

Release Notes for SimBiology[®]

How to Contact MathWorks



www.mathworks.com Web
comp.soft-sys.matlab Newsgroup
www.mathworks.com/contact_TS.html Technical Support



suggest@mathworks.com Product enhancement suggestions
bugs@mathworks.com Bug reports
doc@mathworks.com Documentation error reports
service@mathworks.com Order status, license renewals, passcodes
info@mathworks.com Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



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

For contact information about worldwide offices, see the MathWorks Web site.

Release Notes for SimBiology®

© COPYRIGHT 2005–2012 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

R2012b

| | |
|--|----|
| Speedup of simulation of models with multicore processors and computer clusters using Parallel Computing Toolbox | 2 |
| Text display of SimBiology models as a system of differential equations | 3 |
| Fit Data Tool and Simulation Viewer to interactively visualize and explore data and models | 4 |
| Noncompartmental analysis for characterizing pharmacokinetic data | 5 |
| Deployment of SimBiology models using MATLAB Compiler | 6 |
| SimBiology desktop enhancements | 7 |
| Import SBML from URL | 8 |
| Scaled absolute tolerances | 9 |
| Enhancements to sensitivity analysis | 10 |
| Log compartment data during simulations | 11 |
| Enhancements to simulation results reporting | 12 |
| Enhancement to writing repeated assignment rules | 13 |
| Removal of sbiohelp function | 14 |
| Functionality being removed | 15 |

R2012a

| | |
|---|----|
| SimBiology Desktop Enhancements | 18 |
| SBML Level 2 Version 4 Support | 19 |
| Annotation Property Being Removed | 20 |
| Celsius and Fahrenheit Units Removed | 21 |
| Enhanced Solver Support | 22 |
| Enhanced Simulation Data Logging Support | 24 |
| Changes to Simulation Stop Criteria | 25 |
| Changes to the AbsoluteTolerance Property | 26 |
| Parameter Estimation and Fitting Enhancements | 27 |
| Weighted Least-Squares Fitting | 28 |
| Simultaneously Fit Data from Multiple Dose Groups | 29 |
| Dosing Enhancements | 30 |

| | |
|---|----|
| Covariate Class, Properties, and Methods | 31 |
| Population Fitting Better Supports Covariate Analysis ... | 32 |
| Fitting Functions | 34 |
| Changes to Sensitivity Analysis | 35 |
| SimData Object Stores Simulation Data | 36 |
| Submodel Support Removed | 37 |
| Functionality Being Removed or Changed | 38 |

R2011b

| | |
|--|----|
| Redesigned and Enhanced Desktop | 44 |
| Open Projects in the Desktop from the Command Line ... | 47 |
| New Covariate Class, Properties, and Methods | 48 |
| Population Fitting Better Supports Covariate Analysis ... | 49 |
| Fitting Functions Enhancements | 51 |
| Parameter Estimation Enhancements | 52 |
| Enhancements to Sensitivity Analysis | 53 |
| Additional Syntax Checking of Reaction Rates, Rule Assignments, Event Triggers, and Event Functions ... | 54 |
| Annotation Property Being Removed | 55 |
| Function Elements Being Removed | 56 |

R2011a

| | |
|---|----|
| Redesigned and Enhanced Desktop | 62 |
| Support of Arbitrary Time Units | 64 |
| Parameter Estimation and Population Fitting Enhancements | 65 |
| Calculation of Weighted Residuals for Population Fitting Tasks | 67 |
| Increased Performance When Using SUNDIALS Solvers | 68 |
| Evaluation of Simultaneous Events | 69 |
| New and Updated Demos | 70 |

R2010b

| | |
|---|----|
| Support for Error Models Using sbionlmeft | 72 |
| Support for Covariate Analysis | 73 |
| Support for Multiple Response Fitting | 74 |
| Support for Time Lags | 75 |
| Support for Dimensionless Units | 76 |
| Modeling, Simulation, and Analysis Tools | 77 |
| Parameter Scan Subplots Display Parameter Information | 78 |
| Removal of the Export Tab | 79 |
| Demos for Modeling | 80 |

R2010a

| | |
|--|----|
| Stochastic Approximation Expectation-Maximization (SAEM) Algorithm for Fitting Population Data | 82 |
| Enhanced Support for Importing NONMEM Formatted Files | 83 |
| New Mode for Accelerating Simulations | 84 |
| Enhanced Support for Applying Dosing to a Model and Dosing Multiple Compartments | 85 |
| Support for Parameter Transformations | 88 |
| Support for Error Models | 89 |
| Functions and Properties Being Removed | 90 |

R2009b

| | |
|---|----|
| Increased Performance When Repeatedly Simulating a Model | 92 |
| Enhanced Desktop Support for Scanning Using Monte Carlo Methods | 93 |
| Desktop Support for Copy and Paste | 94 |
| View Status of Parameter Fitting Task During Run | 95 |
| Improved Usability for Model Building and Debugging | 96 |
| Unit Conversion Compatibility Considerations | 97 |
| Functions and Properties Being Removed | 98 |

R2009a

| | |
|---|-----|
| New Feature to Import, Visualize, and Statistically Analyze Clinical and Experimental Data | 100 |
| New Functionality to Create Pharmacokinetic Models ... | 101 |
| New Functionality to Fit Data and Estimate Parameters Using Nonlinear Mixed Effects | 102 |
| New Diagnostic Plots for Individual and Population Fitting Results | 103 |
| New Project Wizard to Add Data, Create Models, and Specify Tasks | 104 |
| New simbiology Command to Open the SimBiology Desktop | 105 |
| Enhanced Usability Features in the SimBiology Desktop | 106 |
| New Demo for Pharmacokinetic Modeling | 107 |

R2008b

| | |
|---|-----|
| Enhanced Usability with the Redesigned Reaction Pane .. | 110 |
| Additional Support for Showing Usages and Generating Reports in the SimBiology Desktop | 111 |
| Support for Specifying Additional Inputs in Custom Plot Types | 113 |
| Edit Graphical Models Using the New Block Property Editor | 114 |
| Manage and Share Libraries Using the New Library Explorer | 115 |
| Additional Options for Renaming Compartments, Species, and Parameters | 117 |
| Change in the Random Number Generator Used During Stochastic Simulations | 119 |
| Functions and Properties Being Removed | 120 |

R2008a

| | |
|--|-----|
| Support for 64-Bit Microsoft Windows | 124 |
| Removing Support for Standalone Model Components ... | 125 |
| Functions and Properties Being Removed | 126 |

R2007b+

| | |
|---|-----|
| Changes to the Model Structure | 130 |
| Events | 132 |
| Variants | 133 |
| Support for Analysis Tasks in the Desktop | 134 |
| Changes to the Library Structure in the Root | 138 |
| New Features for Solvers and Simulation Settings | 139 |
| New Plot Functions | 141 |
| New Sensitivity Analysis Property for Species Outputs ... | 142 |
| New Way to Add Units and Unit Prefixes | 143 |
| Functions and Properties Being Removed | 144 |

R2007b

No New Features or Changes

R2007a

No New Features or Changes

R2006b+

| | |
|--|-----|
| Printing and Exporting the Diagram | 152 |
| Diagram Menu | 153 |
| Block Overview Tool | 154 |
| Miscellaneous Desktop Enhancements | 155 |

R2006b

R2006a+

| | |
|--|-----|
| Diagram Interface | 160 |
| Find and Bookmarks in Projects | 161 |
| Sensitivity Analysis | 162 |
| Parameter Estimation | 163 |
| Ensemble Simulation Runs | 164 |
| Moiety Conservation | 165 |
| Model Verification and Validation | 166 |
| Simulation and Solvers | 167 |
| New Demos for SimBiology Version 2.0 | 168 |

R2006a

| | |
|-------------|-----|
| Rules | 170 |
|-------------|-----|

R14SP3+

| | |
|-------------------------------------|-----|
| Introduction | 172 |
| Features | 173 |
| Known Software Problems | 174 |
| Upgrading from a Beta Release | 179 |

R2012b

Version: 4.2
New Features: Yes
Bug Fixes: Yes


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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

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: Yes

Compatibility Considerations

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

Functionality being removed
Compatibility Considerations: Yes

Functionality Being Removed or Changed

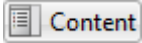

| Functionality | What Happens When You Use This Functionality? | Use This Instead | Compatibility Considerations |
|----------------------|--|-------------------------|---|
| sbiohelp function | Errors | — | See the Compatibility Considerations subheading in “Removal of sbiohelp function” on page 14. |

R2012a

Version: 4.1
New Features: Yes
Bug Fixes: Yes

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: Yes

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 Removed

Compatibility Considerations: Yes

Celsius, celsius, and fahrenheit units are removed from the built-in units library. 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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

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 `sbionlmeffit` and `sbionlmeffitsa` 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`
- `sbionlmeffit`
- `sbionlmeffitsa`

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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

The `sbionlmeffit` and `sbionlmeffitsa` functions accept the `CovariateModel` object.

The `sbionlmeffit` and `sbionlmeffitsa` 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 `sbionlmeffit` and `sbionlmeffitsa` reference pages.

Compatibility Considerations

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

```
results = sbionlmeffit(modelObj, pkModelMapObject, pkDataObject,  
pkCovModel)
```

```
results = sbionlmeffitsa(modelObj, pkModelMapObject, pkDataObject,  
pkCovModel)
```

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

```
results = sbionlmeffit(modelObj, pkModelMapObject, pkDataObject,  
CovModel)
```

```
results = sbionlmeffitsa(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 |
|-----------------------|---|
| <code>estimate</code> | <code>FixedEffects</code> |
| <code>phiP</code> | <code>PopulationParameterEstimates</code> |
| <code>phiI</code> | <code>IndividualParameterEstimates</code> |
| <code>beta</code> | <code>FixedEffects</code> |
| <code>psi</code> | <code>RandomEffectCovarianceMatrix</code> |
| <code>b</code> | <code>RandomEffects</code> |

Fitting Functions

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

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
- `sbiouupdate` function

Functionality Being Removed or Changed
Compatibility Considerations: Yes

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 31. |
| 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 sbionlmeft or sbionlmeftsa | Errors | FixedEffects field | See the Compatibility Considerations subheading in “Population Fitting Better Supports Covariate Analysis” on page 32. |

| 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 | PopulationParameterEstimates field | See the Compatibility Considerations subheading in “Population Fitting Better Supports Covariate Analysis” on page 32. |
| phiI field in the results structure returned by sbionlmeFit or sbionlmeFitsa | Errors | IndividualParameterEstimates field | See the Compatibility Considerations subheading in “Population Fitting Better Supports Covariate Analysis” on page 32. |
| psi field in the results structure returned by sbionlmeFit or sbionlmeFitsa | Errors | RandomEffectCovarianceMatrix field | See the Compatibility Considerations subheading in “Population Fitting Better Supports Covariate Analysis” on page 32. |


| Functionality | What Happens When You Use This Functionality? | Use This Instead | Compatibility Considerations |
|---|--|---|--|
| b field in the results structure returned by sbionlmeffit or sbionlmeffitsa | Errors | RandomEffects field | See the Compatibility Considerations subheading in “Population Fitting Better Supports Covariate Analysis” on page 32. |
| 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 34. |

R2011b

Version: 4.0
New Features: Yes
Bug Fixes: Yes

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 screenshot displays the SimBiology software interface. The top navigation bar shows the 'HOME' tab selected. Below it is a toolstrip with buttons for 'New', 'Save', 'Open', 'Close', 'View Contents', 'Add Model', 'Add Task', and 'Add Data'. The address bar shows the path 'Home > Project > Models > lotka'. The main workspace is divided into two sections: 'Add Expression' and 'Add Quantity'. The 'Add Expression' section contains a table with three rows of reaction data. The 'Add Quantity' section contains a table with eight rows of species and parameter data.

| Type | Name | Active | Expression |
|----------|-----------|-------------------------------------|--|
| reaction | Reaction1 | <input checked="" type="checkbox"/> | $x + y1 \rightarrow 2 y1 + x, c1 y1^2$ |
| reaction | Reaction2 | <input checked="" type="checkbox"/> | $y1 + y2 \rightarrow 2 y2, c2 y2^2$ |
| reaction | Reaction3 | <input checked="" type="checkbox"/> | $y2 \rightarrow z, c3 y2$ |

| Type | Scope | Name | Value | Units | Constant |
|-------------|-------------------------------|---------|-------|-------|-------------------------------------|
| compartment | lotka | unnamed | 1.0 | | <input checked="" type="checkbox"/> |
| species | unnamed | z | 1.0 | | <input type="checkbox"/> |
| species | unnamed | y1 | 500.0 | | <input type="checkbox"/> |
| species | unnamed | y2 | 400.0 | | <input type="checkbox"/> |
| species | unnamed | z | 0.0 | | <input type="checkbox"/> |
| parameter | $x + y1 \rightarrow 2 y1 + x$ | c1 | 10.0 | | <input checked="" type="checkbox"/> |
| parameter | $y1 + y2 \rightarrow 2 y2$ | c2 | 0.01 | | <input checked="" type="checkbox"/> |
| parameter | $y2 \rightarrow z$ | c3 | 10.0 | | <input checked="" type="checkbox"/> |

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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

The `sbionlmeffit` and `sbionlmeffitsa` functions accept the new `CovariateModel` object.

The `sbionlmeffit` and `sbionlmeffitsa` 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 `sbionlmeffit` and `sbionlmeffitsa` reference pages.

Compatibility Considerations

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

```
results = sbionlmeffit(modelObj, pkModelMapObject, pkDataObject,  
pkCovModel)
```

```
results = sbionlmeffitsa(modelObj, pkModelMapObject, pkDataObject,  
pkCovModel)
```

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

```
results = sbionlmeffit(modelObj, pkModelMapObject, pkDataObject,  
CovModel)
```

```
results = sbionlmeffitsa(modelObj, pkModelMapObject, pkDataObject,  
CovModel)
```

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

| Removed Field | New Field |
|-----------------------|---|
| <code>estimate</code> | <code>FixedEffects</code> |
| <code>phiP</code> | <code>PopulationParameterEstimates</code> |
| <code>phiI</code> | <code>IndividualParameterEstimates</code> |
| <code>beta</code> | <code>FixedEffects</code> |
| <code>psi</code> | <code>RandomEffectCovarianceMatrix</code> |
| <code>b</code> | <code>RandomEffects</code> |

Support for the following name-value pair arguments used by `sbionlmeffit` and `sbionlmeffitsa` 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.
- `REParamSelect` — 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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

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: Yes

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
Compatibility Considerations: Yes

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 48. |
| 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 49. |
| 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 49. |

| Function Element Name | What Happens When You Use This Function Element | Use This Instead | Compatibility Considerations |
|--|--|--|--|
| phiI field in the results structure returned by sbionlmeffit or sbionlmeffitsa | Warns | IndividualParameterEstimates field | See the Compatibility Considerations subheading in “Population Fitting Better Supports Covariate Analysis” on page 49. |
| psi field in the results structure returned by sbionlmeffit or sbionlmeffitsa | Warns | RandomEffectCovarianceMatrix field | See the Compatibility Considerations subheading in “Population Fitting Better Supports Covariate Analysis” on page 49. |
| b field in the results structure returned by sbionlmeffit or sbionlmeffitsa | Warns | RandomEffects field | See the Compatibility Considerations subheading in “Population Fitting Better Supports Covariate Analysis” on page 49. |
| estimate field in the results structure returned by sbionlinffit | Warns | ParameterEstimates property of the results object returned by sbionlinffit | See the Compatibility Considerations subheading in “Fitting Functions Enhancements” on page 51. |

| Function Element Name | What Happens When You Use This Function Element | Use This Instead | Compatibility Considerations |
|---|--|---|---|
| Annotation property of SimBiology objects | Warns | — | See the Compatibility Considerations subheading in “Annotation Property Being Removed” on page 55. |
| 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 53. |
| 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 53. |

R2011a

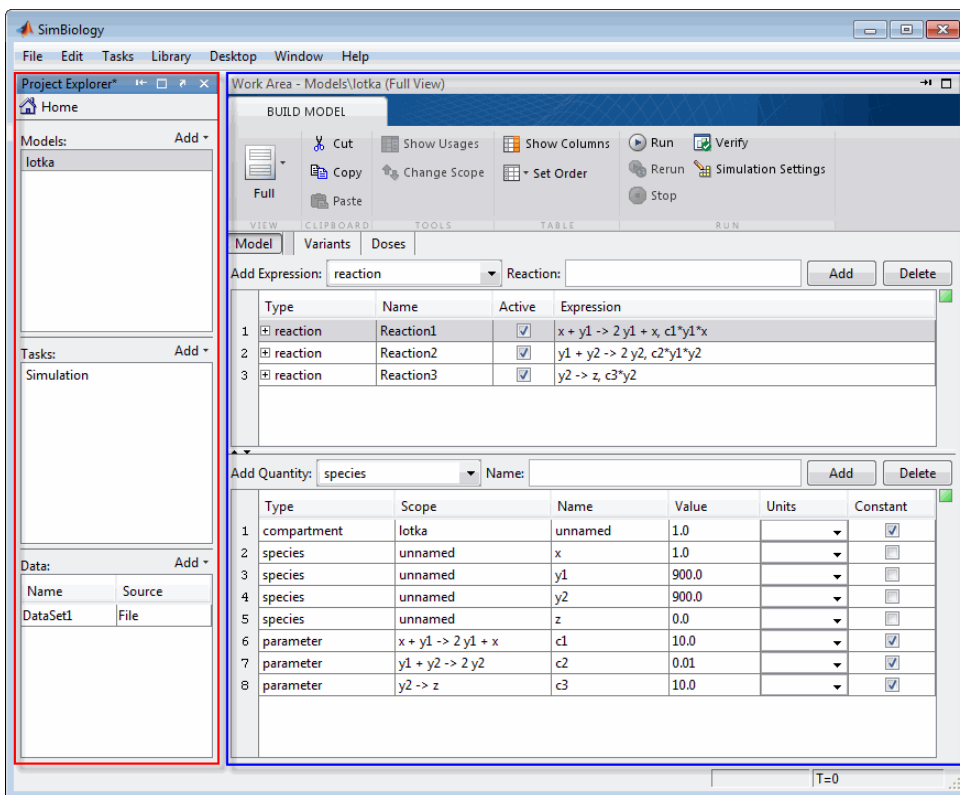
Version: 3.4
New Features: Yes
Bug Fixes: Yes

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

Compatibility Considerations: Yes

Following is a new function:

`sbiopfittool` — 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 $\text{eps}^{(1/3)}$.
- 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 $\text{eps}^{(1/3)}$.

- 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`
- `sbionlmeffit`
- `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

Compatibility Considerations: Yes

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.

R2010b

Version: 3.3
New Features: Yes
Bug Fixes: Yes

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.

For more information, see [Specifying an Error Model](#) in the SimBiology documentation.

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).

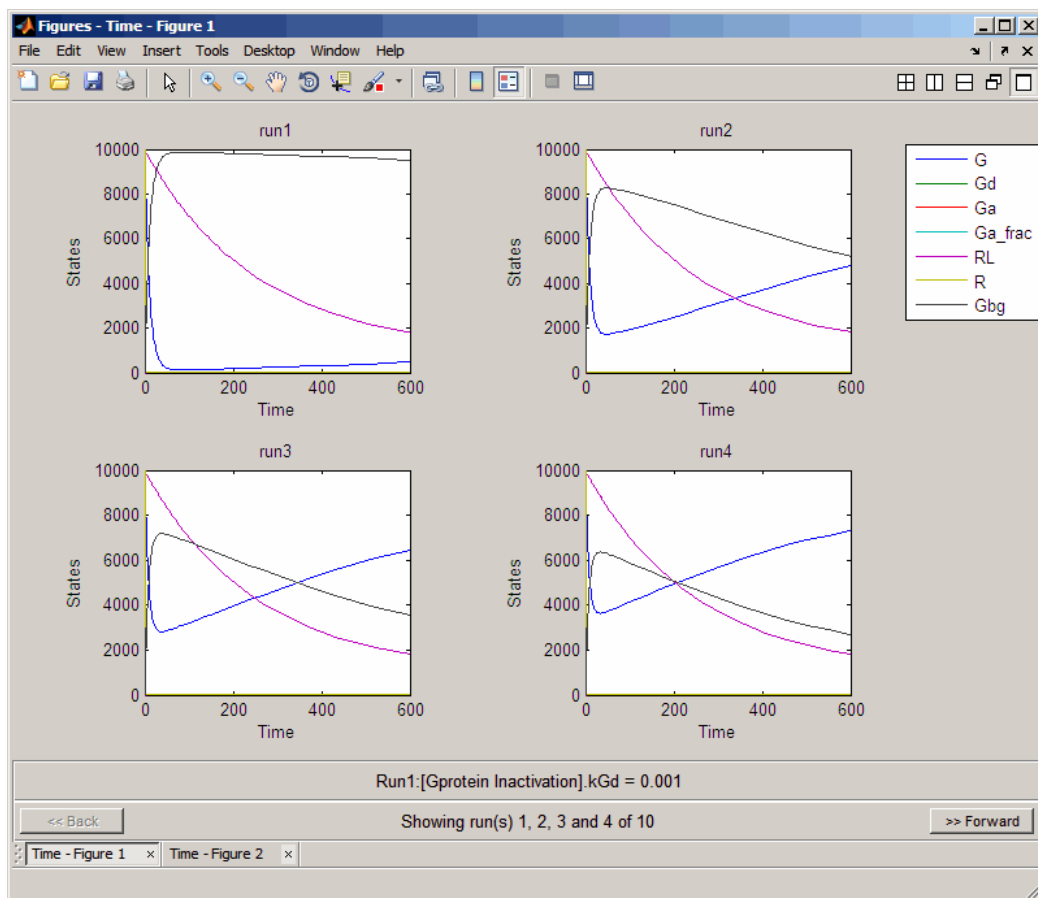
Modeling, Simulation, and Analysis Tools

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  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: Yes
Bug Fixes: Yes

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:

- `sbionlme fitsa` in the SimBiology documentation
- `nlme fitsa` in the Statistics Toolbox documentation
- Pharmacokinetic Modeling Functionality in the SimBiology documentation

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](#) in the SimBiology documentation.

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

Compatibility Considerations: Yes

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

The screenshot shows the SimBiology software interface. The main window is titled "Work Area - My Project \Model Session - G_Protein_wt \Doses". The left pane shows a project tree with "Doses" selected. The main area contains a table of doses and a settings panel for "dose 1".

| Name | Type |
|----------|----------|
| 1 dose | schedule |
| 2 dose 1 | schedule |

Settings for "dose 1":

- Name: dose 1
- Type: schedule
- TargetName: L
- DurationParameterName:
- TimeUnits:
- AmountUnits:
- RateUnits:

Define schedule below or define with MATLAB code: Editor...

| Time | Amount | Rate |
|---------|--------|------|
| 1 2.0 | 9.7 | 0.0 |
| 2 63.5 | 24.6 | 0.0 |
| 3 135.5 | 33.0 | 0.0 |
| 4 | | |

A small graph shows the amount of a substance over time, with a spike at 2.0 units.

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 in the SimBiology documentation.

- 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

Compatibility Considerations: Yes

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. See [Specifying Parameter Transformations](#) in the SimBiology documentation.

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 (`sbionlme fitsa`) function.

For more information see, [Specifying an Error Model in the SimBiology documentation](#).

Functions and Properties Being Removed

Compatibility Considerations: Yes

| 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 85. |

R2009b

Version: 3.1
New Features: Yes
Bug Fixes: Yes

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*.

For models with unchanged topology, analysis tasks involving model simulation now run faster when repeatedly simulating the model. Thus, if you have not changed the structure of the model (number of compartments, reactions, species, or parameters), the time taken to simulate the model is now improved.

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. For more information, see [Obtaining the Status of Fitting \(command line\)](#) or [Obtaining the Status of Fitting \(desktop\)](#).

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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

| Function or Property Name | What Happens When You Use Function or Property? | Use This Instead | Compatibility Considerations |
|--|---|------------------|--|
| <i>-flat</i> as an option for <code>getstoichmatrix</code> and <code>getadjacencymatrix</code> | Errors | Not applicable | Since support for submodels has been removed, this option no longer applies. |

R2009a

Version: 3.0
New Features: Yes
Bug Fixes: Yes

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 104 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 `sbidesktop` 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: Yes
Bug Fixes: Yes

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.

The screenshot shows the SimBiology Reaction Pane interface. At the top, there is a menu bar with 'Build', 'Edit', 'Bind', and 'Unbind' options. Below the menu bar is an 'Enter Reaction:' input field with 'Add' and 'Delete' buttons. The main area contains a table of reactions:

| Name | Reaction | KineticLaw | ReactionRate |
|-------------------------------|------------------------------------|------------|---------------------------------|
| 1 Receptor ligand interaction | $L + R \leftrightarrow RL$ | MassAction | $k_{RL} * L * R - k_{RLm} * RL$ |
| 2 R synthesis/degradation | $R \leftrightarrow \text{null}$ | MassAction | $k_{Rdo} * R - k_{Rs}$ |
| 3 RL degradation | $RL \rightarrow \text{null}$ | MassAction | $k_{RD1} * RL$ |
| 4 Gprotein complex formation | $Gd + Gbg \rightarrow G$ | MassAction | $k_{G1} * Gd * Gbg$ |
| 5 Gprotein activation | $G + RL \rightarrow Ga + Gbg + RL$ | MassAction | $k_{Ga} * G * RL$ |
| 6 Gprotein Inactivation | $Ga \rightarrow Gd$ | MassAction | $k_{Gd} * Ga$ |

Below the table, the 'Settings' tab is selected, showing detailed configuration for the 'Receptor ligand interaction' reaction:

- Reaction:** Reversible
- Reaction:** $L + R \leftrightarrow RL$
- KineticLaw:** MassAction
- Expression:** $(\text{Forward Rate Parameter}) * (\text{MassAction Species}) - (\text{Reverse Rate Parameter}) * (\text{MassAction Species})$
- Map between KineticLaw Parameters and Parameter Names:**

| Kinetic Law Parameter | Parameter Name | Value | Scope | ValueUnits |
|------------------------|----------------|----------|--------------|------------|
| Forward Rate Parameter | kRL | 3.32E-18 | L + R <-> RL | |
| Reverse Rate Parameter | kRLm | 0.01 | L + R <-> RL | |
- Map between KineticLaw Species and Species Names:**

| Kinetic Law Species | Species Name | InitialAmount | Scope | InitialAmountUnits |
|---------------------|--------------|---------------|---------|--------------------|
| MassAction Species | L | 6.022E17 | unnamed | |
| MassAction Species | R | 10000.0 | unnamed | |
- ReactionRate:** $k_{RL} * L * R - k_{RLm} * RL$
- Name:** Receptor ligand interaction
- Active (Select if the reaction is enabled during the simulation.)

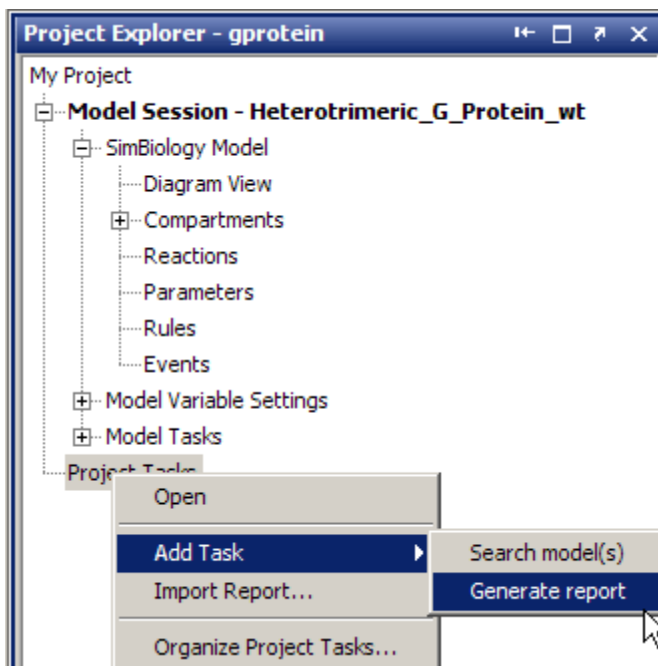
Additional Support for Showing Usages and Generating Reports in the SimBiology Desktop

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

Compatibility Considerations: Yes

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

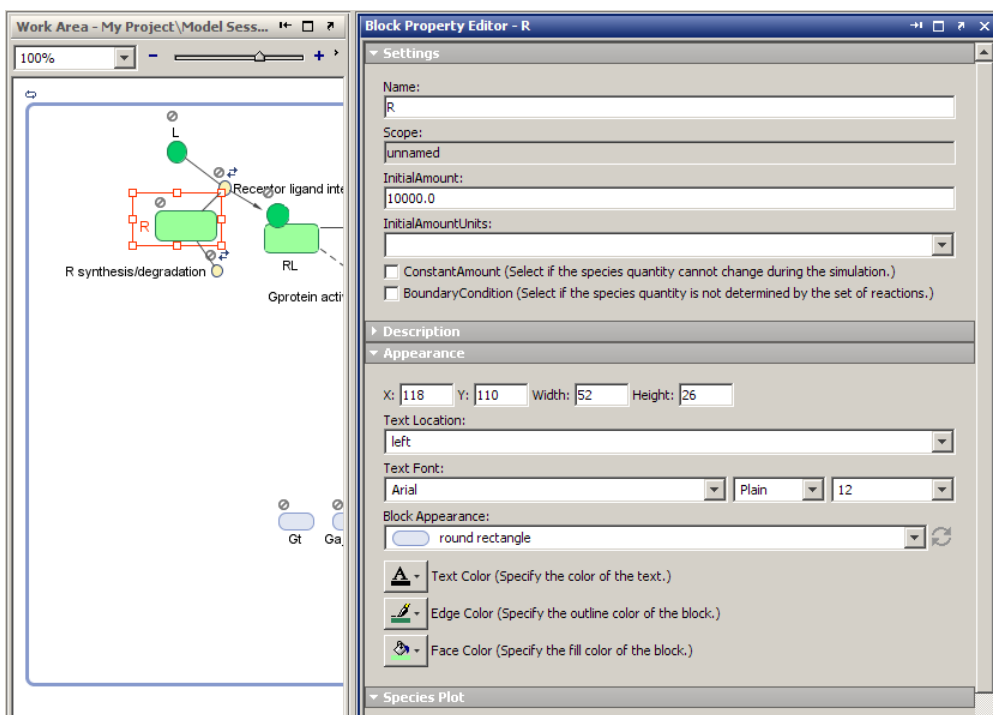
This consideration applies when you previously created custom plot types. Before, you could define the number of species or parameters that could be specified for x or y arguments. This functionality has been removed. You should modify your plot code to enforce this constraint. For an example of enforcing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

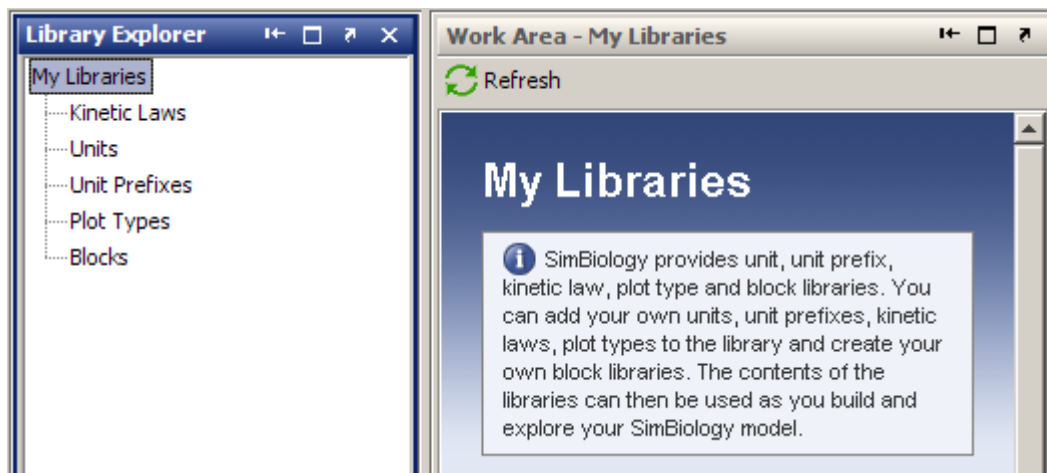
The **Block Property Editor** is a tool that facilitates model building using the **Diagram View** pane by docking next to the diagram and allowing you to change properties of the selected block in the diagram. If you select multiple blocks, you can edit the properties that are common between each block.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



For help about the **Block Property Editor**, open the tool and select **Help > SimBiology Desktop Help** to see the context-sensitive help. You must select a block in the diagram for the **Block Property Editor** and the **SimBiology Desktop Help** to update.

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

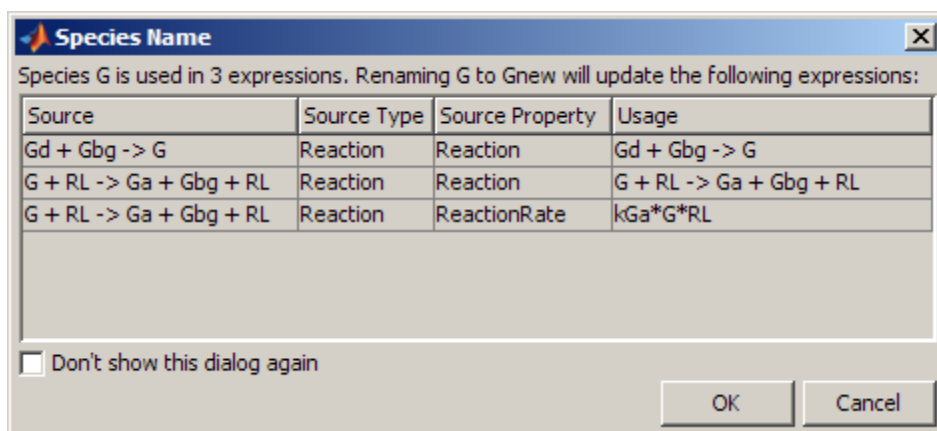
Compatibility Considerations: Yes

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 117.

Change in the Random Number Generator Used During Stochastic Simulations

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

| 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. |
| <i>Model name</i> 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. |

| Function or Property Name | What Happens When You Use Function or Property? | Use This Instead | Compatibility Considerations |
|----------------------------------|--|--|---|
| sbioregisterunit | Errors | Unit object and sbioaddtolibrary | See “New Way to Add Units and Unit Prefixes” on page 143. |
| sbiounregisterunit | Errors | sbioremovefromlibrary | See “New Way to Add Units and Unit Prefixes” on page 143. |
| sbioregisterunitprefix | Errors | UnitPrefix object and sbioaddtolibrary | See “New Way to Add Units and Unit Prefixes” on page 143. |
| sbiounregisterunitprefix | Errors | sbioremovefromlibrary | See “New Way to Add Units and Unit Prefixes” on page 143. |
| BuiltInKineticLaws | Errors | BuiltInLibrary | See “Changes to the Library Structure in the Root” on page 138. |
| BuiltInUnitPrefixes | Errors | BuiltInLibrary | See “Changes to the Library Structure in the Root” on page 138. |
| BuiltInUnits | Errors | BuiltInLibrary | See “Changes to the Library Structure in the Root” on page 138. |
| UserDefinedKineticLaws | Errors | UserDefinedLibrary | See “Changes to the Library Structure in the Root” on page 138. |

| Function or Property Name | What Happens When You Use Function or Property? | Use This Instead | Compatibility Considerations |
|----------------------------------|--|-------------------------|---|
| UserDefinedUnits | Errors | UserDefinedLibrary | See “Changes to the Library Structure in the Root” on page 138. |
| UserDefinedUnitPrefixes | Errors | UserDefinedLibrary | See “Changes to the Library Structure in the Root” on page 138. |

R2008a

Version: 2.3
New Features: Yes
Bug Fixes: Yes

Support for 64-Bit Microsoft Windows

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

Removing Support for Standalone Model Components

Compatibility Considerations: Yes

Functions for creating standalone objects for the following model components will be removed in a future version:

- events (`sbioevent`)
- parameters (`sbioparameter`)
- reactions (`sbioreaction`)
- rules (`sbiorule`)
- species (`sbiospecies`)

All model components must have a relevant parent. Events, reactions, and rules must belong to a model, species to a compartment, and a parameter to a model or a kinetic law. For more information, see .

Compatibility Considerations

If you have scripts that use `sbioevent`, `sbioparameter`, `sbioreaction`, `sbiorule`, or `sbiospecies` for standalone object creation, you must update the script to use the relevant parent object method, as shown in “Functions and Properties Being Removed” on page 126.

Functions and Properties Being Removed

Compatibility Considerations: Yes

| Function or Property Name | What Happens When You Use Function or Property? | Use This Instead | Compatibility Considerations |
|--|---|---------------------------|---|
| <code>sbioevent</code> | Warns | <code>addevent</code> | Events must belong to a model. Replace all existing instances of <code>sbioevent</code> with <code>addevent</code> . |
| <code>sbioparameter</code> | Warns | <code>addparameter</code> | Parameters must belong to a model or a kinetic law. Replace all existing instances of <code>sbioparameter</code> with <code>addparameter</code> . |
| <code>sbioreaction</code> | Warns | <code>addreaction</code> | Reactions must belong to a model. Replace all existing instances of <code>sbioreaction</code> with <code>addreaction</code> . |
| <i>Model name</i> as an input argument for <code>sbioroot</code> | Errors | <code>sbioselect</code> | <code>sbioroot</code> does not accept a model name as an input argument. Use <code>sbioselect</code> to query models by name. |
| <code>sbiorule</code> | Warns | <code>addrule</code> | Rules must belong to a model. Replace all existing instances of <code>sbiorule</code> with <code>addrule</code> . |
| <code>sbiospecies</code> | Warns | <code>addspecies</code> | Species must belong to a compartment. Replace all existing instances of <code>sbiospecies</code> with <code>addspecies</code> . |

| Function or Property Name | What Happens When You Use Function or Property? | Use This Instead | Compatibility Considerations |
|----------------------------------|--|--|---|
| sbioregisterunit | Errors | Unit object and sbioaddtolibrary | See “New Way to Add Units and Unit Prefixes” on page 143. |
| sbiounregisterunit | Errors | sbioremovefromlibrary | See “New Way to Add Units and Unit Prefixes” on page 143. |
| sbioregisterunitprefix | Errors | UnitPrefix object and sbioaddtolibrary | See “New Way to Add Units and Unit Prefixes” on page 143. |
| sbiounregisterunitprefix | Errors | sbioremovefromlibrary | See “New Way to Add Units and Unit Prefixes” on page 143. |
| BuiltInKineticLaws | Errors | BuiltInLibrary | See “Changes to the Library Structure in the Root” on page 138. |
| BuiltInUnitPrefixes | Errors | BuiltInLibrary | See “Changes to the Library Structure in the Root” on page 138. |
| BuiltInUnits | Errors | BuiltInLibrary | See “Changes to the Library Structure in the Root” on page 138. |
| UserDefinedKineticLaws | Errors | UserDefinedLibrary | See “Changes to the Library Structure in the Root” on page 138. |

| Function or Property Name | What Happens When You Use Function or Property? | Use This Instead | Compatibility Considerations |
|----------------------------------|--|-------------------------|---|
| UserDefinedUnits | Errors | UserDefinedLibrary | See “Changes to the Library Structure in the Root” on page 138. |
| UserDefinedUnitPrefixes | Errors | UserDefinedLibrary | See “Changes to the Library Structure in the Root” on page 138. |

R2007b+

Version: 2.2
New Features: Yes
Bug Fixes: Yes

Changes to the Model Structure

Compatibility Considerations: Yes

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

- “Compartments Now Supported” on page 130
- “Submodel Support Will Be Removed” on page 130

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 `sbiupdate`.

If you open a model that contains submodels in the SimBiologydesktop, the submodels are automatically converted to models and placed in separate model sessions. See `sbiupdate` 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.

For more information on setting up models with events in the desktop select **Help > SimBiology Desktop Help** to see the context-sensitive help for tasks in the desktop.

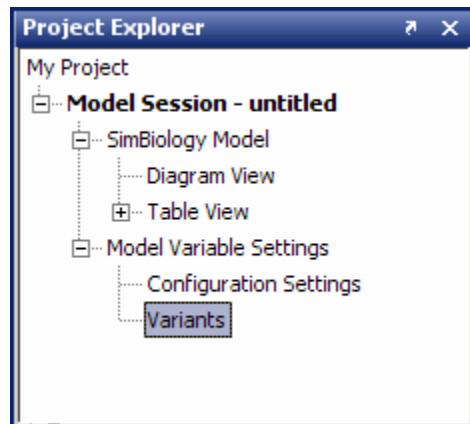
Variants

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**.



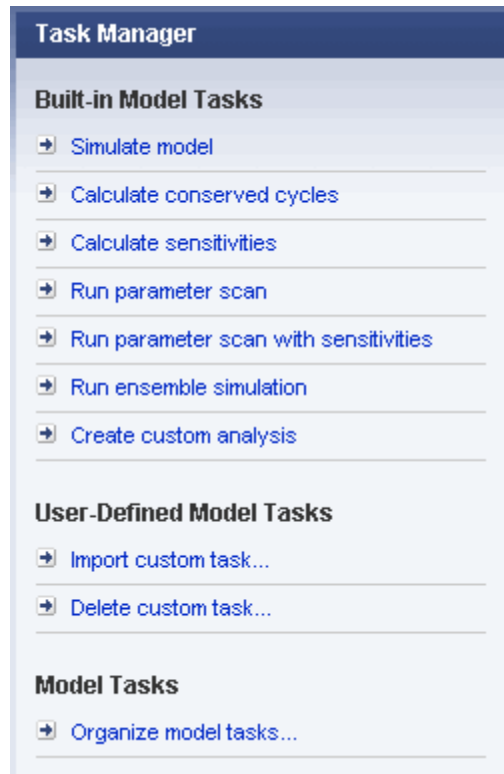
Support for Analysis Tasks in the Desktop

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

- “Task Manager” on page 134
- “Sensitivity Analysis in the Desktop” on page 135
- “Scanning and Scanning with Sensitivities in the Desktop” on page 136
- “Ensemble Simulation Runs in the Desktop” on page 136
- “Conserved Cycle Calculations in the Desktop” on page 137
- “Create Custom Analysis Tasks” on page 137
- “Generate Reports for Projects” on page 137

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 information on how to add the task to a model, see “Task Manager” on page 134. 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 for tasks in the desktop. You need 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 134. 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 134. 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 SimBiologydesktop 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 134. 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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

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

- “Support for Sundials Solvers” on page 139
- “New Property in Configuration Sets to Specify Species Dimensions” on page 139
- “SimData Object Holds All Simulation Data” on page 139

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`.

Compatibility Considerations

??? Describe if this new property has backwards compatibility implications — Nothing to document here according to tech reviews

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 139).

`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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

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

Compatibility Considerations: Yes

| Function or Property Name | What Happens When You Use Function or Property? | Use This Instead | Compatibility Considerations |
|--|---|---|---|
| <code>addmodel</code> | Warns | <code>addcompartment</code> , where applicable | See “Submodel Support Will Be Removed” on page 130. |
| Models | Still runs | Compartments, where applicable | See “Submodel Support Will Be Removed” on page 130. |
| <i>Model name</i> as an input argument for <code>sbioroot</code> | Warns | <code>sbioselect</code> | <code>sbioroot</code> does not accept a model name as an input argument. Use <code>sbioselect</code> to query models by name. |
| <code>sbioregisterunit</code> | Warns | Unit object and <code>sbioaddtolibrary</code> | See “New Way to Add Units and Unit Prefixes” on page 143. |
| <code>sbionregisterunit</code> | Warns | <code>sbioremovefromlibrary</code> | See “New Way to Add Units and Unit Prefixes” on page 143. |
| <code>sbioregisterunitprefix</code> | Warns | UnitPrefix object and <code>sbioaddtolibrary</code> | See “New Way to Add Units and Unit Prefixes” on page 143. |

| Function or Property Name | What Happens When You Use Function or Property? | Use This Instead | Compatibility Considerations |
|---------------------------------------|--|------------------------------------|---|
| <code>sbiounregisterunitprefix</code> | Warns | <code>sbioremovefromlibrary</code> | See “New Way to Add Units and Unit Prefixes” on page 143. |
| <code>sbiogetsensmatrix</code> | Warns | <code>getsensmatrix</code> | See “SimData Object Holds All Simulation Data” on page 139. |
| <code>sbiogetnamedstate</code> | Warns | <code>selectbyname</code> | See “SimData Object Holds All Simulation Data” on page 139. |
| <code>BuiltInKineticLaws</code> | Warns | <code>BuiltInLibrary</code> | See “Changes to the Library Structure in the Root” on page 138. |
| <code>BuiltInUnits</code> | Warns | <code>BuiltInLibrary</code> | See “Changes to the Library Structure in the Root” on page 138. |
| <code>BuiltInUnitPrefixes</code> | Warns | <code>BuiltInLibrary</code> | See “Changes to the Library Structure in the Root” on page 138. |
| <code>UserDefinedKineticLaws</code> | Warns | <code>UserDefinedLibrary</code> | See “Changes to the Library Structure in the Root” on page 138. |

| Function or Property Name | What Happens When You Use Function or Property? | Use This Instead | Compatibility Considerations |
|----------------------------------|--|-------------------------|---|
| UserDefinedUnits | Warns | UserDefinedLibrary | See “Changes to the Library Structure in the Root” on page 138. |
| UserDefinedUnitPrefixes | Warns | UserDefinedLibrary | See “Changes to the Library Structure in the Root” on page 138. |

R2007b

Version: 2.1.2
New Features: No
Bug Fixes: No

No New Features or Changes

R2007a

Version: 2.1.1
New Features: No
Bug Fixes: No

No New Features or Changes

R2006b+

Version: 2.1
New Features: Yes
Bug Fixes: Yes

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

Compatibility Considerations: Yes

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 SimBiologydesktop 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
New Features: No
Bug Fixes: Yes

R2006a+

Version: 2.0
New Features: Yes
Bug Fixes: Yes

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

Compatibility Considerations: Yes

- **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.

For an introduction and an example, see [Estimating Parameters in the SimBiology User's Guide](#).

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:

- `sbioenssemblerun` — 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 `sbioenssemblerun`.
- `sbioensemblestats` — Gets mean and variance as a function of time for all the species in the ensemble data generated by `sbioenssemblerun`.

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
New Features: No
Bug Fixes: No

Rules

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

R14SP3+

Version: 1.0
New Features: Yes
Bug Fixes: No

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, $2 R \rightarrow P$ 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.
- Dimensional analysis cannot evaluate rules or rate equations that contain MATLAB functions.
- 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.

If you are using a species that is not a valid MATLAB variable name, enclose the name in square brackets when writing the reaction rate equation or rules. When creating and assigning species or parameter names, however, you should enter the name without any brackets. For example, enclose `[DNA polymerase+]` and `[K_DNA polymerase+]` within brackets in reaction rates and rules; enter `DNA polymerase+` or `K_DNA polymerase+` when specifying the name of the species or 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.