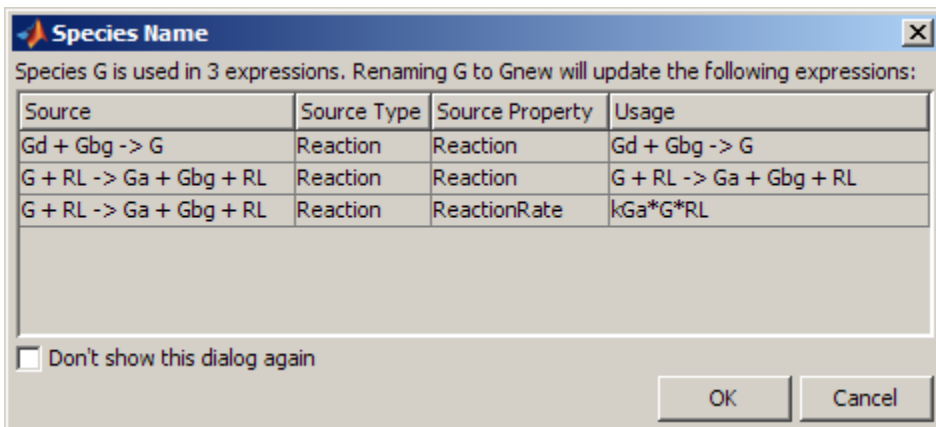
You can choose not to see this dialog box for every instance of renaming. You can also set your preferences on whether expressions should be updated. To select preferences for renaming:

1 Select **File > Preferences**. The Preferences dialog box opens.

2 Click **Rename** to view and select default renaming options.

You can also specify whether to update expressions while renaming a component in the **Compartment**, **Species**, or **Parameter** panes. Right-click (Windows or Linux®) or **Ctrl**+click (Macintosh) the species, parameters, or compartment table and select **Rename**. This allows you to select whether to use the new name in all expressions referring to the component.

Compatibility Considerations

Previously, if you changed the name in a table or the diagram, expressions that used the components were not updated unless you selected **Rename in Expressions** from the context menu. Now the default is to change the name in expressions. But, you will see a dialog box that allows you to cancel the action if necessary, and you can specify the default as shown in "Additional Options for Renaming Compartments, Species, and Parameters" on page 26-5.

Change in the Random Number Generator Used During Stochastic Simulations

The default random number generator used during stochastic simulation has changed. The stochastic solver now uses the random numbers from the MATLAB default stream. When you run a model using a stochastic solver, and have set the `RandomState` property in the configuration set, you may see different simulation results relative to previous releases because the random numbers used may be different.

For more information about the change to the random number generator, see Upgrade to Random Number Generator in the MATLAB release notes.

Compatibility Considerations

If you never set the `RandomState` property, there should be no compatibility considerations. If however, you have previously set `RandomState` for your model and want to reproduce your previous results, type `rand('state',0);` at the command line before running the simulation. This sets the random number generator to the one used in previous releases.

Functions and Properties Being Removed

Function or Property Name	What Happens When You Use Function or Property?	Use This Instead	Compatibility Considerations
sbioevent	Errors	addevent	Events must belong to a model. Replace all existing instances of `sbioevent` with `addevent`.
sbioparameter	Errors	addparameter	Parameters must belong to a model or a kinetic law. Replace all existing instances of `sbioparameter` with `addparameter`.
sbioreaction	Errors	addreaction	Reactions must belong to a model. Replace all existing instances of `sbioreaction` with `addreaction`.

Function or Property Name	What Happens When You Use Function or Property?	Use This Instead	Compatibility Considerations
Model name as an input argument for `sbioroot`	Errors	`sbioselect`	`sbioroot` does not accept a model name as an input argument. Use `sbioselect` to query models by name.
`sbiorule`	Errors	`addrule`	Rules must belong to a model. Replace all existing instances of `sbiorule` with `addrule`.
`sbiospecies`	Errors	`addspecies`	Species must belong to a compartment. Replace all existing instances of `sbiospecies` with `addspecies`.
`sbioregisterunit`	Errors	`Unit object` and `sbioaddtolibrary`	See "New Way to Add Units and Unit Prefixes" on page 28-7.
`sbiounregisterunit`	Errors	`sbioremovefromlibrary`	See "New Way to Add Units and Unit Prefixes" on page 28-7.
`sbioregisterunitprefix`	Errors	`UnitPrefix object` and `sbioaddtolibrary`	See "New Way to Add Units and Unit Prefixes" on page 28-7.
`sbiounregisterunitprefix`	Errors	`sbioremovefromlibrary`	See "New Way to Add Units and Unit Prefixes" on page 28-7.
`BuiltInKineticLaws`	Errors	`BuiltInLibrary`	See "Changes to the Library Structure in the Root" on page 28-5.
`BuiltInUnitPrefixes`	Errors	`BuiltInLibrary`	See "Changes to the Library Structure in the Root" on page 28-5.
`BuiltInUnits`	Errors	`BuiltInLibrary`	See "Changes to the Library Structure in the Root" on page 28-5.
`UserDefinedKineticLaws`	Errors	`UserDefinedLibrary`	See "Changes to the Library Structure in the Root" on page 28-5.
`UserDefinedUnits`	Errors	`UserDefinedLibrary`	See "Changes to the Library Structure in the Root" on page 28-5.
`UserDefinedUnitPrefixes`	Errors	`UserDefinedLibrary`	See "Changes to the Library Structure in the Root" on page 28-5.

R2008a

Version: 2.3

New Features

Bug Fixes

Compatibility Considerations

Support for 64-Bit Microsoft Windows

SimBiology software now has added support for 64-bit Windows (Win64).

Functions and Properties Being Removed

Function or Property Name	What Happens When You Use Function or Property?	Use This Instead	Compatibility Considerations
sbioevent	Warns	addevent	Events must belong to a model. Replace all existing instances of sbioevent with addevent.
sbioparameter	Warns	addparameter	Parameters must belong to a model or a kinetic law. Replace all existing instances of sbioparameter with addparameter.
sbioreaction	Warns	addreaction	Reactions must belong to a model. Replace all existing instances of sbioreaction with addreaction.
Model name as an input argument for sbioroot	Errors	sbioselect	sbioroot does not accept a model name as an input argument. Use sbioselect to query models by name.
sbiorule	Warns	addrule	Rules must belong to a model. Replace all existing instances of sbiorule with addrule.
sbiospecies	Warns	addspecies	Species must belong to a compartment. Replace all existing instances of sbiospecies with addspecies.
sbioregisterunit	Errors	Unit object and sbioaddtolibrary	See "New Way to Add Units and Unit Prefixes" on page 28-7.
sbiounregisterunit	Errors	sbioremovefromlibrary	See "New Way to Add Units and Unit Prefixes" on page 28-7.
sbioregisterunitprefix	Errors	UnitPrefix object and sbioaddtolibrary	See "New Way to Add Units and Unit Prefixes" on page 28-7.
sbiounregisterunitprefix	Errors	sbioremovefromlibrary	See "New Way to Add Units and Unit Prefixes" on page 28-7.
BuiltInKineticLaws	Errors	BuiltInLibrary	See "Changes to the Library Structure in the Root" on page 28-5.

Function or Property Name	What Happens When You Use Function or Property?	Use This Instead	Compatibility Considerations
`BuiltInUnitPrefixes`	Errors	`BuiltInLibrary`	See "Changes to the Library Structure in the Root" on page 28-5.
`BuiltInUnits`	Errors	`BuiltInLibrary`	See "Changes to the Library Structure in the Root" on page 28-5.
`UserDefinedKineticLaws`	Errors	`UserDefinedLibrary`	See "Changes to the Library Structure in the Root" on page 28-5.
`UserDefinedUnits`	Errors	`UserDefinedLibrary`	See "Changes to the Library Structure in the Root" on page 28-5.
`UserDefinedUnitPrefixes`	Errors	`UserDefinedLibrary`	See "Changes to the Library Structure in the Root" on page 28-5.

R2007b+

Version: 2.2

New Features

Bug Fixes

Compatibility Considerations

Changes to the Model Structure

The following new features and changes apply to a model's structure:

- "Compartments Now Supported" on page 28-2
- "Submodel Support Will Be Removed" on page 28-2

Compartments Now Supported

SimBiology models now let you add compartments to a model, specify compartment size, simulate, and do analysis with multiple compartments. Any model containing species must have a compartment, and the species must belong to a compartment. You can also perform dimensional analysis and unit conversion accounting for the specified compartment size. For more information on compartments, refer to the following sources in the documentation:

- `Compartment object` — Reference for compartment objects.

Submodel Support Will Be Removed

Support for submodels will be removed in a future release. You can still open models containing submodels and create submodels, but you cannot simulate or perform any analysis tasks on the model. Therefore, you should transition to using compartments where applicable.

Compatibility Considerations

If you have submodels in your previously created model, you can still open your model, but you must convert the submodels into top-level SimBiology models to be able to perform any tasks.

To convert submodels, use `sbioupdate`.

If you open a model that contains submodels in the SimBiology desktop, the submodels are automatically converted to models and placed in separate model sessions. See `sbioupdate` for more information.

As a result of removing submodels, the `addmodel` method and `Models` property of the model object are no longer relevant. You can still access the `addmodel` method and the `Models` property for this version of the software, though it may be removed in a future release.

Events

You can now add events to SimBiology models. Use events to describe sudden changes in a model system. Events are supported only by the `sundials` solver and the stochastic solver `ssa`.

For more information on events, refer to the following:

- Event Object — Discussion about events including how events are evaluated in the SimBiology User's Guide.
- `Event object` — Reference for event objects.
- SUNDIALS Solvers — Deterministic solver for simulating models with events in the SimBiology User's Guide.

Models containing events do not support sensitivity analysis.

Variants

Variants let you store the names and values of model components and use the values stored in a variant as the alternate value to apply during a simulation. You can store values for species `InitialAmount`, parameter `Value`, and compartment `Capacity` in a variant.

For more information on variants, refer to the following:

- `Variant object` — Reference for variant objects in the SimBiology Reference.

In the SimBiology desktop, expand **Model Variable Settings** and double-click **Variants** to open the **Variants** pane. The **SimBiology Desktop Help** updates with more information on adding and setting variants. If the help is not open in the desktop, select **Help > SimBiology Desktop Help**.
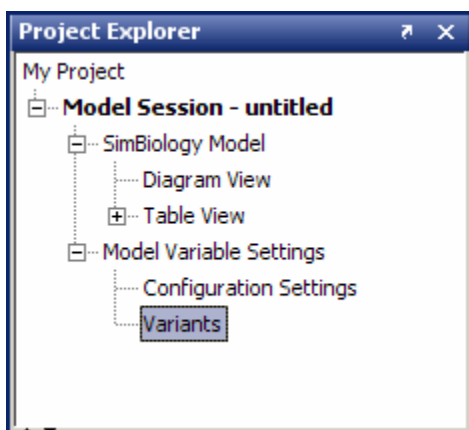
Support for Analysis Tasks in the Desktop

The SimBiology desktop now supports adding and managing analysis tasks through the following features:

- "Task Manager" on page 28-3
- "Sensitivity Analysis in the Desktop" on page 28-4
- "Scanning and Scanning with Sensitivities in the Desktop" on page 28-4
- "Ensemble Simulation Runs in the Desktop" on page 28-5
- "Conserved Cycle Calculations in the Desktop" on page 28-5
- "Create Custom Analysis Tasks" on page 28-5
- "Generate Reports for Projects" on page 28-5

Task Manager

The **Task Manager** lets you add and manage simulation and analysis tasks. In the **Project Explorer**, double-click **Model Session**. The **Model Session** pane opens with the **Task Manager** listed on the right.
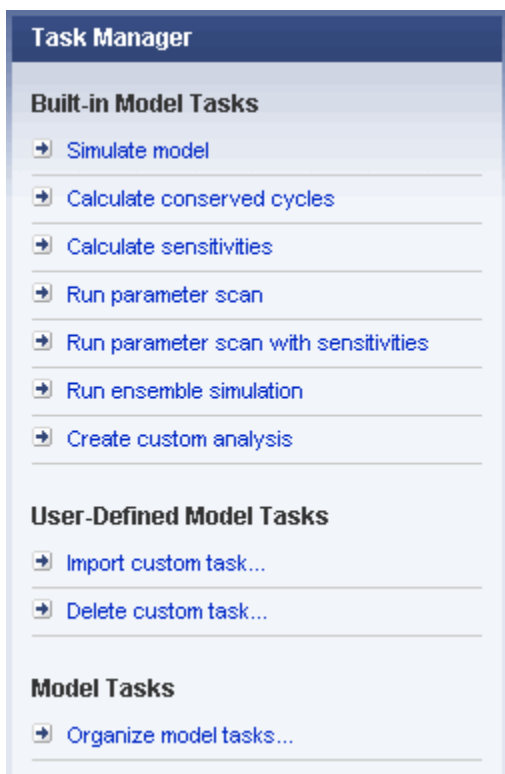
Click a task to add it to your model. The desktop adds the task to the **Project Explorer** and opens the task pane. For more information on setting up and running a task in the desktop, open the task pane and select **Help > SimBiology Desktop Help** to see the context-sensitive help.

Sensitivity Analysis in the Desktop

Sensitivity analysis is now supported in the desktop. Sensitivity analysis was previously available only through command line. Sensitivity analysis lets you calculate the time-dependent sensitivities of a species specified in **SpeciesOutputs** with respect to species initial conditions and parameter values.

See Performing Sensitivity Analysis Using the Desktop in the SimBiology User's Guide for more information. You must have a model in the desktop for this feature to be enabled.

For more information on sensitivity analysis, see Calculating Sensitivities in the SimBiology User's Guide.

Scanning and Scanning with Sensitivities in the Desktop

You can perform species and parameter scanning analysis alone or in combination with sensitivity analysis in the desktop. Scan a parameter value or a species initial amount to determine the effect of a range of values of the parameter or species.

Combine the scan with sensitivity analysis to explore the sensitivity of a species with respect to a range of values of a parameter or a species.

For information on how to add the task to a model, see "Task Manager" on page 28-3. For more information on setting up and running the task in the desktop, open the task pane and select **Help > SimBiology Desktop Help** to see the context-sensitive help. You need a model in the desktop for this feature to be enabled.

Ensemble Simulation Runs in the Desktop

The SimBiology desktop now supports ensemble simulations. You can perform ensemble simulations using the stochastic solvers to gather data from multiple stochastic runs of the model.

See Running Ensemble Simulations in the Desktop in the SimBiology User's Guide for more information. You need a model in the desktop for this feature to be enabled.

Conserved Cycle Calculations in the Desktop

The SimBiology desktop now supports conserved cycle calculations. This feature lets you calculate a complete set of linear conservation relations for the species in a SimBiology model object. For an introduction, see Determining Conserved Moieties in the SimBiology User's Guide.

For information on how to add the task to a model, see "Task Manager" on page 28-3. For more information on setting up and running the task in the desktop, open the task pane and select **Help > SimBiology Desktop Help** to see the context-sensitive help. You need a model in the desktop for this feature to be enabled.

Create Custom Analysis Tasks

The SimBiology desktop lets you create custom tasks that are associated with a project. You can either write new code or copy and modify task code from built-in tasks in the desktop. For more information about creating custom tasks, see Performing Custom Analysis in the Desktop in the SimBiology User's Guide.

For information on how to add the task to a model, see "Task Manager" on page 28-3. For more information on setting up and running the task in the desktop, open the task pane and select **Help > SimBiology Desktop Help** to see the context-sensitive help. You need a model in the desktop for this feature to be enabled.

Generate Reports for Projects

You can now generate reports for your projects. Report templates let you generate a report with specified information about a model. To generate a template, in the **Project Explorer**, click **Report Templates** to open the **Report Templates** pane. Select **Help > SimBiology Desktop Help** to see the context-sensitive help for information on how to generate reports.

Changes to the Library Structure in the Root

Built-in and user-defined libraries for units, unit prefixes, and abstract kinetic laws are now organized under two root object properties, `BuiltInLibrary` and `UserDefinedLibrary`, with subcategories for units, unit prefixes, and abstract kinetic laws. See `Root object`, `BuiltInLibrary`, and `UserDefinedLibrary` in SimBiology Reference for more information.

Compatibility Considerations

In previous versions, the libraries were organized under six properties:

- `UserDefinedKineticLaws`
- `BuiltInKineticLaws`
- `UserDefinedUnits`

- `BuiltInUnits`
- `UserDefinedUnitPrefixes`
- `BuiltInUnitPrefixes`

The changes to the library structure improve the organization of root object properties.

To illustrate the change using an example, previously you would access a user-defined kinetic law using the following syntax:

```
rootObj = sbioroot;
get(rootObj, 'UserDefinedKineticLaws')
```

You must now use the following syntax:

```
rootObj = sbioroot;
get(rootObj.UserDefinedLibrary, 'KineticLaws')
```

New Features for Solvers and Simulation Settings

The following new features and changes apply to solvers and simulation settings:

- "Support for Sundials Solvers" on page 28-6
- "New Property in Configuration Sets to Specify Species Dimensions" on page 28-6
- "SimData Object Holds All Simulation Data" on page 28-6

Support for Sundials Solvers

The Sundials package of solvers has been added in this release. The Sundials solvers are part of a freely available third-party package developed at Lawrence Livermore National Laboratory. Models that contain events are supported by the Sundials solvers and by the stochastic solver `ssa`. For more information, see Sundials Solvers in the SimBiology User's Guide.

New Property in Configuration Sets to Specify Species Dimensions

The new property `DefaultSpeciesDimension` lets you specify whether the default species dimensions should be `concentration` (default) or `substance`. This property thus lets you specify whether the solver should account for compartment capacity. If however, you specify the species units in the `InitialAmountUnits` property, these units define the species dimension regardless of the value in `DefaultSpeciesDimension`.

SimData Object Holds All Simulation Data

The SimBiology `SimData` object now stores the data returned from any simulation. For example, the output from the `sbiosimulate` function is now stored in a `SimData` object which holds time and state data as well as metadata, such as the types and names for the logged states or the configuration set used during simulation.

You can also store data from multiple simulation runs as an array of SimData objects. Thus, the output of `sbioensemblerun` is an array of SimData objects. See `SimData object` for more information and a list of methods and properties.

Compatibility Considerations

The SimData object is now the preferred container for simulation and analysis task data. Previously, simulation and analysis data were stored as time series objects. Functions that used to return time series objects now return SimData objects. If you have time series objects in your projects, you can convert them using `sbioupdate`. Functions that used to take a time series object as an input argument now take SimData object. You can use time series objects in an input argument, but you see a warning. Support for time series objects in SimBiology functions may be removed in a future version.

The `sbiogetsensmatrix` and `sbiogetnamedstate` functions are being replaced by the SimData object methods `getsensmatrix` and `selectbyname` respectively.

New Plot Functions

There are two new plot functions — `sbioplot` and `sbiosubplot`. Both functions let you plot data directly from the SimData object (see "SimData Object Holds All Simulation Data" on page 28-6).

`sbioplot` plots each simulation run for a SimData object or array of SimData objects from a model in the same figure. The plot is a time plot of each state in the object. The figure also shows a hierarchical display of all the runs in a tree, where you can choose which trajectories to show.

`sbiosubplot` plots each simulation run for a SimData object or array of objects into its own subplot. The subplot is a time plot of each state in the object. You can navigate through the plots in the figure window and select a subset of the plots to view.

New Sensitivity Analysis Property for Species Outputs

To set up sensitivity analysis, you must now specify an additional property called `SpeciesOutputs`. In `SpeciesOutputs`, specify the species for which you want to compute sensitivities.

Compatibility Considerations

Previously, sensitivity analysis used the species specified in `StatesToLog` as the species for which sensitivities should be calculated. `SpeciesOutputs` improves the functionality by separating the use of the properties.

If you have models from a previous version configured for sensitivity analysis, you must specify species for which you want to compute sensitivities in the `SpeciesOutputs` property. Until this property is specified, sensitivity analysis gives a warning that the `SpeciesOutputs` property is not set, and continues to use the species specified in `StatesToLog`. The use of `StatesToLog` in this context may not be available in a future version, so you should set the `SpeciesOutputs` property for your models, if applicable.

New Way to Add Units and Unit Prefixes

Units and unit prefixes are now represented by objects. You can create units and prefixes and add them to the user-defined library using the function `sbioaddtolibrary`. See `Unit object` and `UnitPrefix object` for more information.

Compatibility Considerations

Previously, sbioregisterunit and sbiounregisterunit created and removed units respectively. These functions now produce warnings and will be removed in a future version. Use unit objects and sbioremovefromlibrary instead.

Similarly, sbioregisterunitprefix and sbiounregisterunitprefix created and removed unit prefixes respectively. These functions now produce warnings and will be removed in a future version. Use unit prefix objects and sbioremovefromlibrary instead.

Functions and Properties Being Removed

Function or Property Name	What Happens When You Use Function or Property?	Use This Instead	Compatibility Considerations
addmodel	Warns	addcompartment, where applicable	See "Submodel Support Will Be Removed" on page 28-2.
Models	Still runs	Compartments, where applicable	See "Submodel Support Will Be Removed" on page 28-2.
Model name as an input argument for sbioroot	Warns	sbioselect	sbioroot does not accept a model name as an input argument. Use sbioselect to query models by name.
sbioregisterunit	Warns	Unit object and sbioaddtolibrary	See "New Way to Add Units and Unit Prefixes" on page 28-7.
sbiounregisterunit	Warns	sbioremovefromlibrary	See "New Way to Add Units and Unit Prefixes" on page 28-7.
sbioregisterunitprefix	Warns	UnitPrefix object and sbioaddtolibrary	See "New Way to Add Units and Unit Prefixes" on page 28-7.
sbiounregisterunitprefix	Warns	sbioremovefromlibrary	See "New Way to Add Units and Unit Prefixes" on page 28-7.

Function or Property Name	What Happens When You Use Function or Property?	Use This Instead	Compatibility Considerations
sbiogetsensmatrix	Warns	getsensmatrix	See "SimData Object Holds All Simulation Data" on page 28-6.
sbiogetnamedstate	Warns	selectbyname	See "SimData Object Holds All Simulation Data" on page 28-6.
BuiltInKineticLaws	Warns	BuiltInLibrary	See "Changes to the Library Structure in the Root" on page 28-5.
BuiltInUnits	Warns	BuiltInLibrary	See "Changes to the Library Structure in the Root" on page 28-5.
BuiltInUnitPrefixes	Warns	BuiltInLibrary	See "Changes to the Library Structure in the Root" on page 28-5.
UserDefinedKineticLaws	Warns	UserDefinedLibrary	See "Changes to the Library Structure in the Root" on page 28-5.
UserDefinedUnits	Warns	UserDefinedLibrary	See "Changes to the Library Structure in the Root" on page 28-5.
UserDefinedUnitPrefixes	Warns	UserDefinedLibrary	See "Changes to the Library Structure in the Root" on page 28-5.

R2007b

Version: 2.1.2

No New Features or Changes

R2007a

Version: 2.1.1

No New Features or Changes

R2006b+

Version: 2.1

New Features

Bug Fixes

Compatibility Considerations

Printing and Exporting the Diagram

You can annotate and print your pathway's block diagram in SimBiology, or export the diagram using `.svg`, `.jpeg`, or `.pdf` file formats. For example, you can annotate the diagram with the name of the author, the date, notes, and name of the organization. You can choose to place this content as a header or footer on the diagram page.

Diagram Menu

The SimBiology desktop supports multiple diagram features and actions through the **Diagram** menu. Use the menu options to do the following:

- Copy, paste, and delete blocks.
- Select model or diagram components by category (for example, select all lines or all Species blocks).
- Filter selected portions of a model diagram to include only model or diagram components by category.
- Edit multiple blocks or lines.
- Add selected blocks to a Block Library.
- Annotate and print, or export your model diagram.
- Perform layout tasks, for example, hide and show blocks, move blocks to the front or back of a diagram, rotate a block, and pin or unpin selected blocks.
- Copy the style of a block and apply the style to a group of selected blocks.
- Reload a graphic used for a block in the Block Properties dialog box.
- Access diagram tools such as the Diagram Table View, Block Library Browser, Diagram Overview, and Block Overview.

Compatibility Considerations

There is a compatibility consideration regarding the support for editing multiple blocks or lines. In SimBiology Versions 2.0 and 2.0.1, if you selected multiple blocks and/or lines and applied an editing action such as **Hide**, **Pin**, or **Hide Name**, the action applied only to the block on which you selected the right-click (context) menu. Starting in Version 2.1, applicable editing actions are propagated to all selected blocks.

Block Overview Tool

The Block Overview tool provides a summary of key information about a particular block. Hover the mouse over a block to find information in the **Block Overview** pane. Each block includes information pertinent to that type of block, for example, a species block overview shows **Name**, **InitialAmount**, **InitialAmountUnits**, the number and list of reactions the species is involved in, and a description of any indicators shown in the diagram.

Miscellaneous Desktop Enhancements

The enhancements for the SimBiology desktop let you do the following:

- Set your preference to open a SimBiology pane with a single-click in the **Project Explorer**. Select **File > Preferences** to open the Preferences dialog box.
- Interrupt and stop model verification by clicking **Stop** when **Verify** is running.
- Sort the available plot arguments in the **Simulation** and **Data** panes. The X and Y argument lists are now tabulated.
- View a species and any of its cloned blocks from the search results for the species.
- Select and view multiple results for **Find** and **Bookmarks** in either the diagram or the table form.

R2006b

Version: 2.0.1

Bug Fixes

R2006a+

Version: 2.0

New Features

Bug Fixes

Compatibility Considerations

Diagram Interface

The Diagram is a graphical user interface you can use to enter model pathways using block representations for species, reactions, and submodels. Use the Plot block to visualize simulation data during a simulation. For a tutorial, see Modeling Using the SimBiology Diagram in the SimBiology Getting Started Guide. You can also access video demos from a list of `SimBiology demos`.

Find and Bookmarks in Projects

- **Find** — Type a string in the **Find** box to quickly find matching model components in the project and in abstract kinetic laws.
- **Bookmarks** — Use complex rules to identify objects from the project, and create a custom and persistent set of desktop and project objects.

Compatibility Considerations

In Versions 1.0 and 1.0.1, SimBiology projects saved searches. In Version 2.0, the project saves bookmarks. The software converts a previous version's saved search into a bookmark. After you save a project in Version 2.0, a bookmark is saved and the old search is no longer available. If a project contains a search, you see a warning that the project file will contain a bookmark rather than a search after saving.

All functionality available in Version 1.0 searches are present in Version 2.0 bookmarks except for the ability to mix and match `and` and `or` between the search rows. If you have a multiple-row search saved with a mixture of `and` and `or`, when you load this project into Version 2.0, it is converted to either `all` or `any` based on whether the software first encounters `and` or `or`.

Sensitivity Analysis

You can perform sensitivity analysis using the following properties:

- `SensitivityAnalysis` — Configuration set property that lets you calculate the time-dependent sensitivities of all the species states defined by the `StatesToLog` property with respect to species initial conditions and parameter values.
- `SensitivityAnalysisOptions` — An object that holds the sensitivity analysis options in the configuration set object. Properties of `SensitivityAnalysisOptions` are summarized below:
  - `SpeciesInputFactors` — Specify the species with respect to which you want to compute the sensitivities of the species states in your model.
  - `ParameterInputFactors` — Specify the parameters with respect to which you want to compute the sensitivities of the species states in your model.
  - `Normalization` — Specify the normalization for the calculated sensitivities.

For an introduction and an example, see Calculating Sensitivities in the SimBiology User's Guide.

Parameter Estimation

The `sbioparamestim` function lets you estimate any or all parameters in your model using the experimental data you provide. The software uses the optimization functions in the MATLAB, Optimization Toolbox, and Global Optimization Toolbox software to enable parameter estimation.

Optimization Toolbox and Global Optimization Toolbox software are not required for you to use `sbioparamestim`. If you do not have these products installed, `sbioparamestim` uses the MATLAB function `fminsearch` by default.

Ensemble Simulation Runs

You can perform ensemble simulations using the stochastic solvers to gather data from multiple stochastic runs of the model. The following functions let you perform ensemble runs:

- `sbioensemblerun` — Performs multiple stochastic ensemble runs of the SimBiology model object.
- `sbioensembleplot` — Shows a 2-D distribution plot or a 3-D shaded plot of the time varying distribution of one or more specified species in the ensemble data generated by `sbioensemblerun`.
- `sbioensemblestats` — Gets mean and variance as a function of time for all the species in the ensemble data generated by `sbioensemblerun`.

Moiety Conservation

The `sbioconsmoiety` function lets you calculate a complete set of linear conservation relations for the species in a SimBiology model object.

For an introduction and an example, see Moiety Conservation in the SimBiology User's Guide.

Model Verification and Validation

SimBiology software performs model verification and validation either during simulation, or when you explicitly execute the commands for verification before simulation.

Verification at the Command Line

The following new functions let you verify and validate, at the command line, that your model is ready for simulation:

- `verify` — Performs checks on a model to verify that you can simulate the model. You see stacked errors and warnings if any problems are found. To see the entire list of errors and warnings, use `sbiolasterror` and `sbiolastwarning`.
- `sbiolasterror` — Returns a SimBiology diagnostic structure array containing the last errors that are generated.
- `sbiolastwarning` — Returns a SimBiology diagnostic structure array containing the last warnings that are generated.

Verification on the SimBiology Desktop

Click the **Verify** button on the SimBiology desktop toolbar to perform verification and validation of your model. The **Output** pane opens to show the errors and warnings. You can double-click a result row to go to the location of the error or warning.

Simulation and Solvers

The following new features and changes apply to simulation settings and solvers:

- `MaxStep` — Lets you specify the upper bound on solver step size for a deterministic solver. `MaxStep` is a property of the `SolverOptions` object.
- **Implicit Tau solver settings** — For `impltau`, `AbsoluteTolerance` holds the value for convergence tolerance for the nonlinear solver that is used internally by the Implicit Tau solver. You can now specify `AbsoluteTolerance` for `impltau`. Previously, if you selected the Implicit Tau solver, the software ignored any changes to the `AbsoluteTolerance` and `RelativeTolerance` options within a configuration set and used the default values set internally.
- `UnitConversion` — Supported by both deterministic solvers and stochastic solvers. Previously `UnitConversion` was supported only by the stochastic solvers.

Implicit Tau Solver Settings Compatibility Considerations

The `RelativeTolerance` property is no longer valid for the Implicit Tau (`impltau`) solver.

When you load a file created in a previous version, the project loads the `RelativeTolerance` property. But when you save the file, the software updates the change.

Unit Conversion Compatibility Considerations

The `UnitConversion` property default is now `'false'`. If you load a SimBiology project created in a previous version into the SimBiology desktop, the `UnitConversion` setting in each model in the project remains as the saved setting. If however, you are running a script, you must now remember to set the `UnitConversion` property to `true` if you want the software to perform unit conversions

New Demos for SimBiology Version 2.0

There are 14 new demos for SimBiology Version 2.0. Click `SimBiology demos` or type `demo('MATLAB', 'SimBiology')` at the command prompt.

R2006a

Version: 1.0.1

Rules

- The characters \, ^, and * in species and parameter names are now supported in rules.
- Rules are now supported in submodels.

R14SP3+

Version: 1.0

New Features

Introduction

SimBiology Version 1.0 (Release 14SP3+) is an integrated environment for modeling biological processes, simulating the dynamic behavior of these processes, and analyzing simulation and experimental data. Biological processes include metabolic, genetic, and signaling pathways with transform, binding, and transport reactions. You can use SimBiology software as a tool in three major areas:

- **Model** — Design and build models by entering model components with a graphical desktop interface, or use the MATLAB Command Window.
- **Simulate** — Select deterministic or stochastic solvers and observe the changes in species amounts and variable parameter values over time.
- **Analyze** — Log data from a simulation and export the data to the MATLAB workspace. Compare simulation with experimental data for parameter estimation and model validation.

Features

The features in SimBiology Version 1.0 are the following:

- **Graphical user interface** — Provides access to the command-line functionality through a graphical user interface (GUI).
- **Command-line interface** — All the features are accessible and executable from the MATLAB Command Window.
- **Data formats and projects** — Organize and save related models, simulation data, and analysis results in project files. Save user-defined kinetic laws and units. Share models by exporting SBML Level 2 files.
- **Modeling** — Create biological models by adding components that include reactions, species, parameters, kinetic laws, rules, and submodels.
- **Simulation** — Select either deterministic or stochastic solvers with dimensional analysis and unit conversion.
- **Analysis** — SimBiology software is fully integrated with MATLAB. Record data during a simulation and analyze results in MATLAB.

Known Software Problems

To view important open bugs for SimBiology Version 1.0, use the Bug Reports interface on the MathWorks Web site.

Note If you are not already logged in to your MathWorks Account, when you link to the Bug Reports interface (see below), you will be prompted to log in or create an account.

After you are logged in, use this `Bug Reports` link. You will see the bug report for SimBiology. The report is sorted with fixed bugs listed first, and then open bugs. You can select the Status column to list the open bugs first.

If you are viewing these release notes in PDF form on the MathWorks Web site, you can refer to the HTML form of the release notes on the MathWorks Web site and use the link provided.

Software problems include unsupported SBML features and current feature and function limitations.

Unsupported SBML Level 2 Version 1 Features

SimBiology software supports a subset of the SBML Level 2 Version 1 specification. Unsupported features include:

- Compartments — Model compartments are not supported. If an imported SBML model has a single compartment, the model is loaded as a top-level model. If the model has multiple compartments, you see a warning and the software does not load the SBML file.

- Volume — Volume is not supported and cannot be specified.

- Events — Events in an SBML file are ignored when you are importing into a project.

- Piecewise kinetics — Models with piecewise kinetics are loaded in, but the software ignores the definitions for piecewise kinetics.

- Function definitions — Models containing functional definitions are loaded, but you see a warning and the software ignores the function definitions.

- MATLAB incompatible variable names in `UnitDefinition` — Models that have variable names incompatible with MATLAB in `UnitDefinition` are not loaded and you see an error message.

Functional Limitations

Simulation and Solvers

- Stochastic solvers support only mass action kinetics, while ODE solvers support all built-in and user-defined kinetic laws.

- If you use a stochastic solver to simulate a model, the software ignores any rate, assignment, or algebraic rules if present in the model.

- If you have `DimensionalAnalysis` and `UnitConversion` on for the stochastic solvers, note that the software assumes that volume is unity during simulation. The stochastic solvers perform calculations using species units in molecules. Therefore, if you specify the species units in molecules per unit volume or moles per unit volume, the software assumes volume to be unity and converts species amounts to molecules for simulation. The results are finally plotted in the units you specified for the species.

  In addition, if you have reactions with stoichiometric coefficients greater than or equal to 2, you need to convert the deterministic rate constants to stochastic rate constants. For example, 2 R -> P has a reaction rate of v=k[R]^2. If R is moles/liter, the deterministic rate constant k has units of `liter/mole-second`. If the unit of species concentration is `molecule`, then the stochastic rate constant c has units of `1/molecule-second`, and c = 2k/NV where N is Avogadro's Number, `6.022e23 molecules/mole`, and V is the volume of the model in liters.

- When you select the Implicit Tau solver, the software ignores any changes to `AbsoluteTolerance` and `RelativeTolerance` options within a configuration set and uses the default values that are set internally.

- By default, `StatesToLog` is set to `'all'` and all variable parameters are logged. Variable parameters are those that have `ConstantValue` cleared or `false`. If you choose the species to log, however, you cannot log the variable parameters.

- The characters \, ^, and * in species and parameter names are not supported in rules.

Units

- Stochastic solvers support dimensional analysis and unit conversion. ODE solvers support dimensional analysis but not unit conversion.

- You can delete a unit that is being used in a model; however, you will see an error when you try to simulate the model or export to SBML.

Submodels

- The context menus (right-click options) for the **Species**, **Reaction**, **Parameter** and **Rule** nodes that appear beneath a submodel node all act on the corresponding parent model node, and not on the submodel node. For example, if you select **Delete All Species** in the submodel **Species** node, this selection deletes the species in the parent model.
- Rules are not supported in submodels.

SBML Export Limitations

There are features in SimBiology software that are not supported in SBML. When you export a model to an SBML file, you might lose some of these features.

- Submodels are not supported by SBML export.
- The abstract kinetic law name and corresponding expression are not supported by SBML, but the associated reaction rate equation is exported to SBML.
- The properties `Tag`, `UserData`, and `Active` are not supported by SBML export.

Tips

Naming SimBiology Variables

- If you are using a species or parameter name that is not a valid MATLAB variable name, do the following:

  - Enclose the name in square brackets when writing a reaction rate equation or a rule.
  - Enter the name without brackets when you are creating the species or parameter, or when you add a reaction.

  For example, enclose `[DNA polymerase+]` and `[K_DNA polymerase+]` within brackets in reaction rates and rules, but, enter `DNA polymerase+` or `K_DNA polymerase+` when creating a species, adding a reaction, or creating a parameter.

- The names `i` and `j` are reserved MATLAB characters. Because expressions in abstract kinetic laws, reaction rates, and rules are considered to be MATLAB code, the software evaluates `i` or `j` as an imaginary number and not as the value of species `i` or `j`. For example, an expression `V*S*i/K` is interpreted to have three names, `V`, `S`, and `K`, instead of four. Use brackets to protect such variables.

If a variable in a reaction rate equation or rule has the same name as a MATLAB function, the software evaluates the expression as a call to the MATLAB function. In general, when creating variable names, you should avoid using MATLAB function names or variable names that are invalid in MATLAB.

Changing SimBiology Variable Names

- If you change the `Name` of a parameter you must configure all applicable elements, such as rules that use the parameter, any user-specified `ReactionRate`, or the kinetic law object property `ParameterVariableNames`. Use the method `setparameter` to configure `ParameterVariableNames`.

To update parameter names in the SimBiology graphical user interface, access each appropriate pane through the **Project Explorer**.

- If you change the `Name` of a species you must configure all applicable elements, such as rules that use the parameter, any user-specified `ReactionRate`, or the kinetic law object property `SpeciesVariableNames`. Use the method `setspecies` to configure `SpeciesVariableNames`.

  To update species names in the SimBiology graphical user interface, access each appropriate pane through the **Project Explorer**. The software automatically updates species names for reactions that use `MassAction` kinetic law.

Upgrading from a Beta Release

Any projects that you created and saved with the SimBiology beta release version will not load in Version 1.0.

As a workaround, before upgrading to Version 1.0, save your models to SBML, upgrade to Version 1.0, and then import the SBML models into Version 1.0 projects. Alternatively, contact MathWorks for help with your conversion.