



COMSOL Multiphysics

Release Notes

COMSOL Multiphysics Release Notes

© 1998–2015 COMSOL

Protected by U.S. Patents listed on www.comsol.com/patents, and U.S. Patents 7,519,518; 7,596,474; 7,623,991; 8,219,373; 8,457,932; and 8,954,302. Patents pending.

This Documentation and the Programs described herein are furnished under the COMSOL Software License Agreement (www.comsol.com/comsol-license-agreement) and may be used or copied only under the terms of the license agreement.

Support for implementation of the ODB++[®] Format was provided by Mentor Graphics Corporation pursuant to the ODB++ Solutions Development Partnership General Terms and Conditions. ODB++ is a trademark of Mentor Graphics Corporation.

COMSOL, COMSOL Multiphysics, Capture the Concept, COMSOL Desktop, LiveLink, and COMSOL Server are either registered trademarks or trademarks of COMSOL AB. All other trademarks are the property of their respective owners, and COMSOL AB and its subsidiaries and products are not affiliated with, endorsed by, sponsored by, or supported by those trademark owners. For a list of such trademark owners, see www.comsol.com/trademarks.

Version: COMSOL 5.1

Contact Information

Visit the Contact COMSOL page at www.comsol.com/contact to submit general inquiries, contact Technical Support, or search for an address and phone number. You can also visit the Worldwide Sales Offices page at www.comsol.com/contact/offices for address and contact information.

If you need to contact Support, an online request form is located at the COMSOL Access page at www.comsol.com/support/case.

Other useful links include:

- Support Center: www.comsol.com/support
- Product Download: www.comsol.com/product-download
- Product Updates: www.comsol.com/support/updates
- Discussion Forum: www.comsol.com/community
- Events: www.comsol.com/events
- COMSOL Video Gallery: www.comsol.com/video
- Support Knowledge Base: www.comsol.com/support/knowledgebase

Part number: CM010001



Release Notes

COMSOL[®] version 5.1 includes an updated and integrated Application Builder, a more powerful COMSOL Server[™], and updates and improvements for all COMSOL products. These *Release Notes* provide information regarding new functionality in version 5.1 for all COMSOL products.

COMSOL Multiphysics

General New Functionality

GENERAL SEARCH OF THE MODEL TREE

You can now search the entire model tree for variables or expressions. Press Ctrl+F to open the **Find** dialog box. The **Find Results** window then lists in which node in the model tree the search string appears. From here, you can click to move, for example, to the **Settings** window for that node. The search functionality can be useful when you need to find an invalid expression that causes the simulation to fail.

OPEN ALL MPH-FILES

It is now possible to open MPH-files that include functionality from add-on modules and other COMSOL products that are not included in your COMSOL license. However, to re-solve such an MPH-file you need to remove or disable the functionality that is not included in your license.

MERGED MODEL LIBRARIES AND APPLICATION LIBRARIES WINDOWS

The **Model Libraries** window and the **Application Libraries** window have merged into the **Application Libraries** window, which contains all MPH-files that ship with the COMSOL products: models that open in the Model Builder and runnable applications with custom user interfaces created using the Application Builder (available in the **Demo Applications** folder for most products).

INSTALL ALL APPLICATION LIBRARIES AND ALL DOCUMENTATION

During installation, you now have the possibility to install the application libraries and the documentation for all products, not just the ones available with your COMSOL license. You can also choose to not install any documentation and instead access the documentation for the products in your COMSOL license online.

SORTABLE TABLES

You can now click the column headers in tables to sort the contents of the settings for **Parameters** and **Variables**, the **Variables** table in the **Equation View** nodes, and the search results in the **Find Results** window. The sorting is done alphabetically in these steps: ascending, descending, and original order.

PERMANENT SECURITY SETTINGS

If security preferences are set permanently during the installation (stored in a `security.prefs` file in the COMSOL root), the corresponding security settings in the **Preferences** dialog box are disabled.

New Functionality in the Application Builder

INTEGRATED APPLICATION BUILDER TOOLS

The Application Builder is now fully integrated into the COMSOL Desktop®. You can open form windows and method windows in the COMSOL Desktop and work with forms and methods in an integrated environment. To switch between the Application Builder and the Model Builder, and vice versa, you can use the keyboard shortcuts `Ctrl+Alt+M` and `Ctrl+Alt+A`, respectively. It is also possible to use a separate desktop window for the Application Builder, which can be specified as a preference setting on the **General** page in the **Preferences** dialog box.

NEW DEMO APPLICATIONS

About 20 new demo applications are available in **Demo Applications** folders in the **Application Libraries** window. They are all runnable applications with custom interfaces that demonstrate how to use the Application Builder.

MERGED MPH-FILE FORMAT

The MPHAPP-file and MPH-file formats have been merged, and all application files use the `.mph` file suffix. You can still open MPHAPP-files created in version 5.0.

TEST APPLICATIONS IN WEB BROWSERS

You can test launch applications directly in your default browser or in another installed browser to check what the application looks like when run using a web client.

IMPROVED RIBBON NAVIGATION

The **Main**, **Test**, and **View** ribbon tab sections are now available in all three Application Builder ribbon tabs. The **Preview Form** button has been moved from the **Home** to the **Form** tab.

FILE LIBRARIES IMPROVEMENTS

The tables in the library nodes (**Images**, **Sounds**, and **Files**) now have an **Export Selected Image File** button to, for example, save the file in the library to the file system.

IMPROVEMENTS TO THE FORM EDITOR

- The New Form wizard replaces the Application Wizard from version 5.0 and provides a quick starting point for adding inputs, outputs, plots, and buttons to a form.
- Copy-paste operations for forms and form objects can now be done between sessions and within one session after loading another application.
- You can send emails from applications so that users can receive emails when the computation has finished, for example.
- Improved file handling: It is now possible to directly select a file setting in any of the nodes in the model that expects a file as input. In addition, it is also possible to call a command or method that does the same thing as the **Browse** button in the **File Import** node.
- You can now add background images to forms.
- The grid mode for designing forms has been improved so that you can
 - Apply row and column actions to multiple items at the same time.
 - Apply row and column actions from selected cells context menus,
 - Copy, paste, and duplicate rows and columns.
 - Click **Rows & Columns** to set the size of the grid.
 - Click **Extract Subform** to extract a rectangular area from the current **Form** object into a new **Form** and replace the selection in the original form with a **Form Reference** to the new **Form**.
 - Use an **Inherit columns** setting, which makes certain form objects inherit their column settings from a parent Form.
 - Move the selected cell within the grid in the form windows using the arrow keys (holding Shift creates a rectangular selection).
- A new **Information Card Stack** form object makes it possible to display information about the solution status or other information that can be useful in an application. The cards can display an icon and a label with different text depending on the value of a data source.
- The ability to display information about the progress of an application has been enhanced so that Progress Bar form objects can be set to show model progress (main COMSOL progress) in addition to being controlled directly by application language methods. Additionally, there is an option for adding a **Cancel** button to a Progress Bar form object.

- You can now select multiple form objects and specify common properties for all selected objects.
- It is now possible to add custom toolbar menu items to table toolbars in applications.
- The **Input Field**, **Slider**, and **Data Display** form objects as well as ribbon buttons now support adding custom tooltips.
- Text alignment is now available for Input Field and Text form objects in the **Appearance** section of their **Settings** windows.
- You can make tables in applications sortable by selecting the **Sortable** check box in the tables' **Settings** windows.
- Tables (from **Results>Tables** in the model tree) can now be used as sources for **Results Table** form objects.
- You can now load and save data to and from tables in the declaration nodes.
- An **Events** section is now included in the **Settings** window for tables. You can use it to add data change event handlers to tables.

IMPROVEMENTS TO THE METHOD EDITOR

- You can now use code folding to show and hide parts of the code for better overview.
- You can use Ctrl+Pause to stop method execution when testing applications.
- Built-in functions such as `with()` have a special syntax highlighting color.
- When you call another method from within a method, Ctrl+Alt+double-click opens the method window for that method.
- There are a number of new methods, including methods for:
 - Enabling and disabling ribbon, menu, and main toolbar items from methods
 - Controlling the progress information
 - Inserting, appending, and removing 1D and 2D arrays
 - Setting and retrieving file paths and filenames
 - Sending emails
 - Saving a model to file
 - Setting and retrieving the last computation time

For details and a full list of methods, see the *Introduction to Application Builder*.

New Functionality in COMSOL Server

- COMSOL Server now supports a network with one primary server and one or more secondary servers, which makes it possible to run applications in a more distributed way.
- The Application Library page now provides three different views: Grid View, Detailed View, and List View for improved overview of the applications.
- A COMSOL Server administrator can specify an email address for each user so that they can receive emails from applications.
- In the default settings, you can now set a default email address as the sender address for emails. You can also specify if you want to include the demo applications that ship with the COMSOL products in the lists of available applications.
- On Linux[®], `comsol service` is a new shortcut for `comsol server -silent -login never &` to simplify starting COMSOL Server in the background

New Geometry and Mesh Functionality

PART LIBRARIES

The new **Part Libraries** window provides the possibility to save parameterized geometry sequences as predefined parts that you can use when creating geometries. COMSOL Multiphysics and some of the add-on modules include part libraries with geometry parts common for components within the application areas for those products. You can also define your own parts.

GEOMETRY SUBSEQUENCE IMPROVEMENTS

In addition to the **Part Libraries** window, the following changes apply to the geometry subsequences in earlier versions:

- From the main **Parts** node under **Global Definitions**, you can add 3D, 2D, and 1D Part nodes, for which you define a geometry sequence and the input parameters for a geometry part.
- Parts that you insert into a geometry sequence appear as **Part Instance** nodes. When inserting parts, you can choose from existing parts in the **Parts** submenu that you find on the **Geometry** nodes' context menus, or you can open the **Part Libraries** window.
- For defining geometry sequences, especially for parts, a new **Parameter Check** node is available on the **Programming** submenu. You can use it to check the value of an

input parameter and to issue an error with a custom error message if the check returns false.

CUMULATIVE SELECTION IMPROVEMENTS

If the geometry sequence has cumulative selections, a **Cumulative Selections** node appears under the geometry node. You can right-click that node and select **Cumulative Selection** to add a cumulative selection nodes. Additionally, cumulative selections are no longer removed automatically. Instead, you can delete them like other nodes in the model tree.

IMPROVED MESH IMPORT

- To recreate selections made for a NASTRAN mesh, the mesh import now creates **Explicit Selection** nodes in the current component with the selections corresponding to the groups of elements defined by the property IDs in the NASTRAN file.
- Automatic partitioning is now available to quickly partition complex boundaries when importing meshes. In the **Import** node's settings, you can choose to use an automatic or minimal boundary partitioning, or to detect faces. You can specify that the mesh import detects planar faces and also set the values of the associated parameters. There is also a separate **Detect Faces** node where you can, for example, apply boundary partitioning to a part of the geometry.

COPY MESHES OF DIFFERENT SPACE DIMENSIONS

It is now possible to use the **Copy** feature to copy meshes between different components of different space dimensions. However, the entity dimension must be the same (for example, you can copy a 2D domain mesh to a 3D face).

PARTITION EDGES IN 2D

The **Partition Edges** geometry operation is now available in 2D as well as 3D.

New Operators, Functions, and Definitions

- The new `withsol` operator can evaluate an expression using any solution in the current model. Using this operator makes it possible to access solutions in applications that include multiple study steps, for example.
- The new `timemax`, `timemin`, `attimemax`, and `attimemin` operators can compute the maximum or minimum over some expression evaluated at a number of time steps.

- The `timeint` and `timeavg` operators have a new syntax with a `nointerp` flag to evaluate the time integral without doing any interpolation between the stored time steps. This method might be less accurate, but it is faster and more robust.
- The **Mass Properties** feature can now use the density (or other kinds of mass contributions) directly from the physics interfaces.
- In the **Curvilinear Coordinates** interface, a flux condition is now available as an option in the settings for an **Inlet** node under a **Diffusion Method** node. With a flux condition, a potential could represent, for instance, the phase along the path length.

New Functionality in Studies and Solvers

GOAL-ORIENTED ERROR ESTIMATION

Variables can now be computed for evaluating goal-oriented error estimates, which can help to assess the accuracy of the solution to stationary and time-dependent simulations.

DORMAND-PRINCE 5 TIME-DEPENDENT SOLVER

The new Dormand-Prince 5 time-dependent solver is an explicit solver from the Runge-Kutta family of methods. It is efficient for solving systems of ODEs, for example. The new Lorenz Attractor example under **Equation Based** in the COMSOL Multiphysics Application Library uses the Dormant-Prince 5 solver to solve a set of ODEs.

OTHER IMPROVEMENTS

- Improved performance when solving ODEs and for shared-memory matrix algorithms.
- A new **Evaluate Derived Values** node is available under **Results** for batch simulations, for example. It provides improved handling of results data in tables for job sequences. The previous **Derived Values** node is available in this version for backward compatibility.
- Improved progress information from the solvers.
- In the **Advanced** node under the solver node, in the **Assembly Settings** section, you can now select the **Check for undefined numerical values after each operation** check box to make COMSOL check intermediate results for undefined numerical values (for example, from numerical overflow). This option provides more accurate error messages when such undefined numerical values occur.

- Improved recovery file handling:
 - The MPH-file name is now included.
 - The list of recovery files can be sorted.
 - You can select multiple recovery files.
- There is now a distributed version of the Anderson acceleration, which can be useful when solving transport problems in a cluster environment, for example.

New and Improved Results and Visualization Functionality

- A new **Trajectory Plot** plot type is available for plotting particle trajectories. The new Lorenz Attractor example under Equation Based in the COMSOL Multiphysics Application Library uses a Trajectory Plot to visualize the trajectories.
- In the **Graphics** window's toolbar, toggle buttons are available to turn the grid, legend, and axis orientation arrows on and off.
- The layout of **Graphics** windows has been improved to make plots look better, especially when using a small graphics canvas. Especially, in a plot group's **Number Format** section, you can specify a custom number format to allow scientific, decimal, or automatic formatting with a specified precision for the color legend and the grid.
- The **Animation** and **Player** nodes have been merged for improved flexibility. From the **Export** menu you can choose **Animation** and **Player**, where both options create an **Animation** node, but with the new **Target** option set to **Animation** and **Player**, respectively. Old Player nodes are converted to **Animation** nodes with **Target** option set to **Animation** and **Player**.
- New **Array 2D** and **Array 3D** data sets create arrays with copies of some solution data set, which you can use to visualize some periodic pattern, for example.
- New **Time Integral** and **Time Average** data sets are available. They make it possible to compute time integrals and averages of expressions without using operators or model couplings.
- The **Grid 1D**, **Grid 2D**, and **Grid 3D** data sets replace the previous **Function** data sets. They provide domain evaluations of global functions also where there is no domain mesh, which can be useful for evaluating far-field and BEM operators.
- The **Parameterized Curve** and **Parameterized Surface** data sets now support evaluation where there is no domain, which can be useful, for example, for BEM modeling.
- You can now plot 1D meshes using 1D **Mesh** plots.

- In result **Table** windows, you can now choose to use mixed (automatic), scientific, or decimal notation of the numerical data by clicking buttons for these notations in the **Table** windows' toolbars.
- In the settings for an **Image Snapshot**, there is now predefined **Size** settings for **Manual (web)** and **Manual (print)** with suitable default image sizes set for when images will appear either on the web or in print.
- A **Global Matrix Evaluation** node is available under **Derived Values** for evaluation of any matrix quantity. For example, you can use it to evaluate piezoelectric material properties in global and local coordinate systems.
- In the **Sector 3D** data sets you can now choose which sectors to include to provide more flexibility when plotting multiple sectors.
- It is now possible to choose between integration and summation in the following data series operations: Average, Integral, RMS, Standard deviation, and Variance.
- The following new functionality is available in Report Generator:
 - You can add report contents from the **Declaration** submenu, which takes information from the declared variables in the Application Builder. For declarations and variables in the model tree, you can also add **Custom Column Table** and **Custom Row Table** nodes from the **Basic** submenu to create custom tables where you can specify the data to include in the rows and columns.
 - When including images in reports, an extra large image size is available.
 - You can specify a custom number format to allow scientific, decimal, or automatic formatting with a specified precision.
 - For the front matter, you can choose the report date as **Current**, **None**, or **Custom**.
- Progress information is now available for lengthy plot operations so that you can monitor the progress and cancel the plot if needed.
- It is now possible to plot a line graph for a time-adaption solution even if the set of times to plot for correspond to two or more different meshes.
- In **Particle Trajectories** plots, **Ellipse** is a new option in the **Type** list for plotting the polarization of rays by placing an ellipse at the position of each particle.

DEMO APPLICATION: TRAVELING LOAD SIMULATION

In this application, the response to a set of load pulses traveling along a beam placed on equidistant supports is computed. The beam geometry, the speed and width of the load pulses, and their spacing can be varied.

TRAVELING LOAD

This example shows how to model a load which varies in space and time. A series of load pulses travel along a beam which is supported at equal distances. For some combinations of the traveling speed of the load pulses and the spacing between them, it is possible to excite resonances in the beam. The effects of four different combinations of these parameters are investigated.

LORENZ ATTRACTOR

A Lorenz attractor is an attractor that is described by a system of ordinary differential equations: the Lorenz system. In the early 1960s, Lorenz discovered the chaotic behavior of this system for certain parameter values and initial conditions. The solution, when plotted as a phase space, resembles the figure eight. This example uses the Dormand-Prince 5 explicit method for solving the ODEs and a Point Trajectories plot for visualizing the Lorenz attractor.

General Backward Compatibility Considerations

COMSOL 5.1 is backward compatible with COMSOL versions 4.0–5.0. To open models created using COMSOL version 3.5a in version 5.1, follow these steps:

- 1 Open and then save the COMSOL 3.5a mode using any of the COMSOL versions 4.0–4.4.
- 2 Open the model saved in versions 4.0–4.4 in COMSOL 5.1.

Backward Compatibility with Version 5.0

MESH IMPORT

The method used to automatically partition the boundary of imported meshes in 3D has been improved. If you have an existing model, you can work with it without being affected. However, if you click the **Import** button of the **Import** feature in the meshing sequence, the file is read again and the new partitioning method is used. The exception

is if the **Import** feature had **Boundary partitioning** set to **Manual**. In this case, the modified parameter values are preserved under the **Feature detection** setting, which uses the same algorithm as in version 5.0.

COMSOL tries to map the old selection on boundaries to the new boundaries, but it is not always possible to do accurately when new faces have appeared or old faces have disappeared. You may have to manually review and update boundary, edge, and point selections after reimporting the mesh.

If you have a Java[®] or MATLAB[®] program that imports meshes, the number of geometric entities may have changed compared to older versions.

SECURITY SETTINGS

In version 5.1, the **Allow external process and libraries** check box on the Security page in the Preferences dialog box is cleared by default to not allow applications to start external processes on the computer. The default setting in 5.0 was set to allow such external processes.

DISPLAY OF MAX/MIN MARKERS IN PLOTS

The display of max/min markers in plots is now off by default to make plots fit better when using a small graphics canvas. In previous versions, the display of max/min markers in plots was on by default.

Backward Compatibility with Version 4.4

COMSOL SERVER

The COMSOL Server from version 4.4 is now called the COMSOL Multiphysics Server.

MESH IMPORT

An edge in an imported mesh that has a common start and end vertex, or that lacks start and end vertices, is now split into two edges with distinct start and end vertices. For meshes where this happens, the numbering of all geometric entities may change when the mesh is rebuilt.

For MPH-files created in earlier versions, selections are automatically updated with the new entity numbers.

LOCAL COORDINATE VALUES

For model components created in version 5.0 and onward, the variable names `xi1`, `xi2`, and `xi3` are reserved for the predefined local coordinate variables. To access the previous behavior (that local coordinate variables do not exist), you can use the API method `model.modelNode(<tag>).defineLocalCoord(false)`.

TRANSPORT OF DILUTED SPECIES

Running Java[®]-files from previous versions may fail due to the new default name for the Transport of Diluted Species interface. This can be avoided by adding a command that specifies the identifier in accordance with the name of the interface. For example, when creating an interface using the old name (`chds`):

```
model.physics().create("chds", "DilutedSpecies", "geom1",
    new String[][]{{"c"}});
```

Add the following line to specify the identifier accordingly:

```
model.physics("chds").identifier("chds");
```

The Java[®] API syntax for creating and accessing vectors and tensors in the Transport of Diluted Species interface has changed as well as the syntax for settings physics properties. This does not affect MPH-files. See the backward compatibility notes for the Chemical Reaction Engineering Module for additional information that also applies to the Transport of Diluted Species interface.

DOCUMENTATION

The *COMSOL Multiphysics Programming Reference Manual* replaces the *COMSOL API for use with Java[®] Reference Manual*.

Backward Compatibility with Version 4.3b

MODEL NODES ARE NOW COMPONENT NODES

The **Model** nodes (as they were called in previous versions), which contained separate model components in a model file, are called **Component** nodes in version 5.0.

CHANGES TO PARAMETRIC SWEEPS

Old models that use stationary parametric sweeps are loaded with the **Reuse solution for previous step** list set to **Yes**. The **Run continuation for** list is set to the parameter used, unless the continuation algorithm would not have been used for this model in previous versions (for example, if multiple parameters are used or if the parameter list is not monotonous).

CHANGES TO THE PHYSICS SELECTION IN STUDY SETTINGS

The names of the states of a physics in the physics tree, which you can modify under **Physics and Variables Selection** in the study steps' settings windows, have changed:

- **Provide Degrees of Freedom** is now called **Disable in Solvers**.
- **Disable** is now called **Disable in Model**.

REVISED FORMULATION FOR LAMINAR INFLOW AND LAMINAR OUTFLOW

The formulations of the laminar inflow and laminar outflow conditions have been corrected. The modified formulation gives a more accurate mass flux. Some models may now produce a slightly different flow field.

Laminar inflow and laminar outflow are available in the fluid flow physics interfaces in the following modules:

- Batteries & Fuel Cells Module
- CFD Module
- Corrosion Module
- Electrochemistry Module
- Electrodeposition Module
- Heat Transfer Module
- Microfluidics Module
- Plasma Module
- Subsurface Flow Module

NEW DEFAULT FOR GRAPHICS OPTIMIZATION

Under **Graphics and Plot Windows** in the **Preferences** dialog box, the default settings in the **Optimize for** list is now **Quality** instead of **Performance**.

VELOCITY/ACCELERATION INTEGRATION VARIABLE

For the Solid Mechanics interface (and all related multiphysics interfaces) and the Truss interface, a help variable u_0 (velocity integration variable) is used in **Prescribed Velocity** and **Prescribed Acceleration** features for Time Dependent study types. This variable computes the displacement for each point where the condition is prescribed. It is changed to use the full feature scope in order to avoid collisions in cases where several such features exist within the same model component. As a result, when opening and running an old model that uses such features together with a segregated solver, an error message appears, stating that not all dependent variables occur in at

least one of the segregated solver steps. The relevant action is to manually add the velocity integration variable to the segregated step containing the corresponding displacement field. Alternatively, you can regenerate any affected solver sequence.

MATH LIBRARIES ON AMD PROCESSORS

MKL is now the default math library on AMD processors. Switching to the ACML math library for AMD processors might improve performance in some cases.

Backward Compatibility with Version 4.3a

ERROR ESTIMATION CHANGES IN THE SOLVERS

The **Automatic** method for **Check error estimate** for direct linear solvers and for **Validate error estimate** has changed. For nonlinear problems and for time-dependent problems, the underlying nonlinear solver does not accept termination for a linear solution step that does not fulfill the error estimate (unless the step size is very small). The motivation for this change is that the old method can lead to premature termination of the nonlinear solution process, which in some cases introduces a large error for the computed solution.

This change can cause the solvers to take more nonlinear steps for stationary problems and more time steps for time-dependent problems, and it may also lead to convergence problems. For such cases, use the **No** method to obtain the old behavior. However, doing so can hide numerical problems and potentially lead to large numerical errors.

TERMINATION CRITERION FOR STATIONARY SOLVERS

The default termination criterion for stationary solvers has changed. In 4.3a the settings corresponded to **Solution**; now the default is **Solution or residual**. This change in default termination criterion might affect models created in earlier versions of COMSOL if you regenerate the solver sequence in 5.0.

BACKWARD EULER INITIALIZATION TIME STEP

A new setting in the **Advanced** section of the settings window for the **Time-Dependent Solver**, called **Fraction of initial step for Backward Euler**, provides an option for entering a dimensionless quantity that determines the size of the time step for the backward Euler method (in terms of the initial step). This value can improve the accuracy of the initialization step but can also affect the start-up of some models. The default value is 0.001 (this differs from earlier versions, which used a value of 1). When opening models created in version 4.3a or earlier, the value for this fraction is set to 1 to maintain compatibility with those versions.

LOADING EXTERNAL PHYSICS BUILDER JAR FILES

External physics builder JAR archives compiled with earlier versions of COMSOL include a manifest file that contains a reference to the CDO library, which has been removed. To load such JAR files in version 5.0, you must first do one of the following:

- Delete the META-INF/MANIFEST.MF file in the archive source on the file system, and then recompile the JAR archive.
- Manually remove the line with `org.eclipse.emf.cdo` in the META-INF/MANIFEST.MF file in the JAR archive. You can do so directly in a file archive manager such as 7-Zip or similar.

HIGHLIGHTING GEOMETRY OBJECTS WHEN DRAWING IN 2D

When you have drawn one geometry object on top of another object, toggling of the highlighting of these objects occurs when you click several times. It is important that you click without moving the cursor in a position where the objects overlap in order to toggle. For example, if you draw a circle (C1) and then draw another smaller circle (C2) inside of C1, then a first click on C2 may highlight C1, while a second click, without moving the cursor, highlights C2. In previous versions, you would move the cursor after the first click in order to highlight C2.

Backward Compatibility with Version 4.3

NEW TERMINOLOGY FOR CONSTRAINT TYPES

The following constraint types have new names in version 5.0:

- *Bidirectional, symmetric* is now *Apply reaction terms on: All physics (symmetric)*.
- *Unidirectional* is now *Apply reaction terms on: Individual dependent variables*.

WEAK CONSTRAINTS UPDATE FOR THE LAMINAR FLOW INTERFACE

The weak constraint formulations for the following boundary conditions has been updated:

- Symmetry
- The Slip option in the Wall feature

These boundary conditions are now formulated using the same set of Lagrange multipliers as all the other boundary conditions in the Laminar Flow interface. The Lagrange multiplier `un_lm` has been removed.

Models saved in version 4.3 will include `un_1m` until the model is re-solved. In some cases, occurrences of `un_1m` in the solver sequence must be replaced manually. This is the case if `un_1m` was the only Lagrange multiplier component in a segregated group or the only Lagrange multiplier component of a Vanka smoother. Alternatively, you can generate a new automatic solver sequence. Models saved in versions earlier than 4.3 must either be re-solved in version 5.0 for postprocessing, or opened and re-saved in version 4.3 before being opened in version 5.0.

AUTOMATICALLY CREATED IDENTITY/CONTACT PAIRS

The pairs created by the Form Assembly feature in the geometry can differ from their form in 4.3 for certain 3D and 2D geometries. The pairs should now be the same for the CAD and COMSOL representations.

AC/DC Module

New Functionality in Version 5.1

NEW COIL GEOMETRY ANALYSIS FUNCTIONALITY

The Coil Current Calculation functionality available in previous versions, used to compute the wire path in the 3D Multi-Turn Coil domain features, has been replaced with an improved **Coil Geometry Analysis** functionality. The new formulation is more robust and can handle more complex shapes and non-constant cross sections. The modeling procedure and the user interface is unchanged, except for the following improvements:

- Multiple Coil features can be solved for in the same study step,
- An **Electric Insulation** subnode is not required anymore.

The following entities have new names:

- The **Coil Current Calculation** study step is now called **Coil Geometry Analysis**
- The **Automatic Current Calculation** subnode is now called **Geometry Analysis**.

The **Coil Geometry Analysis** step is solved using a **Stationary** solver.

ACCURATE VOLTAGE CALCULATION FOR 3D MULTI-TURN COILS

3D Multi-Turn Coils in frequency studies now apply a “filtering stage” to the coil current density that significantly improves the accuracy of the computed electric field and, consequently, of the computed Coil voltage and other derived variables, such as power, inductance, etc. The current filtering stage is solved together with the main magnetic problem in the same study step, so it does not require any additional setup. The cost of this improvement is a slightly increased number of degrees of freedom. The functionality is active by default and can be turned off if desired.

LOSS TANGENT, DISSIPATION FACTOR

In the **Electric Current** interface, the old **Loss tangent** model is renamed to **Loss tangent, loss angle** and a new electric displacement field model is added: **Loss tangent, dissipation factor**. This model allows directly entering a value for the dissipation factor, $\tan(\delta)$.

COIL USABILITY IMPROVEMENTS

The setup of coil features in 3D has been simplified and the workflow streamlined. The **Settings** window for the **Multi-Turn Coil** feature has been reorganized to make

commonly used inputs more accessible, while some of the controls specific to the **User defined** coil type have been moved to a dedicated subfeature. The subnodes required to set up the coil are added automatically together with the coil feature.

New inputs have been added to simplify the setup of models with symmetry cuts. Circular coils can now also be used in sector-symmetric models.

GAUGE FIXING IMPROVEMENTS

The **Gauge Fixing** feature has been improved to perform better in complex models and require fewer user inputs.

- A new strategy for the boundary conditions for the divergence variable has been implemented.
- The feature can handle automatically antiperiodic models, models with multiple non-connected vector potential regions (for example, complex Rotating Machinery models), and models with mixed A-V and A-formulation.

SPICE EXPORT AND NEW FEATURES

The **SPICE Export** functionality is now available for the **Electrical Circuits** interface. The functionality is available from the contextual menu of the physics interface, and will produce a SPICE netlist file that describes the circuit created with the Electrical Circuits interface.

Four new circuit devices have been added: **PNP BJT** (bipolar junction transistor), **p-channel MOSFET**, **Mutual Inductance**, and **Transformer**.

UNIFORM BACKGROUND MAGNETIC FLUX DENSITY

When solving the **Magnetic Fields** interface using the reduced field formulation, the background field can now be specified as a uniform background magnetic flux density, in addition to the background magnetic vector potential option available in previous versions.

HARMONIC PERTURBATION SUBNODES FOR COILS ARE NOW GLOBAL

Harmonic Perturbation subnodes that can be added to coil features (**Single-Turn Coil**, **Multi-Turn Coil**) are now global features without a selection, since they apply a contribution to global variables only, such as the coil current or voltage.

NEW APPLICATIONS FOR THE AC/DC MODULE APPLICATION LIBRARY

- Demo Application: Transmission Line Calculator
- Modeling a Spiral Inductor Coil

COIL CURRENT CALCULATION

When models saved in previous versions are opened in version 5.1, the Coil Current Calculation study steps are migrated to Coil Geometry Analysis. Old solutions stored in the model can still be postprocessed, and the solver sequence will be regenerated automatically (with a Stationary solver) the first time the study is solved.

The Coil Geometry Analysis study step (formerly Coil Current Calculation) will now solve for all the coils in the active interfaces. To solve only for specific coils (specified using the `CoilName` property), set the `SpecifyCoil` property to 1.

The default solver sequence generated by the Coil Geometry Analysis is different in version 5.1. Code that accesses specific solver features in the generated solver sequence may need to be reviewed.

OTHER COIL IMPROVEMENTS

Harmonic Perturbation subnodes under coil features (Single-Turn Coil, Multi-Turn Coil) are now global features, so the call to the create method should use the appropriate space dimension (-1):

```
model.physics("mf").feature("std1").create("hp1",  
    "CoilHarmonicPerturbation", -1);
```

Global features have no selections, so code that access the selection of the Harmonic Perturbation features may need to be reviewed.

Some of the improvements in the 3D Multi-Turn Coil features may require a review of existing code that uses the COMSOL API.

Subnodes required to set up the coil features are now added automatically. Existing code that uses the coil features may need to be updated.

The parameters `eCoil` and `length` have been moved from the Multi-Turn Coil features (boundary and domain) to the new subfeature `UserDefinedCoilGeometry`.

OTHER IMPROVEMENTS

New functionality introduced in version 5.1 is disabled by default when opening models created in previous versions:

- New boundary conditions for Gauge Fixing features
- Accurate coil voltage calculation

This functionality can be enabled using the appropriate inputs in the Settings window. Refer to the documentation for the individual features for more details.

Acoustics Module

New Functionality in Version 5.1

IMPEDANCE BOUNDARY CONDITION IN PRESSURE ACOUSTICS, FREQUENCY DOMAIN

Several predefined impedance boundary conditions now exist in the Pressure Acoustics, Frequency Domain interface, each used to model a certain acoustic behavior at a boundary. This can be the losses associated with a porous layer, a simple mechanical system (approximated by a combination of loss, compliance, and mass), the behavior at the opening of a wave guide, or the acoustics of different parts of the human ear. More specifically, the ear impedance and skin impedance models provide tools for engineers to add realistic acoustic loads when developing and simulating headphones, hearing aids, headsets, and other mobile devices. The impedance boundary conditions are divided into several categories with different options: **User defined**, **RCL**, **Physiological**, **Waveguide end impedance**, **Porous layer**, and **Characteristic specific impedance**. Note that impedance conditions are, depending on the studied frequency, only approximations to the true behavior; their advantage is that they have very low computational cost and give a good first approximation to complex systems. Depending on the space dimension, the options for the **Impedance model** list are:

- **User defined**: enter a user defined expression of any kind.
- **RCL**: contains options for all possible combinations of an RCL circuit.
- **Physiological**: includes models for human skin and the human ear (ear drum, pinna, and full ear).
- **Waveguide end impedance**: flanged and unflanged pipe end impedance models.
- **Porous layer**: select a layer thickness and select porous model (same options as for the Poroacoustic feature).
- **Characteristic specific impedance** of plane, spherical, and cylindrical waves.

ADDITIONAL POROACOUSTICS MODELS

The list of Poroacoustic fluid models has been extended to include two equivalent density fluid models for modeling sediments and fluids with inclusions. The following new sets of predefined parameters have also been implemented for the Delany-Bazley-Miki model:

- **Wood** model for modeling fluids with inclusions such as particles.

- **Williams EDFM** model, an effective density fluid model used to model the propagation of acoustic waves in sediments.
- **Delany-Bazley-Miki** includes several new predefined empirical coefficients, including the Modified Allard and Champoux coefficients.

DIPOLE POINT SOURCES IN PRESSURE ACOUSTICS, FREQUENCY DOMAIN

Mathematically, a dipole is a source that corresponds to two monopoles that are close to each other and are completely out of phase. Dipoles appear when there are fluctuating forces in the medium, such as a small object that vibrates back and forth. A complex acoustic source can be expanded and approximated by a collection of point sources (monopoles, dipoles, and quadrupoles).

QUADRUPOLE POINT SOURCES IN PRESSURE ACOUSTICS, FREQUENCY DOMAIN

A quadrupole is, mathematically, a source that corresponds to two dipoles that are close to each other. A complex acoustic source may be expanded and approximated by a collection of point sources (monopoles, dipoles, and quadrupoles).

INTERIOR VELOCITY BOUNDARY CONDITION IN THERMOACOUSTICS

This condition is used to specify a velocity on an interior boundary in thermoacoustics. The condition can be used to specify sources, for example, the velocity of a diaphragm in a miniature transducer that is modeled using a lumped circuit model. The velocity components can be prescribed independently, and there is an option to force continuity in pressure across the boundary, as well as options for thermal conditions.

DATA SETS THAT SIMPLIFY EVALUATING AND PLOTTING THE FAR FIELD

The Parameterized Curve and Parameterized Surface data sets now support evaluation where there is no domain mesh. This is achieved by selecting the **Only evaluate globally defined expressions** check box. In this way the far-field variables can be evaluated outside the mesh on a predefined parameterized surface or curve. The new Grid data sets can be used to plot the far-field solution outside the computational domain, in volumes or surfaces.

RAY ACOUSTICS IMPROVEMENTS

Intensity Calculations in Graded Media (Using the Curvature Tensor)

The intensity calculation is now supported for graded media (that is, media where the speed of sound is space dependent). An example is the case of ocean acoustics, where the speed of sound typically depends not only on depth but on temperature and salinity

as well. The intensity calculation is now based on a curvature tensor rather than the principal curvatures. In the settings under the **Ray Properties** at the physics level, select the option **Using curvature tensor** under **Intensity computation**.

Fluid Models with Attenuation

The medium properties now have two fluid model options for modeling the attenuation of the acoustic waves due to bulk losses. Attenuation becomes important in air at high frequencies and in large spaces, like concert halls; it is also important in underwater acoustics applications. The **Linear elastic with attenuation** option allows a user-defined expression to be used for the attenuation coefficient, and the **Thermally conducting and viscous** option sets up the classical attenuation expression due to viscosity and thermal conduction.

Improved Support for Ray Acoustics with Frequency-Dependent Material Properties

In ray acoustics models, it is now possible to specify material properties that are dependent on the ray frequency or another ray property directly in the **Material** node's **Settings** window instead of the **Medium Properties** node's **Settings** window. To do so, all ray properties must be contained within the new `noenv()` operator, which allows quantities that exist only on rays to be included in expressions defined on domains.

OTHER IMPROVEMENTS

- Improved performance for domain-level **Accumulator** nodes: The variable computed by the domain-level **Accumulator** is now faster and more accurate than in version 5.0. Some models with domain Accumulators may be 10 or more times faster. These models no longer require manual changes to the solver sequence.
- There is a new release type **Release from Data File**. You can now import the initial positions and directions of rays from a text file.
- There is a new option for **Release from Grid** to set the **Grid type** to **All combinations** or **Specified combinations**. This gives better control over the initial placement of rays.
- New modeling chapters have been added to the *Acoustics Module User's Guide*. These chapters contain information about modeling, tips and tricks, and good practices, for example, about meshing, solvers, and more.

HELMHOLTZ RESONATOR ANALYZED WITH DIFFERENT FREQUENCY DOMAIN SOLVERS

This model simulates a frequency sweep of a generic Helmholtz resonator—a classic acoustics resonating circuit with a known theoretical solution—to illustrate how to use different solvers in the frequency domain. In addition to the stationary solver, the model uses the Asymptotic Waveform Evaluation solver and the stationary Frequency Domain-Modal solvers, which both reconstruct the result based on expansions around a few exact solutions in the sweep range.

PIEZOELECTRIC TONPILZ TRANSDUCER WITH A PRESTRESSED BOLT

- This application shows how to model prestressed acoustic-structure interaction using the perturbation solver.
- A tonpiliz transducer is used for relatively low frequency, high power sound emission. It is one of the popular transducer configurations for SONAR applications. The transducer consists of piezoceramic rings stacked between a head mass and a tail mass which are connected by a central bolt. This model shows how to incorporate the effect of a pretension in the bolt. The bolt geometry is imported from the Part Libraries. The frequency response of the transducer is studied to determine structural and acoustic response of the device, including deformation, stress, radiated power, sound pressure level, the transmitting voltage response (TVR) curve, and the directivity index (DI) of the sound beam. This model requires the Acoustics Module and the Structural Mechanics Module.

SMALL CONCERT HALL ANALYZER

- This is a runnable application available from the Demo Applications folder.
- This application analyzes the acoustics of a small concert hall using the Ray Acoustics interface. The application lets you define a microphone location to measure the impulse response, an omnidirectional sound source, wall absorption parameters, and the properties of diffusers. The results include a filtered energy impulse response for a given Fourier component.

Backward Compatibility with Version 4.3a and Older

- The old perfectly matched layer (PML) node that is located under the physics node will be discontinued in the next version of COMSOL. From the next version on only the PMLs defined under the Definitions node are supported.
- In order for the old perfectly matched layer (PML) to be backward compatible, the PML feature must be placed after any domain Monopole Source, domain Dipole Source, or Background Pressure Field features.

Backward Compatibility with Version 4.3

- The symbols for the acoustics monopole (Q_m ; was Q) and dipole (\mathbf{q}_d ; was \mathbf{q}) volume sources in pressure acoustics have changed.
- The default value for the **Typical wave speed** property in the Acoustic-Piezoelectric Interaction, Frequency Domain interface has changed to 343 m/s.

Batteries & Fuel Cells Module

New Functionality in Version 5.1

IMPROVED CURRENT DISTRIBUTION INITIALIZATION STEP AND NEW STUDIES

- The **Current Distribution Initialization** study step, when used together with an Electrochemistry interface, solves for a stationary solution for the potential variables only (phi_s, phi_l, and global potential variables set up by boundary features). The Current Distribution type during initialization can be set to either Primary (which is the default) or Secondary. The use of a Primary Current Distribution Initialization step eliminates the need to manually derive and provide Initial Values in the Electrochemistry interfaces for most models. However, a Secondary Current Distribution may be needed in problems where the ohmic drop in the electrolyte is negligible, such as in thin wafer deposition with lateral electronic conduction, cathodic protection, or mixed potential problems. The study step is supported by all Electrochemistry interfaces.
- The **Stationary with Initialization** study is available for the Secondary and Tertiary Current Distribution interfaces. The study adds one (Primary) Current Distribution Initialization study step, followed by a Stationary study step.
- The **Time Dependent with Initialization** study is available for the Secondary and Tertiary Current Distribution and Battery interfaces. The study adds one (Primary) Current Distribution Initialization study step, followed by a Time Dependent study step.

IMPROVED REACTIONS NODE IN TERTIARY CURRENT DISTRIBUTION, NERNST-PLANCK

The Reactions feature in the Tertiary Current Distribution, Nernst-Planck interface can now use reaction rates that are generated by the Chemistry interface (only available with a Chemical Reaction Engineering Module license). It also provides an option to account for the reacting volume from selections: **Total volume** and **Pore volume**.

DANCKWERTS INFLOW CONDITION IN TERTIARY CURRENT DISTRIBUTION, NERNST-PLANCK

The Inflow feature of the Tertiary Current Distribution, Nernst-Planck interface now provides an option for Danckwerts flux.

CROSS SECTIONAL AREA IN 1D

A new property, **Cross Sectional Area**, is now available in the 1D space dimension for the Electrochemistry interfaces. With this property, the boundary features Electrolyte Current and Electrode Current are also made available for the 1D space dimension.

PARTICLE-BASED SURFACE AREA IN POROUS ELECTRODES

The Porous Electrode Reaction subnode in the Secondary and Tertiary Current Distribution interfaces and in the Lithium-Ion and Battery with Binary Electrolyte interfaces now provides an option to calculate the surface area based on particle type and radius.

The above option to calculate the particle-based surface area is also available in the Porous Matrix Double Layer Capacitance subnode.

In line with these changes, the default behavior for the surface area expression in the Lithium-Ion and Battery with Binary Electrolyte interfaces has changed. The change has impact on Java API backward compatibility because the surface area has to be set to **User defined** for a user-defined surface area parameter to become active.

In case of Intercalating particles selection at the Porous Electrode node in the Lithium-Ion and Battery with Binary Electrolyte interfaces, the particle type and radius is read from the parent Porous Electrode node.

POINT AND LINE CURRENT SOURCES IN PRIMARY AND SECONDARY CURRENT DISTRIBUTION INTERFACES

With the Point and Line Current Source features, it is possible to apply a current source at any position within 2D, 2D axisymmetric, and 3D geometries. These are available both for electrolytes and solids.

INITIAL CELL CHARGE DISTRIBUTION

This new feature enables the user to directly set inputs such as cell voltage and state-of-charge in the battery interfaces and completely eliminates the setting of any other initial values. That is, the potential and concentration in the electrolyte and solid phases are now computed.

The feature has a global parent node and two domain subnodes. In the parent node, the initial voltage or state-of-charge of the battery cell are set together with some electrode balancing parameters due to irreversible losses. In the subnodes, the domains representing the positive and negative electrodes are selected.

This feature is available for all studies combined with a Current Distribution Initialization study step in Lithium-Ion Battery and Battery with Binary Electrolyte interfaces.

ADDITIONAL POROUS ELECTRODE MATERIAL

The feature previously named Additional Intercalating Material in the Lithium-Ion Battery and Battery with Binary Electrolyte interfaces has been renamed Additional Porous Electrode Material. It now also enables modeling not only additional intercalating materials but also non-intercalating materials, with or without any electrochemical reaction.

NEW MATERIAL PROPERTIES AND VARIABLES IN POROUS ELECTRODES

Porous electrodes in the Lithium-Ion and Battery with Binary Electrolyte interfaces now contain the option to import the operational state-of-charge window, `socmin` and `socmax`, for intercalating electrode materials. The new functionality is used to compute the local and averaged electrode material state-of-charge of the porous electrode domain.

INTERCALATION IN POROUS ELECTRODES OVER A NORMALIZED EXTRA DIMENSION DOMAIN

In the Porous Electrode domain feature for the Lithium-Ion and Battery with Binary Electrolyte interfaces, particle intercalation is now implemented by solving the governing solid diffusion equation over a normalized extra dimension domain. This implementation allows for modeling battery scenarios where the active material particle size changes. The change may have impact with regards to backward compatibility when postprocessing in the extra dimension because the extra dimension domain is now located between 0 and 1.

AC IMPEDANCE, INITIAL VALUES STUDY TYPE

This new study type adds a Frequency Domain, Linearized study step.

IMPROVED ASSEMBLY OF EXTRA DIMENSIONS IN POROUS ELECTRODES

The assembly time of the extra dimension equations used for the intercalating concentrations in the porous electrodes of the Lithium-Ion Battery and Battery with Binary Electrolyte interfaces has been significantly improved from the 5.0 version.

IMPROVED ELECTROLYTE MASS CONSERVATION IN THE LITHIUM-ION BATTERY AND BATTERY WITH BINARY ELECTROLYTE INTERFACES

The migrative boundary flux definition has been improved for various boundary nodes, which improves the electrolyte salt conservation in models with coarse meshes.

POROSITY TIME DERIVATIVE INCLUDED IN ELECTROLYTE SALT EQUATION

The time derivative is now taken from the product of the porosity and the salt concentration in the Lithium-ion Battery and Battery with Binary Electrolyte interfaces. Models made prior to 5.1 with dynamically changing porosities will render different results.

New Applications in Version 5.1

ID ISOTHERMAL ZINC-SILVER OXIDE BATTERY

This example uses the Battery with Binary Electrolyte interface for studying the discharge of a zinc-silver oxide battery. The electrochemical reactions in the positive and negative electrodes lead to changes in porosity and species concentration in the electrodes.

ID ISOTHERMAL LITHIUM-AIR BATTERY

This example uses the Lithium-Ion Battery interface for studying the discharge of a lithium-air battery. The transport of oxygen in the porous positive electrode is modeled using the Transport of Diluted Species in Porous Media interface. The electrochemical reaction of oxygen reduction in the positive electrode leads to changes in the concentration of the reaction product and electrode porosity.

Backward Compatibility with Previous Versions

TRANSPORT OF CONCENTRATED SPECIES INTERFACE

See [Chemical Reaction Engineering Module](#).

SURFACE REACTIONS INTERFACE

See [Chemical Reaction Engineering Module](#).

CAD Import Module, Design Module, and LiveLink Products for CAD

New Functionality in the CAD Import Module, Design Module, and LiveLink Products for CAD in Version 5.1

The updated Parasolid® geometric kernel and CAD file translation libraries that support the import of the following new CAD format versions:

- Parasolid® R27
- PTC® Creo® Parametric™ 3.0

New Functionality in LiveLink™ for Inventor®

- Object types, such as solids and surfaces, can now be excluded from the synchronization with the bidirectional interface. While it is always possible to prevent objects being synchronized by hiding them in the CAD design, this new functionality provides a faster way to exclude selected object types using just one click. Doing so can also result in faster synchronization for large CAD files that contain many geometric objects that do not need to take part in the simulation of, for example, construction surfaces.
- The loading time of the add-in for the bidirectional and One Window interfaces has been optimized so that their contribution to the start-up time of Inventor is kept to a minimum.

New Functionality in LiveLink™ for PTC® Creo® Parametric™

- The bidirectional interface now synchronizes selections for materials. Especially useful when analyzing assemblies of many components, this feature detects which materials are applied to the components of the CAD design, and uses this information to create selections in the COMSOL application. These selection are available when applying model settings, for example when assigning materials.
- Object types, such as solids and surfaces, can now be excluded from the synchronization with the bidirectional interface. While it is always possible to prevent objects being synchronized by hiding them in the CAD design, this new functionality provides a faster way to exclude selected object types by just one click.

Doing so can also result in faster synchronization for large CAD files that contain many geometric objects that do not need to take part in the simulation of, for example, construction surfaces.

New Functionality in LiveLink™ for PTC® Pro/ENGINEER®

- The bidirectional interface now synchronizes selections for materials. Especially useful when analyzing assemblies of many components, this feature detects which materials are applied to the components of the CAD design, and uses this information to create selections in the COMSOL application. These selection are available when applying model settings, for example when assigning materials.
- Object types, such as solids and surfaces, can now be excluded from the synchronization with the bidirectional interface. While it is always possible to prevent objects being synchronized by hiding them in the CAD design, this new functionality provides a faster way to exclude selected object types by just one click. Doing so can also result in faster synchronization for large CAD files that contain many geometric objects that do not need to take part in the simulation of, for example, construction surfaces.

New Functionality in LiveLink™ for Revit®

It is now possible to synchronize conceptual masses between Revit Architecture and COMSOL. This functionality enables multiphysics simulations at the early stages of the architectural design process to validate and explore architectural ideas.

Mass parameters are now also synchronized, which allows for the optimization of conceptual shapes and forms in Revit from the COMSOL application. Parameter values in Revit can be updated from the COMSOL application either manually or automatically, for example, by an optimization solver. The geometry in Revit is regenerated based on the updated parameter values and synchronized with COMSOL for updated simulation.

New Functionality in LiveLink™ for Solid Edge®

The bidirectional interface now synchronizes selections for materials. Especially useful when analyzing assemblies of many components, this feature detects which materials are applied to the components of the CAD design, and uses this information to create selections in the COMSOL application. These selection are available when applying model settings, for example, when assigning materials.

BACKWARD COMPATIBILITY WITH VERSION 5.0

The default value for the `keepfree` property of the `LiveLinkSolidEdge` function is now set to `on`. Previously the default was set to `off`.

New Functionality in LiveLink™ for SOLIDWORKS®

- Object types, such as solids, surfaces or curves and points, can now be excluded from the synchronization with the bidirectional interface. While it is always possible to prevent objects being synchronized by hiding them in the CAD design, this new functionality provides a faster way to exclude selected object types using just one click. Doing so can also result in faster synchronization for large CAD files that contain many geometric objects that do not need to take part in the simulation of, for example, construction surfaces.
- The loading time of the add-in for the bidirectional and One Window interfaces has been optimized so that their contribution to the start-up time of SOLIDWORKS is kept to a minimum.

BACKWARD COMPATIBILITY WITH VERSION 5.0

The default value for the `keepfree` property of the `LiveLinkSOLIDWORKS` function is now set to `on`. Previously the default was set to `off`.

CFD Module

New Functionality in Version 5.1

INCOMPRESSIBLE AS DEFAULT SETTING IN SINGLE-PHASE FLOW

The default setting for the **Compressibility** property (at the physics interface level) has been changed from **Compressible flow (Ma<0.3)** to **Incompressible flow**. This is to avoid unnecessary restrictions on the time step. In compressible flow simulations employing a material for which the density depends on the pressure, discontinuities in the initial conditions trigger pressure waves, even for small Mach numbers, which propagate through the domain at the speed of sound. These pressure waves survive during a short transient which must be temporally resolved, and this gives a restriction on the time step. The transient wave regime is most often an artifact of an unknown initial state.

The new default setting applies to all interfaces with **Name: spf**. Weakly coupled multiphysics interfaces such as the **nitf** interfaces inherit the default setting.

Models built in versions prior to 5.1 retain the old default setting (**Compressible flow (Ma<0.3)**).

POROUS DOMAIN FEATURE IN SINGLE-PHASE FLOW

Under **Physical Model**, at the physics interface level, there is a new check box for **Enable porous media domains**. When selected, the new domain feature **Fluid and Matrix Properties** is added to the physics interface. The feature may have all of the single-phase flow domains or a subset thereof as its selection. On the selected domain(s), the Brinkman equations are solved instead of the Navier-Stokes equations. The velocity and pressure are continuous across boundaries between free and porous domains. The **Neglect inertial term (Stokes flow)** check box under **Physical Model** (at the physics interface level) also controls the corresponding terms in the Brinkman equations. A Forchheimer drag correction is available as a subnode to the **Fluid and Matrix Properties** node for cases with high interstitial velocities.

The new domain feature is compatible with the algebraic turbulence models (**Algebraic yPlus** and **L-VEL**) enabling studies of coupled turbulent and porous media flow.

IMPROVED INLET BOUNDARY CONDITION IN BUBBLY FLOW

The **Inlet** feature in the Bubbly Flow interfaces has been updated with a new **Pressure** condition. The new **Pressure** condition sets the normal stress on the boundary, which

is numerically more robust than the previous **Pressure/No viscous stress** condition. It includes an option to **Suppress backflow** and the choice between **Normal flow** or **User defined Flow direction**.

Models built in versions prior to 5.1 retain the old **Inlet** feature, but you can replace it by the new feature by adding a new **Inlet** node from the context menu or the Physics toolbar.

IMPROVED OUTLET BOUNDARY CONDITION IN BUBBLY FLOW

The **Outlet** feature in the Bubbly Flow interfaces has been updated with a new **Pressure** condition including **Normal flow** and **Suppress backflow** options. The new **Pressure** condition sets the normal stress on the boundary, which is numerically more robust than the previous **Pressure/No viscous stress** condition. In some cases (even when **Suppress backflow** is selected), it is not possible to avoid backflow on the entire boundary. For this reason, the **Outlet** feature has been provided with an **Exterior gas conditions** section including an input field for the **Effective gas density** and options for specifying the number density when solved for.

Models built in versions prior to 5.1 retain the old **Outlet** feature, but you can replace it by the new feature by adding a new **Outlet** node from the context menu or the Physics toolbar.

IMPROVED INLET BOUNDARY CONDITION IN THE MIXTURE MODEL

For the **Inlet** feature in the Mixture Model, when the **Mixture Boundary Condition** is set to **Velocity**, there is now a **Normal inflow** velocity option in addition to the **Velocity field** option.

The **Inlet** feature in the Mixture Model interface has been updated with a new **Pressure** condition including a **Suppress backflow** option and the choice between **Normal flow** or **User defined Flow direction**. The new **Pressure** condition sets the normal stress on the boundary, which is numerically more robust than the previous **Pressure/No viscous stress** condition.

Models built in versions prior to 5.1 retain the old **Inlet** feature, but you can replaced it by the new feature by adding a new **Inlet** node from the context menu or the Physics toolbar.

IMPROVED OUTLET FEATURE IN THE MIXTURE MODEL

For the **Outlet** feature in the Mixture Model, when the **Mixture Boundary Condition** is set to **Velocity**, there is now a **Normal outflow** velocity option in addition to the **Velocity field** option.

The **Outlet** feature in the Mixture Model interface has been updated with a new **Pressure** condition including **Normal flow** and **Suppress backflow** options. The new **Pressure** condition sets the normal stress on the boundary, which is numerically more robust than the previous **Pressure/No viscous stress** condition. In some cases (even when **Suppress backflow** is selected), it is not possible to avoid backflow on the entire boundary. For this reason, the **Outlet** feature has been provided with an **Exterior dispersed phase conditions** section including an input field for the **Dispersed phase volume fraction** and options for specifying the number density when solved for.

Models built in versions prior to 5.1 retain the old **Outlet** feature, but you can replace it by the new feature by adding a new **Outlet** node from the context menu or the Physics toolbar.

NEW PHYSICS INTERFACE: EULER-EULER MODEL, TURBULENT FLOW

A new interface for high Reynolds number dispersed two-phase flow has been introduced. Turbulence is modeled using the standard k - ϵ turbulence model including realizability constraints. The interface includes the option to solve one set of k - ϵ equations for the mixture (**Two-Phase Turbulence** is set to **Mixture**) or to solve two sets of k - ϵ equations, one for each phase (**Two-Phase Turbulence** is set to **Phase specific**). By default the former setting is applied.

PSEUDO TIME STEPPING FOR THE EULER-EULER MODEL INTERFACES

The Euler-Euler Model interfaces now support pseudo time stepping. This makes it easier to solve stationary models, especially for turbulent flow. The settings are found in the **Advanced Settings** section at the interface level. Pseudo-time stepping is enabled by default for both laminar and turbulent flow.

Pseudo time stepping is disabled for models built in versions prior to 5.1.

IMPROVED FORMULATION FOR THE DISPERSED PHASE VISCOUS STRESS IN THE EULER-EULER MODEL INTERFACES

The viscous stress in the momentum equation for the dispersed phase has been multiplied by the dispersed phase volume fraction. This formulation is more general than the previous one, which mainly applies to granular flows.

Results may differ slightly for models built in versions prior to 5.1.

ACCURATE BOUNDARY FLUXES IN THE DARCY'S LAW INTERFACE

The Darcy's Law interface now includes accurate boundary fluxes, which can be activated from the **Compute boundary fluxes** check box under the **Discretization** section at the physics interface level.

CAPILLARY PRESSURE IN THE TWO-PHASE FLOW DARCY'S LAW INTERFACE

Capillary pressure is now available as an option for the **Capillary Model** in the Two-Phase Darcy's Law interface. The available capillary pressure models are **van Genuchten**, **Brooks and Corey**, and **User defined**.

PERFORATIONS FEATURE IN THIN FILM FLOW

A new perforations feature is available for thin film damping, enabling the modeling of thin film flow in structures with etch holes. The perforations feature acts as a sink for gas that is proportional to both the ambient pressure and to the pressure difference with respect to the ambient pressure on the other side of the perforated surface. The constant of proportionality is known as the **Perforation admittance** and can either be defined directly, or determined from a model due to Bao.

IMPROVED BORDER FLOW FEATURE IN THIN FILM FLOW

A new option is available for the **Border flow** boundary condition for thin film flow. Selecting **Out-of-plane motion** for the **Border flow type** calculates the pressure gradient at the boundary using a model of Gallis and Torczynski. This model has been shown to agree well with detailed CFD and Monte Carlo simulations that model both the thin film flow domain and the surrounding gas. The model applies for both rarefied and nonrarefied flows up to Knudsen numbers of approximately one.

NEW FUNCTIONALITIES IN THE HIGH MACH NUMBER FLOW INTERFACES

New functionalities are now available in the high Mach number flow interfaces, including **Surface-to-surface radiation** with **Wavelength dependence of emissivity** (requires an additional Heat Transfer Module license) and **Thin Layer**, which is now compatible with turbulent wall functions for temperature.

INTELLIGENT RENAMING OF THE SINGLE-PHASE FLOW INTERFACES

The label of a single-phase flow interface (node name in the **Model Builder**) is now dynamic, and it is reset according to the turbulence model and the **Neglect inertial term (Stokes flow)** property.

Suppose a Laminar Flow interface has been added to the component. If the SST turbulence model is now selected, the interface **Label** changes to **Turbulent Flow, SST**, the same **Label** that would have resulted if the corresponding interface had been added directly from the **Model Wizard**.

Enabling **Neglect inertial term (Stokes flow)**, the **Label** is changed to Creeping Flow, also in accordance with the corresponding entry in the **Model Wizard**.

The renaming is based on the following logic:

- A default **Label** will be changed to a new default **Label**.
- If the **Label** has been manually changed, such as by adding a prefix or suffix to the default string, the new **Label** will retain these.
 - **My Laminar Flow Interface** is changed to **My Turbulent Flow, k-ε Interface**.
 - Note that the default string depends on the language used in the GUI (default string localized).
- If the **Label** has been replaced so that it does not include the default **Label**, it will not be affected by any property changes.

Backward Compatibility with Version 5.0

EULER-EULER MODEL, LAMINAR FLOW

The results from models using the Euler-Euler Model, Laminar Flow interface may differ in version 5.1 due to the more general formulation for the viscous stress in the momentum equation for the dispersed phase. The behavior in the old formulation may be reproduced by dividing the dispersed phase viscosity, μ_D , by the variable `ee.phidPos`.

TRANSPORT OF CONCENTRATED SPECIES

See [Chemical Reaction Engineering Module](#).

Backward Compatibility with Version 4.4

INLET FEATURE

The old inlet features still exist, but have been excluded from the physics context menu. Old models retain old Inlet features, but adding a new Inlet feature comes with the new functionality. Old Java files create Inlet features of the old types. The Normal

stress condition can still be prescribed on inlets using an Open Boundary or a Boundary Stress feature.

NEW NAMES FOR MASS TRANSPORT INTERFACES

Running Model Java-files from previous versions may fail due to the new default Name in the Transport of Diluted Species and Transport of Concentrated Species interfaces. This can be avoided by adding a command that specifies the identifier in accordance with the Name of the interface. For example, when creating a Transport of Concentrated Species interface using the old Name (chcs):

```
model.physics().create("chcs", "ConcentratedSpecies", "geom1",
                        new String[][]{{"wO2", "wH2O", "wN2"}});
```

Add the following line to specify the identifier accordingly:

```
model.physics("chcs").identifier("chcs");
```

WALL DISTANCE EQUATION

The old Continuity node is still included in the interface but is excluded from the feature list. Hence, old models retain their old Continuity node when opened in version 5.0 and old Model Java-files are fully backward compatible.

EULER-EULER MODEL, LAMINAR FLOW

The results from models using the Euler-Euler Model, Laminar Flow interface may differ in version 5.1 due to the more general formulation for the viscous stress in the momentum equation for the dispersed phase. The behavior in the old formulation may be reproduced by dividing the dispersed phase viscosity, μ_D , by the variable `ee.phidPos`.

Backward Compatibility with Version 4.3b

BUBBLE-INDUCED TURBULENCE IN BUBBLY FLOW

Equation terms accounting for bubble-induced turbulence in the Turbulent Bubbly Flow interface are now added correctly. As a result, models with Turbulent Bubbly Flow now show higher levels of turbulence.

EULER-EULER MODEL, LAMINAR FLOW

The results from models using the Euler-Euler Model, Laminar Flow interface may differ in version 5.1 due to the more general formulation for the viscous stress in the momentum equation for the dispersed phase. The behavior in the old formulation may

be reproduced by dividing the dispersed phase viscosity, μ_D , by the variable `ee.phidPos`.

Backward Compatibility with Version 4.3a

MIXTURE MODEL EQUATION FORMULATION

The Mixture Model equation formulation has been revised to increase the accuracy and facilitate the solution of a number of problem setups. As a consequence of the revision, old models can return slightly different results in the new version than they did in the previous versions. As part of the revision, a penalty diffusion on the dispersed phase has been deactivated, because after the other revisions, it made the equation system too nonlinear and therefore more difficult to make it converge. However, you can still activate the penalty diffusion in the Advanced section of the Mixture Model settings.

EULER-EULER MODEL, LAMINAR FLOW

The results from models using the Euler-Euler Model, Laminar Flow interface may differ in version 5.1 due to the more general formulation for the viscous stress in the momentum equation for the dispersed phase. The behavior in the old formulation may be reproduced by dividing the dispersed phase viscosity, μ_D , by the variable `ee.phidPos`.

Backward Compatibility with Version 4.3

FLUID-STRUCTURE INTERACTION

The Fluid-Structure Interaction (FSI) multiphysics interface has been updated. The separate `vWall` field is no longer required and has been removed. FSI models from 4.3 and earlier versions that include Time Dependent study steps will be affected in the following way:

- Model files for Java will fail. Any reference to the `vWall` field must be removed.

- Models older than 4.3 (that is, 4.2a and older) must either clear their time-dependent solutions when opened in 5.0. An alternative procedure is to:
 - Open the model in version 4.3.
 - Right-click any study with a time-dependent study step and select *Update Solution*.
 - Save the model.
 - Open the model in version 5.0.

FSI models with only stationary study steps will not be affected. Note that `vWall` will still be available as a variable. Hence, references to `fsi.vWall` made in, for example, another physics still work.

BRINKMAN EQUATIONS AND FREE AND POROUS MEDIA FLOW

The momentum equations in the Brinkman Equations interface and the Free and Porous Media Flow interface have been corrected. The term $-Q_{br} \cdot \mathbf{u} / \epsilon_p^2$ in the right hand side previously lacked the factor $1/\epsilon_p^2$, where ϵ_p is the porosity.

REACTING FLOW INTERFACE NAME CHANGE

The Reacting Flow interfaces have been renamed Reacting Flow in Porous Media. If you open a model using either the Reacting Flow, Diluted Species or the Reacting Flow, Concentrated Species interface, the interface is renamed Reacting Flow in Porous Media.

WEAK CONSTRAINTS UPDATE FOR FLUID FLOW INTERFACES

The weak constraint formulations for the following boundary conditions in the following interfaces has been updated:

Laminar Flow and Creeping Flow

- Symmetry
- The Slip boundary condition in the Wall feature

Turbulent Flow, k-ε and Turbulent Flow, k-ω

- Symmetry
- The Wall functions boundary condition in the Wall feature
- The Moving wall (wall functions) boundary condition in the Wall feature
- The Slip boundary condition in the Wall feature

Turbulent Flow, low-Re k - ϵ and Turbulent Flow, Spalart-Allmaras

- Symmetry
- The Slip option in the Wall feature

Rotating Machinery

- Symmetry
- The Slip boundary condition in the Wall feature
- The Wall functions boundary condition in the Wall feature (turbulent flow only)
- The Moving wall (wall functions) boundary condition in the Wall feature (turbulent flow only)
- Rotating Wall (turbulent flow only)

Bubbly Flow

- Symmetry
- The Slip boundary condition for the liquid phase in the Wall feature
- The Wall functions boundary condition for the liquid phase in the Wall feature (turbulent flow only)

Mixture Model

- Symmetry
- The Slip boundary condition for the mixture in the Wall feature
- Wall functions boundary condition for the mixture in the Wall feature (turbulent flow only)

Euler-Euler Model, Laminar Flow

- Symmetry
- The Slip boundary condition for either phase in the Wall feature
- The Slip boundary condition for either phase in the Inlet feature

Brinkman Equations and Free and Porous Media Flow

- Symmetry
- The Slip boundary condition in the Wall feature

Two-Phase Flow, Level Set

- Symmetry
- The Slip boundary condition in the Wall feature
- The Wetted wall boundary condition in the Wall feature
- The Moving wetted wall boundary condition in the Wall feature

- The Wall function boundary condition in the Wall feature (turbulent flow only)
- The Moving wall (wall functions) boundary condition in the Wall feature (turbulent flow only)

Two-Phase Flow, Phase Field

- Symmetry
- The Slip boundary condition in the Wall feature
- The Wall function boundary condition in the Wall feature (turbulent flow only)
- The Moving wall (wall functions) boundary condition in the Wall feature (turbulent flow only)

Non-Isothermal Flow and Conjugate Heat Transfer

- Symmetry
- The Slip boundary condition in the Wall feature
- The Wall functions boundary condition in the Wall feature (turbulent flow k - ϵ and turbulent flow k - ω only)
- The Moving wall (wall functions) boundary condition in the Wall feature (turbulent flow k - ϵ and turbulent flow k - ω only)

High Mach Number Flow

- Symmetry
- The Slip boundary condition in the Wall feature
- The Wall functions boundary condition in the Wall feature (turbulent flow k - ϵ only)
- The Moving wall (wall functions) boundary condition in the Wall feature (turbulent flow k - ϵ only)

These boundary conditions are now formulated using the same set of Lagrange multipliers as all the other boundary conditions for the dependent velocity variables. The previously used Lagrange multiplier `un_1m` has been removed.

When you open models saved in version 4.3, they include `un_1m` until the model is re-solved. In some cases, occurrences of `un_1m` in the solver sequence must be replaced manually. This is the case if `un_1m` was the only Lagrange multiplier component in a segregated group or the only Lagrange multiplier component of a Vanka smoother. Alternatively, you can generate a new automatic solver sequence. Models saved in versions prior to version 4.3 must either be re-solved in version 5.0 for postprocessing, or opened and re-saved in version 4.3 before being opened in version 5.0.

Weak constraints for the Interior Wall feature are no longer available.

REVISION OF THE TURBULENCE MODELS

The formulations of some variables in the turbulence models have been revised in order to improve accuracy. Models using a turbulence model can display a different convergence behavior in version 5.0 than in version 4.3 and the result can differ slightly between the versions.

EULER-EULER MODEL, LAMINAR FLOW

The results from models using the Euler-Euler Model, Laminar Flow interface may differ in version 5.1 due to the more general formulation for the viscous stress in the momentum equation for the dispersed phase. The behavior in the old formulation may be reproduced by dividing the dispersed phase viscosity, μ_D , by the variable `ee.phidPos`.

Chemical Reaction Engineering Module

New Functionality in Version 5.1

SURFACE CHEMKIN IN CHEMISTRY AND REACTION ENGINEERING INTERFACES

CHEMKIN import of files that contain physical properties of chemical species and chemical reactions kinetics in reacting volumes have been available for a long time. Now it is also possible to import Surface CHEMKIN files with surface species and surface reaction data. This applies for all types of CHEMKIN files: reaction kinetics, transport properties, and thermodynamics. The Surface CHEMKIN standard is particularly useful when modeling, for example, catalytic reactions and you have reaction data available in the Surface CHEMKIN format. The GaAs CVD application (<http://www.comsol.com/model/chemical-vapor-deposition-of-gaas-945>) has been updated to use this new functionality.

NEW GAS MIXTURE VISCOSITY CORRELATION

A dynamic viscosity prediction method is now available for concentrated gas mixtures in the Reaction Engineering and Chemistry interfaces.

FILM RESISTANCE CAPABILITY ADDED TO REACTIVE PELLET BED

The Reactive Pellet Bed feature now has two alternatives to couple the macroscale concentration to microscale concentration at the pellet-fluid surface:

- Continuous concentration
- Film resistance (mass flux) (new)

The new Film resistance condition relates the mass transfer to/from the pellet to a film coefficient. This is a common model when studying bioreactors and catalytic beds. Here, resistance to mass transfer is also assumed in a thin film near the pellet surface in addition to inside the porous pellet. The mass transfer coefficients are automatically calculated from the Sherwood number, which in turn are defined from either of these three fluid-dynamic empirical expressions:

- Frössling

- Rosner
- Garner and Kee

IMPROVED REACTIONS IN POROUS MEDIA

The Reactions feature in the Transport of Diluted Species in Porous Media interfaces now provides the following options to account for the reacting volume base for saturated and unsaturated porous media:

- Total volume
- Pore volume
- Liquid phase
- Gas phase

This makes it simpler and less error prone to use literature data for kinetic expressions that are either based on total volume or pore space. It also facilitates the coupling between interfaces that operates in different parts within a porous media domain.

HYGROSCOPIC SWELLING

Hygroscopic swelling is an effect of internal material strain caused by changes in moisture content.

The new Hygroscopic Swelling multiphysics coupling is used for moisture concentration coupling between the Transport of Diluted Species, or Transport of Diluted Species in Porous Media interfaces, and the Solid Mechanics interface.

DUSTY GAS MODEL CAPABILITY ADDED TO THE TRANSPORT OF CONCENTRATED SPECIES INTERFACE

Knudsen diffusion is now available as an additional transport mechanism in the Transport of Concentrated Species interface to be enabled in Dusty Gas models. The mechanism is available for the Fick's law and Mixture-averaged diffusion models.

The Dusty Gas model is sometimes preferred to accurately predict mass transport accompanied by chemical reactions in porous media, for example catalytic membranes and fuel cell applications.

In gases, this mechanism is important for the transport rate if the mean free path of transported molecules is on the same order of magnitude or larger than the length scale of the system. For example, in a long pore with a narrow diameter (2–50 nm) the molecules frequently collide with the pore wall and the diffusion needs to be adjusted accordingly. It is a necessary feature for setting up Dusty Gas Models.

MASS-BASED CONCENTRATION VARIABLES ADDED TO THE TRANSPORT OF CONCENTRATED SPECIES INTERFACE

The Transport of Concentrated Species interface now provides mass-based concentration variables (kg/m^3) in addition to mass fractions. This can be used in postprocessing, reports, and visualization. This adds the flexibility to present data in different units depending in the preferences of the person interpreting the results.

RENAMED DEFAULT DOMAIN FEATURE IN THE TRANSPORT OF CONCENTRATED SPECIES INTERFACE

The label of the default domain feature is now **Species Properties**. In previous versions, the label was dynamic and dependent on the transport mechanisms used, for example Convection and Diffusion.

New and Updated Models in Version 5.1

BIOSENSOR DESIGN

This application shows how to use the Application Builder to create a tailored modeling application for designing a biosensor. A flow cell in a biosensor contains an array of micropillars. The concave surfaces of the pillars are coated with an active material that selectively adsorbs biomolecules in the sample stream. This application allows the user to change the design of the sensor by altering input parameters such as pillar diameter, grid spacing, and inlet velocity in order to see how it affects the detection results.

Application Library Path: **Chemical_Reaction_Engineering_Module/Demo_Applications/biosensor_design**

PROTEIN ADSORPTION

This application simulates an ion-exchange column for protein adsorption. Ion-exchange is a powerful method to separate proteins from solutions that is readily used in the biotech and pharmaceutical industries. The application demonstrates the chemical equilibrium features in COMSOL.

Application Library Path: **Chemical_Reaction_Engineering_Module/Mixing_and_Separation/protein_adsorption**

MULTICOMPONENT TUBULAR REACTOR

The application demonstrates how the Chemistry and Transport of Diluted Species interfaces can be used for easy modeling and investigation of complex reaction kinetics and multicomponent mass transfer.

Application Library Path: **Chemical_Reaction_Engineering_Module/Tutorials/multicomponent_tubular_reactor**

MULTISCALE 3D PACKED BED REACTOR

This application has been improved for industrial relevance.

- A plate with holes has been installed at the inlet of the reactor to simulate a more realistic design.
- More complicated second order reversible reaction kinetics.
- Time-dependent study that also shows the start-up behavior of the reactor.

Application Library Path: **Chemical_Reaction_Engineering_Module/Reactors_with_Porous_Catalysts/packed_bed_reactor_3d**

GALLIUM ARSENIDE CHEMICAL VAPOR DEPOSITION (CVD)

The GaAs CVD application has been completely updated and now shows a much easier way to organize the complex system of bulk and surface reactions that are involved in a CVD process. It utilizes the new Reversible Reaction Group feature for surface CHEMKIN import.

In semiconductor manufacturing, CVD reactors are used for depositing thin films on a substrate through molecules and molecular fragments adsorbing and reacting on a surface.

Application Library Path: **Chemical_Reaction_Engineering_Module/Reactors_with_Mass_and_Heat_Transfer/gaas_cvd**

Backward Compatibility with Version 5.0 and Earlier

REACTION ENGINEERING AND CHEMISTRY INTERFACES

The Java API syntax for creating surface species has changed. Running Java files from previous versions may fail due to the new default name index **ads** for surface species and **s** for solids in the *Reaction Engineering* and *Chemistry* interfaces. The surface

species names and solid species names need to be changed accordingly in any Java code:

- Change any species index (s) to (ads)
- Change any species index (S) to (s)

The following shows an example of the needed code change:

Version 5.0 and earlier:

```
model.physics("re").feature("rch1").set("formula", "A+B(S)=>C+D(s)");
model.physics("re").feature("spec1").set("specName", "As(s)");
model.physics("chem").feature("rch1").set("formula", "A+B(S)=>C+D(s)");
model.physics("chem").feature("spec1").set("specName", "As(s)");
```

Change this to

Version 5.1:

```
model.physics("re").feature("rch1").set("formula", "A+B(s)=>C+D(ads)");
model.physics("re").feature("spec1").set("specName", "As(ads)");
model.physics("chem").feature("rch1").set("formula", "A+B(s)=>C+D(ads)");
model.physics("chem").feature("spec1").set("specName", "As(ads)");
```

TRANSPORT OF CONCENTRATED SPECIES INTERFACE

Updated Internal Variable Names

The internal variable names of vector and tensor components have changed in the *Transport of Concentrated Species* interface. In most cases, users will not be affected by this change. When opening a model file created in a previous version, the previous variable definitions remain. However, when updating or re-computing the solution, new variable names will be generated. This means that if any of these variable names have been used in a user-defined expression or in a plot, they need to be updated manually as described below.

In version 5.1, the spatial direction indicator string (x, xy and so on) has been moved to the end of the variable name. This applies to all vector and tensor variables generated by the interface.

For example, the diffusive flux vector components have changed from (5.0 and earlier):

```
tcs.dfluxx_w1, tcs.dfluxy_w1, tcs.dfluxz_w1
```

to (5.1)

```
tcs.dflux_w1x, tcs.dflux_w1y, tcs.dflux_w1z
```

and the components of the diffusion tensor, defined by the Fick's Law diffusion model have been changed from

(5.0 and earlier):

```
tcs.Dfxx_w1, tcs.Dfxy_w1, tcs.Dfxz_w1, tcs.Dfyx_w1, ...
```

to (5.1)

```
tcs.Df_w1xx, tcs.Df_w1xy, tcs.Df_w1xz, tcs.Df_w1yx, ...
```

Java API Syntax Changes

The Java API syntax for setting physics properties has been changed in the *Transport of Concentrated Species* interface. Now the property name corresponds to the name of the GUI section containing the property. In order to run Java files from previous versions, apply the syntax changes as given in the examples below:

Version 5.0 and Earlier:

```
model.physics("tcs").prop("DiffusionModel").setIndex("DiffusionModel",
"MaxwellStefan", 0);
model.physics("tcs").prop("DiffusionModel").setIndex("DiffusionModel",
"MixtureAveraged", 0);
model.physics("tcs").prop("DiffusionModel").setIndex("DiffusionModel",
"FicksLaw", 0);
model.physics("tcs").prop("Convection").setIndex("Convection", "0", 0);
model.physics("tcs").prop("Migration").setIndex("Migration", "0", 0);
model.physics("tcs").prop("Regularization").setIndex("Regularization", "1",
0);
```

Change this to

Version 5.1:

```
model.physics("tcs").prop("TransportMechanism").set("DiffusionModel",
"MaxwellStefan");
model.physics("tcs").prop("TransportMechanism").set("DiffusionModel",
"MixtureAveraged");
model.physics("tcs").prop("TransportMechanism").set("DiffusionModel",
"FicksLaw");
model.physics("tcs").prop("TransportMechanism").set("Convection", false);
model.physics("tcs").prop("TransportMechanism").set("Migration", false);
model.physics("tcs").prop("AdvancedSettings").set("Regularization", "0");
```

SURFACE REACTIONS INTERFACE

Updated Internal Variable Names

The same pattern for vector and tensor variable name changes as in [Transport of Concentrated Species Interface](#), described above, holds for the Surface Reactions interface in version 5.1.

Java API Syntax Changes

The Java API syntax for setting the diffusion tensor of surface species, the molar mass and density of the bulk species, as well as initial values have been changed in the *Surface Reactions* interface.

Setting Diffusion Tensor

The surface species name needs to be included in the first argument of the set method. In order to run Java files from previous versions, apply the syntax changes as in the example below:

Version 5.0 and earlier:

```
model.physics("sr").feature("sp1").set("D_0", new String[]{"Dcs1", "0", "0", "0", "Dcs1", "0", "0", "0", "Dcs1"});
model.physics("sr").feature("sp1").set("D_1", new String[]{"Dcs2", "0", "0", "0", "Dcs2", "0", "0", "0", "Dcs2"});
```

Change this to

Version 5.1:

```
model.physics("sr").feature("sp1").set("D_cs1", new String[]{"Dcs1", "0", "0", "0", "Dcs1", "0", "0", "0", "Dcs1"});
model.physics("sr").feature("sp1").set("D_cs2", new String[]{"Dcs2", "0", "0", "0", "Dcs2", "0", "0", "0", "Dcs2"});
```

Setting Molar Mass and Density

The bulk species name needs to be included in the first argument of the `setIndex` method. Apply the Java API syntax changes as in the example below:

Version 5.0 and earlier:

```
model.physics("sr").feature("sp1").set("M_0", "Mcb1");
model.physics("sr").feature("sp1").set("M_1", "Mcb2");

model.physics("sr").feature("sp1").set("rho_0", "rhocb1");
model.physics("sr").feature("sp1").set("rho_1", "rhocb2");
```

Change this to

Version 5.1:

```
model.physics("sr").feature("sp1").setIndex("M_cb1", "Mcb1", 0);
model.physics("sr").feature("sp1").setIndex("M_cb2", "Mcb2", 0);

model.physics("sr").feature("sp1").setIndex("rho_cb1", "rhocb1", 0);
model.physics("sr").feature("sp1").setIndex("rho_cb2", "rhocb2", 0);
```

Setting Initial Values

Apply the Java API syntax changes as in the example below:

Version 5.0 and earlier:

```
model.physics("sr").feature("init1").set("cs1", "cs10");
model.physics("sr").feature("init1").set("cs2", "cs20");
model.physics("sr").feature("init1").set("cb1", "cb10");
model.physics("sr").feature("init1").set("cb2", "cb20");
```

Change this to

Version 5.1:

```
model.physics("sr").feature("init1").setIndex("initcs", "cs10", 0);
model.physics("sr").feature("init1").setIndex("initcs", "cs20", 1);
model.physics("sr").feature("init1").setIndex("initcb", "cb10", 0);
model.physics("sr").feature("init1").setIndex("initcb", "cb20", 1);
```

Corrosion Module

New Functionality in Version 5.1

IMPROVED CURRENT DISTRIBUTION INITIALIZATION STEP AND NEW STUDIES

- The **Current Distribution Initialization** study step, when used together with an Electrochemistry interface, solves for a stationary solution for the potential variables only (phi_s, phi_l, and global potential variables set up by boundary features). The Current Distribution type during initialization can be set to either Primary (which is the default) or Secondary. The use of a Primary Current Distribution Initialization step eliminates the need to manually derive and provide Initial Values in the Electrochemistry interfaces for most models. However, a Secondary Current Distribution may be needed in problems where the ohmic drop in the electrolyte is negligible, such as in thin wafer deposition with lateral electronic conduction, cathodic protection or mixed potential problems. The study step is supported by all Electrochemistry interfaces.
- The **Stationary with Initialization** study is available for the Secondary and Tertiary Current Distribution interfaces. The study adds one (Primary) Current Distribution Initialization study step, followed by a Stationary study step.
- The **Time Dependent with Initialization** study is available for the Secondary and Tertiary Current Distribution and Battery interfaces. The study adds one (Primary) Current Distribution Initialization study step, followed by a Time dependent study step.

DEFAULT BEHAVIOR OF THE CURRENT DISTRIBUTION INITIALIZATION STUDY STEP

The Current Distribution Initialization study step now uses a Primary initialization as the default (see previous section). The use of a Primary Current Distribution Initialization step eliminates the need to manually derive and provide Initial Values in the Electrochemistry interfaces for most models. However, a Secondary Current Distribution may be needed in applications where the ohmic drop in the electrolyte is negligible, such as in thin wafer deposition with lateral electronic conduction, cathodic protection, or mixed potential problems. The study step is supported by all Electrochemistry interfaces. The changed default may have an impact on the Java API backward compatibility.

IMPROVED REACTIONS NODE IN TERTIARY CURRENT DISTRIBUTION, NERNST-PLANCK

The Reactions feature in the Tertiary Current Distribution, Nernst-Planck interface can now use reaction rates that are generated by the Chemistry interface (only available with a Chemical Reaction Engineering Module license). It also provides an option to account for the reacting volume from selections: Total volume and Pore volume.

DANCKWERTS INFLOW CONDITION IN TERTIARY CURRENT DISTRIBUTION, NERNST-PLANCK

The Inflow feature of the Tertiary Current Distribution, Nernst-Planck interface now provides an option for Danckwerts flux.

CROSS SECTIONAL AREA IN 1D

A new property, **Cross Sectional Area**, is now available in 1D space dimension for the Electrochemistry interfaces. With this property, the boundary features Electrolyte Current and Electrode Current are also made available for 1D space dimension.

PARTICLE-BASED SURFACE AREA IN POROUS ELECTRODES

Porous Electrode Reaction subnode in the Secondary and Tertiary Current Distribution interfaces and in the Lithium-Ion and Battery with Binary Electrolyte interfaces now provides an option to calculate the surface area based on particle type and radius.

The above option to calculate the particle-based surface area is also available in the Porous Matrix Double Layer Capacitance subnode.

POINT AND LINE CURRENT SOURCES IN PRIMARY AND SECONDARY CURRENT DISTRIBUTION INTERFACES

With the Point and Line Current Source features, it is possible to apply a current source at any position within 2D, 2D axisymmetric, and 3D geometries. These are available both for electrolytes and solids.

AC IMPEDANCE, INITIAL VALUES STUDY TYPE

This new study type adds a Frequency Domain, Linearized study step.

New Applications in Version 5.1

ANODE FILM RESISTANCE EFFECT ON CATHODIC CORROSION PROTECTION

This example shows the impact of a buildup of a resistive film on the sacrificial anodes in a cathodic protection problem.

Backward Compatibility with Earlier Versions

SURFACE REACTIONS INTERFACE

See [Chemical Reaction Engineering Module](#).

ECAD Import Module

New Functionality in Version 5.1

It is now possible to rename layers when importing ECAD files. The layer names are used when assigning the names of the geometric objects and selections generated by the import feature. By assigning informative layer names when configuring the import, working with the simulation setup becomes much easier.

Backward Compatibility with Version 5.0

- The default value for the `grouping` property of the `Import` function is now `layer`. Previously the default was `all`.
- For ODB++® and ODB++(X) files (the `type` property is `odb`) new rules apply for the initialization of the `importlayer` property of the `Import` function. Now only layers of type *Metal* and *Dielectric* are initialized with the string on in the `importlayer` string array. Previously this also included layers of type *Drill*.

Electrochemistry Module

New Functionality in Version 5.1

IMPROVED CURRENT DISTRIBUTION INITIALIZATION STEP AND NEW STUDIES

- The **Current Distribution Initialization** study step, when used together with an Electrochemistry interface, solves for a stationary solution for the potential variables only (phi_s, phi_l, and global potential variables set up by boundary features). The Current Distribution type during initialization can be set to either Primary (which is the default) or Secondary. The use of a Primary Current Distribution Initialization step eliminates the need to manually derive and provide Initial Values in the Electrochemistry interfaces for most models. However, a Secondary Current Distribution may be needed in problems where the ohmic drop in the electrolyte is negligible, such as in thin wafer deposition with lateral electronic conduction, cathodic protection or mixed potential problems. The study step is supported by all Electrochemistry interfaces.
- The **Stationary with Initialization** study is available for the Secondary and Tertiary Current Distribution interfaces. The study adds one (Primary) Current Distribution Initialization study step, followed by a Stationary study step.
- The **Time Dependent with Initialization** study is available for the Secondary and Tertiary Current Distribution and Battery interfaces. The study adds one (Primary) Current Distribution Initialization study step, followed by a Time Dependent study step.

IMPROVED REACTIONS NODE IN TERTIARY CURRENT DISTRIBUTION, NERNST-PLANCK

The Reactions feature in the Tertiary Current Distribution, Nernst-Planck interface can now use reaction rates that are generated by the Chemistry interface (only available with a Chemical Reaction Engineering Module license). It also provides an option to account for the reacting volume from selections: Total volume and Pore volume.

CROSS SECTIONAL AREA IN 1D

A new property, **Cross Sectional Area**, is now available in 1D space dimension for the Electrochemistry interfaces. With this property, the boundary features Electrolyte Current and Electrode Current are also made available for 1D space dimension.

PARTICLE-BASED SURFACE AREA IN POROUS ELECTRODES

Porous Electrode Reaction subnode in the Secondary and Tertiary Current Distribution interfaces and in the Lithium-Ion and Battery with Binary Electrolyte interfaces now provides an option to calculate the surface area based on particle type and radius.

The above option to calculate the particle-based surface area is also available in the Porous Matrix Double Layer Capacitance subnode.

POINT AND LINE CURRENT SOURCES IN PRIMARY AND SECONDARY CURRENT DISTRIBUTION INTERFACES

With the Point and Line Current Source features, it is possible to apply a current source at any position within 2D, 2D axisymmetric, and 3D geometries. These are available both for electrolytes and solids.

AC IMPEDANCE, INITIAL VALUES STUDY TYPE

This new study type adds a Frequency Domain, Linearized study step.

Backward Compatibility with Previous Versions

SURFACE REACTIONS INTERFACE

See [Chemical Reaction Engineering Module](#).

Electrodeposition Module

New Functionality in Version 5.1

IMPROVED CURRENT DISTRIBUTION INITIALIZATION STEP AND NEW STUDIES

- The **Current Distribution Initialization** study step, when used together with an Electrochemistry interface, solves for a stationary solution for the potential variables only (phi_s, phi_l, and global potential variables set up by boundary features). The Current Distribution type during initialization can be set to either Primary (which is the default) or Secondary. The use of a Primary Current Distribution Initialization step eliminates the need to manually derive and provide Initial Values in the Electrochemistry interfaces for most models. However, a Secondary Current Distribution may be needed in problems where the ohmic drop in the electrolyte is negligible, such as in thin wafer deposition with lateral electronic conduction, cathodic protection or mixed potential problems. The study step is supported by all Electrochemistry interfaces.
- The **Stationary with Initialization** study is available for the Secondary and Tertiary Current Distribution interfaces. The study adds one (Primary) Current Distribution Initialization study step, followed by a Stationary study step.
- The **Time Dependent with Initialization** study is available for the Secondary and Tertiary Current Distribution and Battery interfaces. The study adds one (Primary) Current Distribution Initialization study step, followed by a Time dependent study step.

IMPROVED REACTIONS NODE IN TERTIARY CURRENT DISTRIBUTION, NERNST-PLANCK

The Reactions feature in the Tertiary Current Distribution, Nernst-Planck interface can now use reaction rates that are generated by the Chemistry interface (only available with a Chemical Reaction Engineering Module license). It also provides an option to account for the reacting volume from selections: Total volume and Pore volume.

DANCKWERTS INFLOW CONDITION IN TERTIARY CURRENT DISTRIBUTION, NERNST-PLANCK

The Inflow feature of the Tertiary Current Distribution, Nernst-Planck interface now provides an option for Danckwerts flux.

CROSS SECTIONAL AREA IN 1D

A new property, **Cross Sectional Area**, is now available in 1D space dimension for the Electrochemistry interfaces. With this property, the boundary features Electrolyte Current and Electrode Current are also made available for 1D space dimension.

PARTICLE-BASED SURFACE AREA IN POROUS ELECTRODES

Porous Electrode Reaction sub node in the Secondary and Tertiary Current Distribution interfaces and in the Lithium-Ion and Battery with Binary Electrolyte interfaces now provides an option to calculate the surface area based on particle type and radius.

The above option to calculate the particle-based surface area is also available in the Porous Matrix Double Layer Capacitance subnode.

POINT AND LINE CURRENT SOURCES IN PRIMARY AND SECONDARY CURRENT DISTRIBUTION INTERFACES

With the Point and Line Current Source features, it is possible to apply a current source at any position within 2D, 2D axisymmetric, and 3D geometries. These are available both for electrolytes and solids.

AC IMPEDANCE, INITIAL VALUES STUDY TYPE

This new study type adds a Frequency Domain, Linearized study step.

New Applications in Version 5.1

COPPER ELECTROLESS DEPOSITION

This example demonstrates transient simulation of electroless deposition of copper using the mixed potential theory.

PRINTED CIRCUIT BOARD DESIGNER

This printed circuit board (PCB) electroplating application, available from the Demo Applications folder, allows designers to study important parameters and run simulations with a few simple clicks. With this resource, a PCB designer can use simulation to analyze a number of factors in the design and manufacturing process. It is possible to evaluate whether a design will be adequate for achieving copper wiring specifications, assess the performance of such devices, and estimate the electroplating process' manufacturing cost.

Backward Compatibility with Previous Versions

SURFACE REACTIONS INTERFACE

See [Chemical Reaction Engineering Module](#).

Fatigue Module

New Application in Version 5.1

DEMO APPLICATION: FRAME FATIGUE LIFE

In this application, the fatigue life of a frame with a cutout subjected to a complex load history is computed. The geometrical parameters can be varied. Load histories and S-N curves can be read from text files.

Geomechanics Module

New Functionality in Version 5.1

NEW MATERIAL MODELS IN MEMBRANE INTERFACE

The Membrane interface has been augmented with several new material models. On the top level, the new **Nonlinear Elastic Material** has been added. It is primarily intended for small strains, and contains the following nonlinear models:

- Ramberg-Osgood
- Hyperbolic Law
- Uniaxial Data
- Hardin-Drnevich
- Duncan-Chang
- Duncan-Selig
- User defined

To both **Linear Elastic Material** and **Nonlinear Elastic Material**, the options to include **Viscoelasticity**, **Plasticity**, and **Creep** have been added.

Viscoelasticity models

- Generalized Maxwell
- Standard Linear Solid
- Kelvin-Voigt

Plasticity models

- von Mises, Tresca, or user-defined yield function
- Associated or non-associated hardening
- Isotropic or kinematic hardening
- Linear or user defined hardening function

Creep Models

- Potential
- Volumetric
- Deviatoric
- User defined

VISCOUS DAMPING

In the Solid Mechanics and Membrane interfaces, it is now possible to specify a viscous damping together with the **Nonlinear Elastic Material**. When using viscous damping, an additional stress proportional to the strain rate will be generated.

Viscous damping can be used both in the time domain and frequency domain.

PLASTICITY IN THE TRUSS INTERFACE

It is now possible to add **Plasticity** to the **Linear Elastic Material** in the Truss interface.

EXTERNAL STRESS

A new subnode to the **Nonlinear Elastic Material**, called **External Stress**, has been added. It is used for adding an extra stress contribution to the stress computed by the constitutive model. This is especially important for geomechanical applications, where it is common to have a pore pressure that does not contribute to the ‘effective stress’

The effect is somewhat similar to entering a stress in the existing **Initial Stress and Strain** node, but with the following differences:

- You can select to only use the external stress as a load contribution, and not add it to the stress tensor. This is typically the case when there is a pore pressure in a porous elastic medium.
- Stress tensors computed by other physics interfaces are visible, and can be selected.
- A pore pressure can be directly selected.
- There can be several external stress contributions to the same domain.

New Applications in Version 5.1

DEMO APPLICATION: PARAMETRIZED CONCRETE BEAM

This application allows the user to calculate the stress and deformation in a reinforced concrete beam. Many inputs can be edited, like loads and boundary conditions, beam dimensions, distribution of reinforcement bars, and material properties of concrete and rebars. Displayed results are axial stress in concrete and bars, displacement, and regions where plasticity occurs.

Updated Applications in Version 5.1

DEEP EXCAVATION

This example has been updated to use the new **External Stress** feature.

TRIAxIAL TEST

This example has been updated to use the new **External Stress** feature.

TUNNEL EXCAVATION

This example has been updated to use the new **External Stress** feature.

BLOCK VERIFICATION

This example has been updated to use the new **External Stress** feature.

Heat Transfer Module

New and Improved Functionality in Version 5.1

LOCAL THERMAL NON-EQUILIBRIUM MULTIPHYSICS INTERFACE

The **Local Thermal Non-Equilibrium** (LTNE) multiphysics interface is a macroscale model designed to simulate heat transfer in porous media where the temperatures into the porous matrix and the fluid are not in equilibrium. It differs from simpler macroscale models for heat transfer in porous media where the temperature difference between the solid and fluid is neglected. The absence of thermal equilibrium can result from fast transient changes but can also be observed in stationary cases. Typical applications are rapid heating or cooling of a porous media using a hot fluid or internal heat generation in one of the phases (due to inductive or microwave heating, exothermic reactions, and so on). This is observed in, for example, nuclear devices, electronics, and fuel cells.

The **Local Thermal Non-Equilibrium** multiphysics interface is available in the **Heat Transfer** folder. It corresponds to a **Heat Transfer in Solids** and a **Heat transfer in Fluids** interface, as well as a **Local Thermal Non-Equilibrium** multiphysics feature.

DEPOSITED BEAM POWER

The new **Deposited Beam Power** feature is available as a boundary condition in 3D. It is used to model beams, such as laser or electron beams, that deposit power on a localized spot. The user interface provides different options to define the beam properties: the profile type—**Gaussian** or **Top-hat disk**—, the beam origin point, its direction vector, its thickness, and the deposited power. From these inputs the **Deposited Beam Power** feature determines the intersection point with the selected boundaries, and a localized heat source is applied following the selected distribution function.

MARANGONI EFFECT

A new boundary multiphysics coupling feature, **Marangoni Effect**, makes it possible to couple a Single-Phase Flow interface and a Heat Transfer interface to model the Marangoni effect induced by temperature-dependent surface tension. Marangoni convection or thermo-capillary convection occurs when the surface tension of an interface between two phases (generally liquid-air) depends on the temperature. It is of primary importance in the fields of welding, crystal growth, and electron beam melting of metals.

The **Marangoni Effect** coupling is available in the **Multiphysics** node as soon as a **Single Phase Flow** and a **Heat Transfer** interfaces are present in a component.

NON-ISOTHERMAL FLOW COUPLING IN POROUS DOMAINS

In parallel to the introduction of the **Enable Porous media domains** option in the **Single-Phase Flow** interface (see [Porous Domain Feature in Single-Phase Flow](#)), the **Non-Isothermal Flow** multiphysics coupling has been updated to handle the coupling between **Heat Transfer in Porous Media** feature and **Fluid and Matrix Properties feature**, which requires the Batteries & Fuel Cells Module, CFD Module, Chemical Reaction Engineering Module, Corrosion Module, Electrochemistry Module, Electrodeposition Module, Microfluidics Module, or Subsurface Flow Module. This capability can be used to model non-isothermal flow in porous media, like natural convection. The viscous dissipation and the work done by pressure force is also available in porous media domains.

Furthermore, it is possible to use the **Non-Isothermal Flow** coupling to simulate non-isothermal flow using an algebraic turbulence model in free domains and to couple it with porous media flow.

OPTIMIZED DEFAULT MESH SETTINGS FOR HEAT TRANSFER INTERFACES

The default mesh settings are now reflecting the use of periodic conditions and pair conditions in any heat transfer interface. When these features are used, the default mesh suggestion provides an identical mesh on the source and destination boundaries in order to minimize the numerical error induced by extrapolation when the meshes on both sides do not coincide.

In addition, the physics-controlled auto mesh suggestion automates meshing for infinite elements. The new auto-mesh suggestion automatically applies swept (3D) or mapped (2D) meshing to domains with infinite elements.

ADDITIONAL CORRELATIONS FOR HEAT TRANSFER COEFFICIENTS

Two convective heat transfer coefficient correlations have been added to the heat transfer coefficients library. These correlations correspond to external flow induced by natural convection around a sphere or a long horizontal cylinder. Such a heat transfer coefficient can be used to reduce the simulation cost when the model configuration corresponds with one of the predefined correlations. In that case, the flow computation and the heat convection in the fluid are replaced by a heat flux boundary condition on the solid boundaries using the corresponding predefined correlation.

PREDEFINED FUNCTIONS FOR BLACKBODY INTENSITY AND BLACKBODY EMISSIVE POWER

The heat transfer interfaces provide new functions, `ht.fIb(T)` and `ht.feb(T)`, to evaluate the blackbody intensity and the blackbody emissive power, respectively. For both functions, the refractive index of the media is accounted for. Because these quantities are defined as functions of a blackbody temperature, it is possible to evaluate them for arbitrary temperatures. For example, `ht.feb(5770[K])` returns the emissive power at 5770 K, which is a temperature used to model the sun as a blackbody.

GENERAL IMPROVEMENTS IN ISOTHERMAL DOMAIN AND ISOTHERMAL DOMAIN INTERFACE FEATURES

- The number of degrees of freedom solved per **Isothermal Domain** feature has been reduced to zero or one (depending on if temperature is prescribed) and is therefore independent from the mesh.
- The **Isothermal Domain** node provides an **Opaque** subnode to define if the domain is opaque or transparent.
- The **Isothermal Domain Interface** boundary condition offers a new **Interface type** option, **Continuity**.
- The **Isothermal Domain Interface** now overrides the default **Thermal Insulation** feature.
- The **Isothermal Domain Interface** feature now contributes with **Thin Layer**. In this case, **Isothermal Domain Interface** represents a layer, at the side of an isothermal domain, in series with **Thin Layer**. Note that it is not possible to define a **Thin Layer** between two isothermal domains or through an isothermal domain.
- Compatibility between **Isothermal Domain Interface** and **Diffuse Surface** has been fixed.

GENERAL IMPROVEMENTS FOR THIN LAYER FEATURE WITH WALL FUNCTIONS (TURBULENT FLOW)

- The **Thin Layer** feature can be combined with **Wall** or **Interior Wall** features when **Non-Isothermal Flow** multiphysics coupling is used.
- For **Turbulent Flow** using wall functions, k -epsilon and k -omega, the turbulent wall function temperature is automatically synchronized with the wall temperature and the inner temperature on the fluid side.

- Depending on the configurations, extra degrees of freedom are created for the wall temperature (`nitf1.TWall`, `nitf1.TWall_u`, and `nitf1.TWall_d`).
- Note that **High Mach number flow** is also supporting **Thin Layer** and wall functions on the same boundary.

GENERAL IMPROVEMENTS FOR SURFACE-TO-SURFACE RADIATION

- When modeling surface-to-surface radiation, it is possible to set exterior domains as opaque. This is done in the **Radiation Settings** section of the physics node.
- A new variable `ht.Famb_gp` gives the possibility to plot the ambient view factor at Gauss points (the `gp` suffix stands for Gauss points). This provides higher accuracy since the simulations use ambient view factors at Gauss points. In addition it makes plots much faster because the view factors are computed at Gauss points and cached during the simulation.
- The **Diffuse Surface**, **Diffuse Mirror**, and **Prescribed Radiosity** features have been removed from the list of fallback features under the existing pair features in heat transfer.
- When surface-to-surface radiation is active, the radiosity postprocessing plot group now has three plots: One for the radiosity defined by the variable `J_plot`, when radiation occurs only on one side and, when radiosity is defined on both sides, two plots for `Ju_plot` and `Jd_plot`.

MISCELLANEOUS

- The variable definition for `ht.DeltaH` in **Heat Transfer with Phase Change** feature has been improved to avoid numerical errors in situations where the latent heat capacity peak was missed.
- The equations displayed in the **Equation** section of all features have been reviewed for more readability and consistency.
- The normal vector variables (`phys.nx`, `phys.ny`, and `phys.nz`, where `phys` corresponds the physics identifier) have been added to the boundary physics interfaces **Heat Transfer In Thin Shells** and **Surface-to-Surface Radiation**.
- The **Periodic Heat Condition** boundary feature was renamed to **Periodic Condition**.
- Unwanted contributions from a **Heat Transfer** interface to a **Solid Mechanics** interface that were observed in some particular cases have been removed when **Pair Thermal Contact** is active on a pair where the structural mechanics **Contact** feature is defined.

- **Thermal Contact** update: `ht.TnRe1` variable was missing when neither joint conductance model nor gap conductance model were selected. This variable is now defined for use in radiative conductance models.
- In **Heat Transfer in Fluids**, when enabling the **Effective Conductivity for Convection option**, the feature ignores the user-defined or predefined velocity model input and sets the fluid velocity to zero.
- When accurate fluxes are evaluated during solving, a warning informs that the accurate value is only available in postprocessing.
- In **Biological Tissue** when the **Include damage integral analysis** with **Temperature threshold** option is active, the detection of temperature out of the **Temperature of necrosis** bounds has been improved. The new implementation improves the computational time and the stability.
- **Inflow Heat Flux** is no longer applicable on interior boundaries nor on the symmetry axis.
- Pair suffixes have been introduced for continuity feature variables to avoid conflicts that have been noticed when several pair identities selections were sharing at least one boundary.
- The feature **Thin Layer** has been updated so that it can be used as a fallback condition of pair features. In this case, the **Layer type** is enforced to **Conductive**. **Resistive** and **General Layer types** are not available.
- A thickness issue in the heat flux definition has been fixed for non-3D geometries so that the flux is given in W/m^2 in all dimensions.
- **Non-Isothermal Flow** is now compatible with **Heat Transfer in Porous Media** provided that **Fluid and Matrix Properties** is used in the flow interface for the corresponding domains.
- The direction of the arrow indicating the flow direction for fans and the radiation direction for surface-to-surface features has been updated to show the actual direction in all circumstances.
- Default plots in 2D and 2D axial symmetry now use tubes instead of lines for boundary plots.

VIEW FACTOR COMPUTATION

This benchmark demonstrates how to compute geometrical view factors for two concentric spheres that irradiate each other. It compares simulation results to exact analytical values.

EVAPORATION IN POROUS MEDIA WITH SMALL EVAPORATION RATE

Evaporation in porous media is an important process in the food and paper industries among others. Many physical effects must be considered: fluid flow, heat transfer, and transport of participating fluids and gases. This tutorial describes laminar air flow through a humid porous medium. The air is dry at the inlet, and its moisture content increases as it flows through the porous media. It is assumed that the initial amount of liquid water in the porous media is large enough and that the evaporation rate is small enough so that the liquid water is never fully evaporated. In addition, the evaporation rate is small enough to neglect the induced properties changes in the porous media.

MODELING A CONICAL DIELECTRIC PROBE FOR SKIN CANCER DIAGNOSIS

The response of a millimeter wave with frequencies of 35 GHz and 95 GHz is known to be very sensitive to water content. This model utilizes a low-power 35 GHz Ka-band millimeter wave and its reflectivity to moisture for non-invasive cancer diagnosis. Since skin tumors contain more moisture than healthy skin, it leads to stronger reflections on this frequency band. Hence the probe detects abnormalities in terms of S-parameters at the tumor locations. A circular waveguide at the dominant mode and a conically tapered dielectric probe are quickly analyzed, along with the probe's radiation characteristics, using a 2D axisymmetric model. Temperature variation of the skin and the fraction of necrotic tissue analyses are also performed.

CONCENTRIC TUBE HEAT EXCHANGER DIMENSIONING TOOL

A heat exchanger made of two concentric tubes contains two fluid domains at different temperatures. The Non-Isothermal Flow multiphysics interface is used to model heat transfer in the heat exchanger. This application computes quantities that characterize the heat exchanger such as exchanged power, pressure drop, and effectiveness. The pipe structure, fluid properties, and boundary conditions are customizable.

HEAT SINK WITH FINS

A parameterized heat sink geometry is studied in combination with conjugate heat transfer. The fluid flow is modeled using the algebraic y^+ model. The model can simulate different heat sink widths and fin dimensions at a given cooling air velocity. The number of heat sinks can also be varied.

The output gives the cooling power and the average pressure drop over the length of the system. The more fins added, the higher the cooling power, but the pressure drop over the heat sink increases accordingly.

Updated Applications in Version 5.1

VACUUM FLASK

This model calculates how much heat a vacuum flask holding hot fluid dissipates over time. The updated version takes benefit of the recently introduced Isothermal Domain feature to monitor the coffee temperature.

ELECTRONIC ENCLOSURE COOLING

This model has been updated and now uses an algebraic turbulence model, y^+ , for the flow. This makes it possible to model faster flow in the device.

In addition, the mesh and solver have been simplified, which make the model setup faster.

The updated model solves 1.1 MDOF and require about 6 GB of memory for solving.

Backward Compatibility with Version 5.0

OBSOLETE FEATURES

A number of features are now obsolete because they have been merged in an existing feature or replaced by a more general one. However, when models from COMSOL 5.0 or before contain these features, some of the obsolete features are visible in version 5.1 with a warning sign notifying that the feature is now obsolete (see [Table 1-1](#)). Some features have been completely removed (see [Table 1-2](#)). A model containing any of them can be opened in 5.1, but the feature will be missing.

It is recommended to update the models. Use the newly available feature and remove the obsolete one to make sure that the model will remain compatible with the future versions of COMSOL Multiphysics.

The table below summarize the changes:

TABLE I-1: OBSOLETE FEATURES THAT ARE OBSOLETE BUT THAN CAN BE LOADED IN 5.1

OBSOLETE FEATURE	REPLACE IN 5.1 BY
Convective Heat Flux	Heat Flux with Convective heat flux option selected
Surface-to-Ambient Radiation	Diffuse Surface with Include surface-to-surface radiation deselected
Surface-to-Surface Radiation	Diffuse Surface with Include surface-to-surface radiation selected
Highly Conductive Layer	Thin Layer with Layer type set to Conductive
Thin Thermally Resistive Layer	Thin Layer with Layer type set to Resistive

TABLE I-2: OBSOLETE FEATURES THAT ARE NO LONGER AVAILABLE

OBSOLETE FEATURE	REPLACE IN 5.1 BY
Infinite Element	Infinite Element are no longer defined in the physic interface. They are defined in the definitions section instead.
Electrochemical Heat flux	Boundary Heat Source
Reaction Heat Flux	Boundary Heat Source
Heat Transfer in Participating Media	Heat Transfer in Solids contributing with Radiation in Participating Media
Porous Matrix	Heat Transfer in Porous Media

Note that it is not possible to add any of the obsolete feature from the COMSOL Desktop in version 5.1; only the new features are available.

OBSOLETE POSTPROCESSING VARIABLES DELETED IN 5.1

List of deleted postprocessing variables: `ndflux_acc`, `ndflux_acc_u`, `ndflux_acc_d`, `ntflux_acc`, `ntflux_acc_u`, `ntflux_acc_d`, `nteflux_acc`, `nteflux_acc_u`, `nteflux_acc_d`, `qout`, `qout_u`, `qout_d`, `qin`, `qin_u`, `qin_d`, `qw`, `qw_u`, and `qw_d`.

CHANGES IN VARIABLE NAMES

The variable `ht.Tvar` used for non-isothermal flow wall functions becomes a fully scoped variable in the following features:

- **Heat Flux** (`ht.hf1.Tvar`), **Line Heat flux**, **Point Heat flux**, and **Out of plane heat flux**.
- **Heat Source** (`ht.hs1.Tvar`), **Boundary heat source**, **Line heat source** and **Point heat source**.
- **Diffuse Surface** (`ht.ds1.Tvar`).

The fully scoped variables P_{tot} , P_{btot} , P_{ltot} , and P_{stot} become P_0 , P_b , P_l , and P_s , respectively, in the following features:

- **Heat Flux**
- **Inflow Heat Flux**
- **Line Heat Flux**
- **Heat Source**
- **Boundary Heat Source**
- **Line Heat Source**
- **Line Heat Source (Thin Rod subfeature)**
- **Line Heat Source on Axis**
- **Layer Heat Source (Thin Layer subfeature)**

In the **Thermal Contact** feature:

- Q_{fric} , $Q_{fricPow}$, and Q_{rate} become Q_b , P_b , and Q_{bpart} , respectively
- Q_{fric0} , Q_{rate0} , $E_{contact0}$, h_{rad0} , h_{gap0} , and $h_{constr0}$ are removed and migrated to existing variables Q_b , Q_{bpart} , $E_{contact}$, h_{rad} , h_{gap} , and h_{constr} , respectively, in old models.

The **Change Effective Thickness** feature from the **Heat Transfer in Thin Shells** physics interface merges $htsh.de$ into the existing variable $htsh.ds$.

The **Heat Transfer in Thin Shells** physics interface also merges the property de and variable $htsh.de$ into existing property ds and variable $htsh.ds$. An API command such as `model.physics("htsh").prop("de").set("de", [...])` now does not have any effect.

Backward Compatibility with Version 4.4

OUT-OF-PLANE HEAT TRANSFER UPDATE

The Out-of-plane heat transfer property, which was available in 2D and 1D geometries, has been removed as of COMSOL Multiphysics 5.0. Now all geometries are considered as 3D geometries and thickness or area parameters are always available.

API Users

For the following features:

- **Heat Source**
- **Heat Flux**

- **Boundary Heat Source**
- **Layer Heat Source**
- **Inflow Heat Flux**

With the **Overall heat transfer rate** option (previously named **Total power** or **Total heat flux**), the thickness parameters (`dz_entr` and `Ac_input`) have been removed.

Compatibility of previous code is maintained for most of the cases. However, three configurations require manual update of the code:

- If you specified 2 times the thickness parameter
- If you specified first the thickness and in a second command the total power value
- If you specified first the thickness and then activate the out of plane heat transfer option.

For these 3 cases you need to rewrite this part of the java code by doing the following, for heat source example (2D and 1D axi symmetry geometries):

`Ptot_50 = Ptot_44_value*ht.d/dz_entr_44_value`

where `Ptot_44_value` is the total power value used in previous versions, `Ptot_50` is the total power value to use since COMSOL Multiphysics 5.0 and `dz_entr_44_value` is the value that were used in previous versions. In 1D geometries, the following update should be done:

`Ptot_50 = Ptot_44_value*ht.d/Ac_input_44_value`

where `Ac_input_44_value` is the value that were used in previous versions.

Flux variables

The flux variables are always given in W/m^2 (also for 2D or 1D geometries). To obtain the corresponding flux variables in W/m on a boundary in 2D for example, the flux variables have to be multiplied by the thickness `ht.d`.

The following variables have been updated following this convention:

- Conductive heat flux, `ht.dflux`
- Convective heat flux, `ht.cflux`
- Translational heat flux, `ht.tr1flux`
- Total energy flux, `ht.teflux`
- Total heat flux, `ht.tflux`
- Boundary convective heat flux, `ht.chflux`
- Radiative heat flux, `ht.rflux`

All flux variables now have the following sign convention: positive heat flux corresponds to heating, whereas negative heat flux corresponds to cooling. `chflux` and `rflux` may have opposite sign compared to previous versions depending on the context where they are defined.

INFLOW HEAT FLUX IS ONLY APPLICABLE ON NON-SOLID DOMAINS

The **Inflow Heat Flux** feature is no longer applicable on boundaries adjacent to the following domain features: **Heat Transfer in Solids**, **Biological Tissue**, and **Isothermal Domain**. This restriction avoids ill-posed configurations due to the null velocity field in these features. This may change a model results if this boundary condition was previously applied to a boundary adjacent to a non-fluid domain.

HEAT TRANSFER WITH PHASE CHANGE NOW USES A VOLUMETRIC FORMULATION

The variables `ht.thetai` now represent volume fractions (they were mass fractions before). This implies changes in the definition of density and heat capacity at constant pressure close to the phase interface. Far from the phase change interface the density and heat capacity are unchanged.

Backward Compatibility with Version 4.3b

INTERIOR FLUXES UPDATED

Because the interior flux variable definitions have been revised, models that use interior fluxes in definition, physics interfaces, or postprocessing show different results in versions since 5.0 than in earlier versions.

IMPROVED ENERGY BALANCE

Changes introduced in the equation formulation to enable better energy balance may slightly modify the results.

Among the changes, the reference enthalpy, H_{Ref} , the value of which is defined up to a constant, is now set to 0 J/kg at the reference temperature and pressure. This changes the absolute values of the enthalpy and several variables based on it. In particular, the value of the convective heat flux is changed. Even the sign may be changed in some cases. However, the quantity of interest, the net convective heat flux, is identical because the shift induced by the change of H_{Ref} applies for the enthalpy definition at the inlet as well as at the outlet.

HEAT TRANSFER IN POROUS MEDIA

Old models using the Porous feature will be supported in the COMSOL Multiphysics interface. It is however recommended to replace this feature by Heat Transfer in Porous Media (see [Obsolete Features and Variables](#)). For a model using the API, it will be necessary to update the model to use Heat Transfer in Porous Media.

SURFACE-TO-SURFACE RADIATION

The surface radiosity, previously named J , is now named `ht.J` by default (assuming that `ht` is the physics interface tag).

Models created using versions 4.3 and 4.3a that contain surface-to-surface radiation support the old syntax in multiphysics interfaces. It is, however, recommended that you update the models to use the new syntax. Models created in a version before 4.2a need to be opened in version 4.3 or 4.3a and resaved before being opened in 5.0 or later version.

For a model using the API, you must update the model to use the new syntax.

RADIATION IN PARTICIPATING MEDIA

The radiative intensity variables names, previously named $I1$ and so on, became `ht.I1` (when there are less than 10 discrete directions) or `ht.I01` (when there are 10 or more discrete directions) by default.

Models created in versions 4.3 and 4.3a that include radiation in participating media support the old syntax in multiphysics interfaces. It is, however, recommended that you update the models to use the new syntax. Models created in a version before 4.2a need to be opened in version 4.3 or 4.3a and resaved before being reopened in 5.0 or later version. In addition, for all versions, it may be necessary to regenerate the default solver to get the correct solver settings.

The $S2$ quadrature in 2D has been replaced by the LSE symmetric quadrature. It has been updated so that it coincides with the other quadratures. LSE symmetric quadratures were already used for $S4$, $S6$, and $S8$ in 2D and for all 3D quadratures. This quadrature update can result in significant changes in 2D models using $S2$ quadrature. However, the theoretical error is not different and the 2D and 3D implementations are now consistent.

THE DEFAULT MODEL LIST HAS BEEN REMOVED

Heat transfer user interfaces no longer have the default model list. This list was used to change the default feature for heat transfer models. You can obtain a similar result by adding the corresponding domain feature immediately below the default feature and setting its domain selection to *All domains*.

Any model files for Java that modify the default model value require a manual update.

IMPROVED STABILIZATION OF HEAT TRANSFER IN SOLIDS

The streamline diffusion stabilization for *Heat Transfer in Solids* and *Biological Tissue* features has been improved. It now accounts for contributions from linear source terms from the *Heat Source*, *Out-of-Plane Convective Cooling*, *Out-of-Plane Radiation*, and *Out-of-Plane Heat Flux* features. This improves the robustness of the convergence when these contributions are large. This change may modify the convergence behavior of existing models.

FRAME SELECTION IN DISCRETIZATION SECTION

The *Frame type* list, which was previously available when a moving mesh was detected, has been removed. The frame type is now controlled by the features.

UPDATE OF FEATURE VARIABLE NAMES

In order to avoid name conflicts between feature variable names, some of them have been renamed. This change concerns contributive features. Because they are contributing, it is possible to have similar features active on the same boundary. To make it possible to distinguish the variables from each feature, the feature scope has been introduced in the variable name so that variable names now use a prefix. These changes affect the following feature variables:

- Heat flux
- Out-of-plane heat flux
- Convective cooling
- Out of plane convective cooling
- Boundary heat source
- Heat source
- Line heat source
- Point heat source

- Edge/Point heat flux
- Electrochemical reaction heat flux
- Reaction heat flux

For example, in a *Boundary Heat Source* feature, `ht.surf` has been renamed into `ht.bhs1.surf` (assuming that `ht` is the physics interface tag). In the *Heat Flux* feature, the variable that was previously named `ht.q0_hf1` is now `ht.hf1.q0`.

Any model files for Java that use the old variable names in expressions (such as expressions used for plotting or data evaluation that include such old variable names) need to be updated manually.

NEW DEFAULT FLUID FEATURES AND OPAQUE SUBFEATURE

All Model MPH-files created in version 4.3 are converted to appear and behave the same way as in 5.0 or later version.

Due to these new default features, model files for Java can be simplified. In addition, model files for Java that add the *Fluid* feature or the *Opaque* subfeature with the default tag require a manual update to avoid duplicate tag conflicts.

WEAK CONSTRAINTS UPDATE FOR FLUID FLOW USER INTERFACES

The weak constraint formulations for some boundary conditions in the following interfaces have been updated:

- Laminar Flow
- Turbulent Flow, k - ϵ
- Turbulent Flow, low-Re k - ϵ
- Non-Isothermal Flow
- Conjugate Heat Transfer

See the CFD Module Release Notes for details.

Weak constraints for the Interior Wall feature are no longer available.

REVISION TO THE TURBULENCE MODELS

The formulations of some variables in the turbulence models have been revised in order to improve accuracy. Simulations using a turbulence model can display a different convergence behavior in version 5.0 and later compared to version 4.3, and the numerical result can differ slightly between the versions.

Obsolete Features and Variables

The following features are still supported when models created in older versions are loaded in version 5.1. However, they will be removed in future versions. Hence, you should replace them by the new corresponding feature.

- **Porous Matrix** is replaced by **Heat Transfer in Porous Media**.
- **Infinite Element** domain feature should be removed and **Infinite Element Domain** should be declared in the **Definitions** node.
- **Electrochemical Heat Flux** is replaced by **Boundary Heat Source**.
- **Reaction Heat Flux** is replaced by **Boundary Heat Source**.
- **Heat Transfer in Participating Media** is replaced by **Heat Transfer with Radiation in Participating Media**.

The table below contains a list of variables that are available but obsolete. They should each be replaced by the new corresponding expressions because they will be removed in future versions.

TABLE 1-3: OBSOLETE VARIABLES AND CORRESPONDING EXPRESSIONS

OBSOLETE VARIABLE	CORRESPONDING EXPRESSION
qin	-qr_in
qout	qr_out
qw	-qr_net
ndflux_acc	ndflux
ndflux_acc_u	ndflux_u
ndflux_acc_d	ndflux_d
ntflux_acc	ntflux
ntflux_acc_u	ntflux_u
ntflux_acc_d	ntflux_d
nteflux_acc	nteflux
nteflux_acc_u	nteflux_u
nteflux_acc_d	nteflux_d

MEMS Module

New Functionality in Version 5.1

PERFORATIONS FEATURE FOR THIN FILM FLOW

A new **Perforations** feature is available for thin film damping, enabling the modeling of thin film flow in structures with etch holes.

The perforations feature acts as a sink for gas that is proportional to both the ambient pressure and to the pressure difference with respect to the ambient on the other side of the perforated surface. The constant of proportionality is known as the admittance (Y) and can either be defined directly or determined from the model due to Bao (M. Bao and H. Yang “Squeeze film air damping in MEMS,” *Sensors and Actuators A: Physical*, vol. 136, no. 1, pp 3–27, 2014).

OUT-OF-PLANE MOTION OPTION FOR BORDER FLOW BOUNDARY CONDITION IN THIN FILM FLOW

A new option is available for the **Border Flow** boundary condition for thin film flow. Selecting **Out-of-plane** motion for the Border flow type calculates the pressure gradient at the boundary using the model of Gallis and Torczynski (M. A. Gallis and J. R. Torczynski, “An Improved Reynolds-Equation Model for Gas Damping of Microbeam Motion,” *Journal of Microelectromechanical Systems*, vol. 13, pp. 653–659, 2004). This model has been shown to agree well with detailed CFD and Monte Carlo simulations that model both the thin film flow domain and the surrounding gas. The model applies for both rarefied and nonrarefied flows up to Knudsen numbers of approximately 1.

POINT MATRIX EVALUATION FEATURE ENABLES THE DISPLAY OF TENSOR QUANTITIES AT A POINT

The new **Point Matrix Evaluation** feature (available in the base package) enables the convenient display of tensor quantities at a point. This is particularly useful for the **Piezoelectric Devices** interface, which defines tensor material properties in both the local and the global coordinate systems. Consequently, it is now possible to view, for example, the elasticity matrix in the global as well as the local system.

PIEZOELECTRIC RATE GYROSCOPE

This tutorial shows how to analyze a tuning fork based piezoelectric rate gyroscope, using the **Piezoelectric Devices** interface. The direct piezoelectric effect is used to drive an in-plane tuning fork mode. This mode is coupled to an out-of-plane mode by the Coriolis force and the resulting out-of-plane motion is sensed by the reverse piezoelectric effect. The geometry of the tuning forks is designed so that the eigenfrequencies of the nearby modes are separated in frequency space. The frequency response of the system is computed and the rotation rate sensitivity is evaluated.

PIEZOELECTRIC ENERGY HARVESTER

This tutorial shows how to analyze a simple, cantilever based, piezoelectric energy harvester, using the **Piezoelectric Devices** interface. A sinusoidal acceleration is applied to the energy harvester and the output power is evaluated as a function of frequency, load impedance and acceleration magnitude.

Input mechanical power, output electrical power, and voltage as a function of load impedance.

PIEZOELECTRIC VALVE

Piezoelectric valves are frequently employed in medical and laboratory applications due to their fast response times and quiet operation. Their energy-efficient operation, also dissipates little heat, which is often important for these applications.

This tutorial shows how to model a piezoelectric valve in COMSOL using the **Piezoelectric Devices** interface together with a **Hyperelastic Material** node. The valve is actuated by a stacked piezoelectric actuator. A hyperelastic seal is compressed against a valve opening by the actuator and the contact is modeled.

This model requires the Nonlinear Structural Materials Module in addition to the MEMS Module.

DISC RESONATOR ANCHOR LOSSES

This tutorial shows how to compute the anchor loss limited quality factor of a diamond disc resonator, using the **Solid Mechanics** interface. The resonator is anchored to the substrate by a polysilicon post and power is transmitted to the substrate through the post. A perfectly matched layer is used to represent the essentially infinite substrate. The tutorial is based on a paper presented at the COMSOL Conference 2007 in

Grenoble (P. Steeneken, “Parameter Extraction and Support-Loss in MEMS Resonators,” COMSOL Users Conference 2007, Grenoble).

MICROPUMP MECHANISM

Micropumps are key components of microfluidic systems with applications ranging from biological fluid handling to microelectronic cooling. This tutorial simulates the mechanism of a valveless micropump that is designed to be effective at low Reynolds numbers, overcoming hydrodynamic reversibility. Valveless pumps are often preferred in microfluidic systems because they minimize the risk of clogging and are gentle on biological material. The **Fluid-Structure Interaction** interface is used to solve for the flow of the fluid and the associated deformation of the structure. In addition, the **Global ODEs and DAEs** interface is used to demonstrate how to perform a time-resolved integration of the total flow throughout the pumping cycle.

Backward Compatibility with Version 4.3b

For models using contact, plot expressions are not updated when you open an old model where the pair names have been edited manually. Also, opening an old model where a pair name has been manually edited might affect the solver configuration. It is recommended that you regenerate a new default solver configuration.

Microfluidics Module

Backward Compatibility with Version 4.3a

The Free Molecular Flow and Transitional Flow user interfaces now form the Molecular Flow Module. A Molecular Flow Module license is required to use these interfaces.

Backward Compatibility with Version 4.3

BRINKMAN EQUATIONS AND FREE AND POROUS MEDIA FLOW

The momentum equations in the Brinkman Equations interface and the Free and Porous Media Flow interface have been corrected. The term $-Q_{br} \cdot \mathbf{u} / \varepsilon_p^2$ in the right hand side previously lacked the factor $1/\varepsilon_p^2$, where ε_p is the porosity.

MOBILITY

In physics interfaces modeling migration (Transport of Diluted Species), anisotropic mobility is now supported, and the mobility can be set to follow the Nernst-Einstein relation. The default setting when making new models has been changed to use the Nernst-Einstein relation. Model files for Java generated prior to 4.3b using Migration will have to be modified manually to account for these changes.

Mixer Module

New Functionality in Version 5.1

PART LIBRARY IN MIXER MODULE

In order to facilitate fast and efficient setup of mixer geometries, the Mixer Module includes a part library with predefined geometry components typical of mixer equipment. The part library includes impeller geometries for axial impellers, radial impellers, and impellers designed for highly viscous fluids. Apart from impellers, three types of different tank geometries and a cylindrical impeller shaft geometry are available in the part library. All mixer parts are modularized through a number of input parameters corresponding to important geometrical properties of each part. These can be adjusted in order to fit the mixer system under investigation.

Molecular Flow Module

New Functionality in Version 5.1

NUMERICAL IMPROVEMENTS FOR FASTER FREE MOLECULAR FLOW CALCULATIONS

The **Free Molecular Flow** interface is more efficiently parallelized, allowing more cores to be used more efficiently during the computation. The table below shows how much faster three models were run in COMSOL Multiphysics 5.1 compared to the previous version. A 10 core machine was used to run the simulations.

TABLE 1-4: PERFORMANCE IMPROVEMENTS IN SELECTED MODELS FROM THE APPLICATION LIBRARY

MODEL	CPU TIME (5.0)	CPU TIME (5.1)	SPEEDUP
Ion Implanter	5m 15s	2m 1s	2.6
Evaporator	2h 24m 4s	18m 31s	7.8
Outgassing Pipes	2m 57s	45s	3.9

MULTIPLE SPECIES FOR MOLECULAR FLOW

It is now possible to model multiple species in the **Free Molecular Flow** interface. This is done in the **Dependent Variables** section, where the number of species and name for each of the **Incident molecular fluxes** can be specified. COMSOL will then create dependent variables for each species, and there will be one input field for each species in all appropriate boundary conditions.

NEW OPTION FOR SPECIFYING OUTGASSING RATE

The outgassing rate can now be specified in units of $[(\text{torr}\cdot\text{l})/\text{cm}^2/\text{s}]$ or $[(\text{mbar}\cdot\text{l})/\text{cm}^2/\text{s}]$ (the equivalent SI unit is W/m^2). Using these units is available in the new **Thermal desorption rate** option when **Outgassing wall** is selected in the **Wall** boundary condition.

New Applications in Version 5.1

A new application showing high vacuum chemical vapor deposition is available, which showcases the multiple species capabilities. Chemical vapor deposition (CVD) is a process often used in the Semiconductor industry to grow layers of high-purity solid material on top of a wafer substrate. CVD is achieved using many different techniques and across a range of pressures, from atmospheric to ultrahigh vacuum (UHV/CVD).

Since UHV/CVD is performed at pressures below 10^{-6} Pa (10^{-8} Torr), gas transport is achieved by molecular flow, and lacks any hydrodynamic effects such as boundary layers. In addition, with no gas-phase chemistry involved (due to the low frequency of molecular collisions), growth rate will be determined by the number density of species and surface molecular decomposition processes.

This tutorial uses multiple species in the **Free Molecular Flow** interface to model the growth of silicon wafers from CVD. The effects of different pumping curves are explored after running an auxiliary sweep.

Backwards Compatibility with Version 5.0 and Prior

Due to the fact that multiple species are now supported, some variable names have changed. In general, variables will now have `_G` appended to their names, where **G** is the name specified for the **Incident molecular fluxes** in the **Dependent Variables** section. So, if a model is updated and resolved, the expressions used in results should be changed as follows:

- The number density, `fmf.N` should be changed to `fmf.N_G`.
- The pressure should be changed from `p` to `fmf.p_G`.
- The reconstructed number density, `fmf.N` should be changed to `fmf.ndr1.Nr_G`, where `ndr1` is the tag of the **Number Density Reconstruction** feature.
- Any other variables can be updated using the **Replace Expression** button available during results processing.

In addition, new variables exist for the total pressure, number density etc.

- The total number density, `fmf.ntot`.
- The total pressure `fmf.ptot`.
- The total incident molecular flow, `fmf.Gtot`.

Multibody Dynamics Module

New Functionality in Version 5.1

APPLIED FORCE/MOMENT ON JOINT DEGREE OF FREEDOM

It is now possible to apply a load directly on the degree of freedom of a joint. As an example, when using a prismatic joint to model a hydraulic cylinder, the cylinder force can be prescribed. The change appears in the **Applied Force and Moment** subnode of a joint, where it is now possible to select also **Joint** in the **Applied on** section.

ATTACHMENT FILTER ON JOINTS

In a large Multibody Dynamics model, the selection lists for **Source** and **Destination** in the **Attachment selection** section for a **Joint** can become very long. It is now possible to filter these lists using a selection of objects. When the new **Use selection filter** check box is selected, two new subnodes (**Source Filter** and **Destination Filter**) will appear under the joint node. It is then possible to select domains, so that only attachments or rigid domains present on the selected domains will appear in the **Source** and **Destination** lists in the **Attachment selection** section.

VISCOUS DAMPING

It is now possible to specify a viscous damping together with the **Linear Elastic Material** in the Multibody Dynamics interface. When using viscous damping, an additional stress proportional to the strain rate will be generated.

ROLLER CONDITION

It is now possible to use a **Roller** boundary condition in Multibody Dynamics interface. This condition can be used to model a set of sliding boundaries, or a symmetry condition.

POINT TRAJECTORY PLOT

It is now possible to plot the trajectory of geometric points or cut-points of a moving body using **Point Trajectories** in **3D Plot Group** and **2D Plot Group**. This new plot is used in number of existing Multibody Dynamics Application Library models.

New Applications in Version 5.1

DEMO APPLICATION: TRUCK MOUNTED CRANE ANALYZER

Many trucks are equipped with cranes for load handling. Such cranes have a number of hydraulic cylinders controlling the motion, and several mechanisms. In this application, a rigid body analysis of a crane is performed in order to find the payload capacity for the specified orientation and extension of the crane. This application also provides the usage of all the hydraulic cylinders and highlights the limiting one. The capacity of hydraulic cylinders can be modified in order to improve the payload capacity and the usage of hydraulic cylinders.

Updated Applications in Version 5.1

Point Trajectories plot added to the following models:

- Centrifugal Governor
- Crooked Four Bar Mechanism
- Double Pendulum
- Golf Swing Mechanics
- Gyroscope
- Slider Crank Mechanism

Nonlinear Structural Materials Module

New Functionality in Version 5.1

NEW MATERIAL MODELS IN MEMBRANE INTERFACE

The Membrane interface has been augmented by several new material models. On the top level, the new **Nonlinear Elastic Material** node has been added. It is primarily intended for small strains, and contains the following nonlinear models:

- Ramberg-Osgood
- Power Law
- Uniaxial Data
- Bilinear Elastic
- User Defined

For both the **Linear Elastic Material** and **Nonlinear Elastic Material** nodes, options to include **Viscoelasticity**, **Plasticity**, **Creep**, and **Viscoplasticity** have been added.

Viscoelasticity models

- Generalized Maxwell
- Standard Linear Solid
- Kelvin-Voigt

Plasticity models

- von Mises, Tresca, Hill, or user-defined yield functions
- Associated or non-associated hardening
- Isotropic or kinematic hardening
- Linear or user-defined hardening function

Creep models

- Norton
- Norton-Bailey
- Garofalo
- Nabarro-Herring

- Coble
- Weertman
- Potential
- Volumetric
- Deviatoric
- User defined

Viscoplasticity models

- Anand

VISCOUS DAMPING

In the Solid Mechanics and Membrane interfaces, it is now possible to specify a viscous damping together with the **Nonlinear Elastic Material** and the **Hyperelastic Material**.

When using viscous damping, an additional stress proportional to the strain rate will be generated.

Viscous damping can be used both in the time domain and frequency domain.

PLASTICITY IN THE TRUSS INTERFACE

It is now possible to add **Plasticity** to the **Linear Elastic Material** in the Truss interface.

EXTERNAL STRESS

A new subnode, called **External Stress**, has been added to the **Nonlinear Elastic Material** and the **Hyperelastic Material**. It is used for adding an extra stress contribution to the stress computed by the constitutive model. This is somewhat similar to entering a stress in the existing **Initial Stress and Strain** node, but with the following differences:

- You can choose to only use the external stress as a load contribution, and not add it to the stress tensor. This is typically used when there is a pore pressure in a porous elastic medium.
- Stress tensors computed by other interfaces are visible, and can be selected.
- A pore pressure can be selected directly.
- There can be several external stress contributions to the same domain.

HYPERELASTIC SEAL

The geometry is created with a smaller number of operations, and selections already exist in the geometry sequence. The shape of the plate against which the seal is pressed has been modified to be angled instead of straight.

COMPRESSION OF AN ELASTOPLASTIC PIPE

The penalty method is now used in the contact analysis. This improves performance for a case like this where high accuracy in the contact results are not important.

Optimization Module

New Functionality in Version 5.1

TABLE DATA INPUT TO GLOBAL LEAST-SQUARES OBJECTIVE

You can now use a locally defined table or a **Table** under **Results** to define the data source for experimental data in the **Settings** window for the **Global Least-Squares Objective** node.

Backward Compatibility with Version 4.3a

The Optimization check box in the Stationary and Time Dependent study steps have been removed. Instead, use the Optimization study step. The Optimization check box does appear in models created in earlier versions of COMSOL where you have selected that check box.

Particle Tracing Module

General New Functionality in Version 5.1

PERFORMANCE IMPROVEMENTS TO ACCUMULATORS ON DOMAINS

The **Accumulator** for domains is now much faster and more accurate than in Version 5.0. It is no longer necessary to specify a manual time step to ensure the stability and accuracy of models with domain **Accumulator** nodes.

In the **Accumulator** settings, if accumulation is performed over **Elements and time**, it is now possible to specify the **Source interpolation**, which controls the way in which accumulated variables are computed in the mesh elements each particle passes through in each time step.

RELEASE OF PARTICLES FROM A TEXT FILE

It is now possible to release particles by importing data from a text file. With the **Release from Data File** node it is possible to specify the data columns from which the initial position, velocity, and initial values of auxiliary dependent variables are taken.

RELEASE FROM GRID WITH SPECIFIED COMBINATIONS

When releasing particles with the **Release from Grid** node, it is now possible to specify that the particles should be released at **Specified combinations** of coordinates or at **All combinations**. If the **All combinations** option is selected, the number of specified initial values for each coordinate must be equal.

NEW OPTIONS FOR SAMPLING FROM DISTRIBUTIONS

When initializing particle velocity using a spherical, hemispherical, conical, or Maxwellian distribution, it is possible to specify whether the **Sampling from distribution** is set to **Deterministic** or **Random**.

NEW PARTICLE-PARTICLE INTERACTION SETTINGS.

In the settings for **Particle-Particle Interaction** it is now possible to select a **Linear elastic** force. It is also possible to apply a cutoff length; particles that are separated by a distance greater than the cutoff length will not exert particle-particle interaction forces on each other.

BIDIRECTIONALLY COUPLED PARTICLE TRACING STUDY STEP

The **Bidirectionally Coupled Particle Tracing** study step solves for particle trajectories using a time-dependent solver and computes all other dependent variables using a Stationary solver. It then sets up an iterative solver loop using the **For** and **End For** nodes that solves for the particle trajectories and fields in a self-consistent manner.

New Functionality in Particle Tracing for Fluid Flow

MULTIPHYSICS INTERFACE

The Fluid-Particle Interaction interface automatically sets up the physics interfaces needed for modeling fluid-particle interactions, including the Laminar Flow interface and the Particle Tracing for Fluid Flow interface. It also adds the **Fluid-Particle Interaction** node in the Multiphysics branch, which computes the volume force acting on the fluid due to drag forces exerted on the particles.

New Functionality in Charged Particle Tracing

MULTIPHYSICS INTERFACES

The Particle-Field Interaction, Non-Relativistic interface sets up the physics interfaces needed for modeling particle-field interactions, including the Electrostatics interface and the Charged Particle Tracing interface. It also adds the **Electric Particle-Field Interaction** node, which computes the space charge density of particles and applies it as a source term for the electric potential computed by the Electrostatics interface.

The Particle-Field Interaction, Relativistic interface sets up the physics interfaces needed for modeling particle-field interactions when the particles are relativistic; this includes the Electrostatics, Magnetic Fields, and Charged Particle Tracing interfaces. It also adds the **Electric Particle-Field Interaction** node and the **Magnetic Particle-Field Interaction** node to model the self-induced electric and magnetic forces that act on beams of relativistic particles.

SPACE CHARGE LIMITED EMISSION

A dedicated multiphysics node, **Space Charge Limited Emission**, can now be used to release charged particles. When used with the **Electric Particle-Field Interaction** node, the **Space Charge Limited Emission** node can be used to compute the space charge limited current of charged particles that are released from a surface.

PARTICLE BEAM FEATURE

The **Particle Beam** feature can be used to release beams of particles by specifying the emittance and sampling the initial velocity from a transverse phase space distribution.

NEW COLLISION FORCES

Stochastic collision forces between particles and a rarefied background gas are now implemented by adding a **Collisions** node. Different types of collisions can be specified by adding subnodes to the **Collisions** node. The following subnodes are available:

- **Elastic**
- **Attachment**
- **Excitation**
- **Ionization**
- **User Defined**

In addition, a dedicated **Friction Force** node can now be used for deterministic modeling of the interaction between particles and a background gas.

New Applications in Version 5.1

RED BLOOD CELL SEPARATION

This application examines the separation of red blood cells and platelets in a microfluidic channel using dielectrophoresis. The red blood cell and platelet diameter are inputs, as well as the electromagnetic frequency and applied potential. The separation efficiency is computed, and there are visual plots for the particle trajectories, the electric potential, and the fluid velocity.

CHILD'S LAW BENCHMARK

Space charge limited emission is a phenomenon that restricts the current of charged particles that can be released from a surface. As the electron current released by a cathode increases, so does the magnitude of the charge density in the immediate vicinity of the cathode. This distribution of charge density exerts an electric force on the emitted electrons, directed toward the cathode. The space charge limited current is the maximum current that can be released such that the emitted particles are not repelled back toward the cathode.

In this example, the space charge limited current in a plane-parallel vacuum diode is computed using the **Space Charge Limited Emission** node. The resulting electric potential distribution and current are compared with the analytical solution given by Child's Law. The current density is computed using a study called **Bidirectionally**

Coupled Particle Tracing, which establishes a bidirectional coupling between the particle trajectories and the electric potential.

RELATIVISTIC DIVERGING ELECTRON BEAM

When modeling the propagation of charged particle beams at high currents and relativistic speeds, the space charge and beam current create significant electric and magnetic forces that tend to expand and focus the beam, respectively. The Charged Particle Tracing interface uses an iterative procedure to efficiently compute the strongly coupled particle trajectories and electric and magnetic fields for a beam operating at constant current. A mesh refinement study confirms that the solution agrees with the analytical expression for the shape of a relativistic beam envelope.

Backwards Compatibility with Version 5.0 and Earlier

OLD NODES FOR PARTICLE-FIELD AND FLUID-PARTICLE INTERACTIONS ARE OBSOLETE

The old **Particle Field Interaction** and **Fluid-Particle Interaction** nodes, which could be added directly to the Charged Particle Tracing and Particle Tracing for Fluid Flow interfaces, respectively, are now obsolete. They will be removed in a future version. Simulations of particle-field and fluid-particle interactions should instead use the dedicated Multiphysics nodes.

ELASTIC COLLISION FORCE NODE HAS BEEN REPLACED

If a model created in Version 5.0 or earlier containing the **Elastic Collision Force** node is opened in Version 5.1, this node will be replaced. If the **Collision model** in the old **Elastic Collision Force** node was set to **Monte Carlo**, a **Collisions** node with the **Elastic** subnode will be created. If instead the **Collision model** was set to **Friction**, the **Friction Force** subnode will be added.

Pipe Flow Module

New Functionality in Version 5.1

NEW Y-JUNCTIONS AND N-WAY JUNCTIONS AND EXTENDED FUNCTIONALITY OF T-JUNCTIONS

More junction conditions have been added. They now allow the user to add arbitrary expressions for pressure drop due to turbulent friction, in addition to the built-in loss factor formulation.

- T-junctions, bends, and valves now have the option to specify pressure drop with an arbitrary expression.
- Y-junction pressure drop has been added as a new feature, with both loss factor and arbitrary pressure drop specification.
- An n-way junction pressure drop has been added as a new feature.
- An angle tolerance setting has been added under the Advanced section in T-junction and Y-junction. This makes it possible to adjust the tolerance for a geometry not being exactly a T- or Y-junction.
- Noncritical errors in the geometry for a T-junction generates a warning when solving.
- The geometry analysis code for T- and Y-junctions now allows curved segments connecting to a junction.

New Applications in Version 5.1

GAS BOX DESIGNER

This application is created with the Application Builder and computes the back pressure in a network of pipes fed by a system of mass flow controllers (MFC). Each MFC supplies a given flow rate to one of the pipes, and can only function correctly if the back pressure remains below 760 Torr.

Application Library Path: **Pipe_Flow_Module/Demo_Applications/gas_box**

Plasma Module

Backward Compatibility with Version 4.4

Old models will be updated to use the new boundary conditions, but re-solving a model still gives the same results as before. The following changes to the model are made upon opening:

- If an Electric Potential feature is applied to boundaries adjacent to a Plasma Model feature, it is converted to a Metal Contact feature. The settings from the original model are retained. If an Electric Potential feature is applied to boundaries adjacent to a Charge Conservation feature, it remains the same.
- If an Electric Potential feature is applied to a combination of boundaries adjacent to a Plasma Model and Charge Conservation feature, a Metal Contact feature is created for the boundaries adjacent to the Plasma Model feature, and the Electric Potential feature remains on boundaries adjacent to the Charge Conservation feature.
- Any External Surface Charge Accumulation boundary conditions are replaced by a new Dielectric Contact feature. The settings from the original model are retained. The External Surface Charge Accumulation boundary condition is now obsolete.

Ray Optics Module

New Functionality in Version 5.1

PERFORMANCE IMPROVEMENTS TO ACCUMULATORS ON DOMAINS

The **Accumulator** for domains is now much faster and more accurate than in Version 5.0. It is no longer necessary to specify a manual time step to ensure the stability and accuracy of models with domain **Accumulator** nodes.

In the **Accumulator** settings, if accumulation is performed over **Elements and time**, it is now possible to specify the **Source interpolation**, which controls the way in which accumulated variables are computed in the mesh elements each ray passes through in each time step.

RELEASE OF RAYS FROM A TEXT FILE

It is now possible to release rays by importing data from a text file. With the **Release from Data File** node it is possible to specify the data columns from which the initial position, ray direction, and initial values of auxiliary dependent variables are taken.

RELEASE FROM GRID WITH SPECIFIED COMBINATIONS

When releasing rays with the **Release from Grid** node, it is now possible to specify whether the rays should be released at **Specified combinations** of coordinates or at **All combinations**. If the **All combinations** option is selected, the number of specified initial values for each coordinate must be equal.

NEW OPTIONS FOR SAMPLING FROM DISTRIBUTIONS

When initializing ray direction using a spherical, hemispherical, or conical distribution, it is possible to specify whether the **Sampling from distribution** is **Deterministic** or **Random**.

INTENSITY CALCULATION IN GRADED MEDIA

It is now possible to compute ray intensity in a graded medium. In the settings window for the Geometrical Optics interface, if **Using curvature tensor** is selected from the **Intensity computation** list, the intensity is computed in a way that accounts for first and second derivatives of the refractive index.

MULTIPHYSICS INTERFACE

The **Ray Heating** multiphysics interface can be used to model heating by rays as they propagate through an absorbing medium. It sets up the Geometrical Optics and Heat Transfer in Solids interfaces, as well as the **Ray Heat Source** Multiphysics node. The **Ray Heat Source** node uses the energy lost by rays in an absorbing medium as a heat source that is automatically applied when computing the temperature.

BIDIRECTIONALLY COUPLED RAY TRACING STUDY STEP

The **Bidirectionally Coupled Ray Tracing** study step solves for ray trajectories using a time-dependent solver and computes all other dependent variables using a Stationary solver. It then sets up an iterative solver loop using the **For** and **End For** nodes that solves for the ray trajectories and fields in a self-consistent manner. This study step can be used, for example, to model ray propagation in an absorbing medium with a temperature-dependent refractive index.

NEW OPTIONS FOR APPLYING THIN DIELECTRIC FILMS

The **Material Discontinuity** settings window now includes built-in options to set up anti-reflective coatings and single-layer films of specified reflectance or transmittance.

It is also possible to specify that the **Thin Dielectric Film** subnodes that are added to a **Material Discontinuity** are periodic, and to specify a number of repeating unit cells. Individual **Thin Dielectric Film** subnodes can be added to or removed from the repeating unit cell using the **Repeat layer in multilayer films** check box.

IMPROVED SUPPORT FOR FREQUENCY-DEPENDENT MATERIAL PROPERTIES

It is now always possible to include frequency dependence in the definition of the refractive index. When including the frequency in an expression for the refractive index, it must be included within the `noenv()` operator, which indicates that a quantity should be computed on the rays instead of the domain.

IMPROVED HANDLING OF TOTAL INTERNAL REFLECTION

Rays can now undergo total internal reflection at material discontinuities even if no preallocated secondary rays are available. Combinations of reflection and refraction at material discontinuities still require secondary rays.

PART LIBRARY FOR THE RAY OPTICS MODULE

The Ray Optics Module now includes a Part Library that consists of ready-to-use parameterized geometric entities for ray optics simulations. The Part Library includes the following geometric entities:

- Beam Splitters
- Cemented Doublets
- Cylindrical Lenses
- Mirrors
- Prisms
- Retroreflectors
- Spherical Lenses

OPTICAL MATERIALS DATABASE

The new Optical Materials Database is available for the Ray Optics Module and the Wave Optics Module. It contains data for a large number of materials for the dispersion of the real and the imaginary parts of the refractive index. Among the available materials there are a large number of glasses used for lenses, semiconductor materials, and other areas.

New Applications in Version 5.1

DISTRIBUTED BRAGG REFLECTOR (DBR) FILTER

A distributed Bragg reflector (DBR) consists of multiple alternating layers of two materials. Each material has a different refractive index, resulting in a repeating pattern of high and low refractive indices in the direction perpendicular to the DBR layers. As light propagates through this structure, reflections occur at each interface between the layers. This application computes the reflectance of a DBR filter for a distribution of free-space wavelengths. Either a band-stop filter or a notch filter can be analyzed. The user inputs include the refractive index of each layer, the number of periods in the DBR, and the threshold reflectance within the stop-band.

TRANSPARENT LIGHT PIPE

Light pipes are structures that can be used to transport light between different locations. In general, they can be divided into two major groups: tubes lined with a reflective coating and transparent solids that contain light via total internal reflection.

In this example, light is transported through a bent light pipe by total internal reflection. The effect of the pipe shape on the transmittance is investigated.

Backward Compatibility with Version 5.0

RENAMED SETTINGS

The Wall condition **Bounce** has been renamed to **Specular reflection**.

CHANGES TO FREQUENCY-DEPENDENT MATERIAL PROPERTIES

The **Frequency-dependent refractive indices** check box has been renamed to **Allow frequency distributions at release features**.

Expressions involving the ray frequency must always be included within the `noenv()` operator. Some expressions for user-defined refractive indices may fail to be evaluated properly in Version 5.1 unless this operator is applied.

THIN DIELECTRIC FILM SETTINGS

In version 5.0, if the ray intensity was computed, the option to add **Thin Dielectric Film** nodes to a **Material Discontinuity** node was always available. In Version 5.1, the Thin Dielectric Film subnode can only be added if **Add layers to surface** or **Add layers to surface, repeating** is selected from the **Thin dielectric films on boundary** list in the **Material Discontinuity** settings window. If a model created in Version 5.0 is opened in Version 5.1, and any **Thin Dielectric Film** nodes have been added, then **Add layers to surface** is selected from the **Thin dielectric films on boundary** list instead of the default **None**.

CHANGES TO THE CIRCULAR WAVE RETARDER

The **Retardance** of the **Circular Wave Retarder** now indicates the rotation angle of the polarization ellipse of a ray that passes through the device. In Version 5.0, the retardance corresponded to half of this rotation angle. The behavior of the **Circular Wave Retarder** is now consistent with the documentation. If a model created in Version 5.0 is opened in Version 5.1, the expression for the **Retardance** of any **Circular Wave Retarder** node is multiplied by 2.

OBSOLETE DEPOSITED RAY POWER NODE

The **Deposited Ray Power** node for absorbing domains is obsolete and will be removed in future versions. For simulations in which rays generate heat due to absorption, the **Ray Heat Source** Multiphysics node should be used instead.

RF Module

New Functionality in Version 5.1

POSTPROCESSING WAVE VECTOR VARIABLES

Postprocessing variables have been added for the wave vectors for the incident wave and the various diffraction orders (including the reflected wave) in periodic ports and diffraction order ports. These variables can be used in arrow plots for visualization of the directions the different waves are propagating in.

IMPROVED SCATTERING BOUNDARY CONDITION IN 2D AXISYMMETRY

The Scattering boundary condition in 2D axisymmetry now includes a plane wave option for the scattered wave type. This means that you now, for example, can set up the Scattering boundary condition to absorb a wave propagating along a coaxial waveguide. Furthermore, it is also possible to enter the field of the incident wave, assumed to propagate along the symmetry axis. This is useful for exciting and absorbing waves propagating along coaxial waveguides, for example, but also for propagating Gaussian beams in free space.

LOSS TANGENT, LOSS ANGLE AND LOSS TANGENT, DISSIPATION FACTOR

The old loss tangent model has been renamed as Loss tangent, Loss angle, and a new electric displacement field model, the Loss tangent, dissipation factor model, is added, from which it is possible to directly enter a value directly for the dissipation factor.

VOLTAGE STANDING WAVE RATIO (VSWR) POSTPROCESSING VARIABLE

Many commercial off-the-shelf (COTS) antennas or one-port devices are characterized with VSWR. VSWR is available as a variable for excited ports.

SURFACE ROUGHNESS ON LOSSY CONDUCTIVE SURFACES

Surface roughness is available as a subfeature for the Transition Boundary condition and the Impedance Boundary condition. Surface current is scaled based on the surface loss caused by roughness using two asymptotic models.

SURFACE CURRENT DENSITY ON TRANSITION BOUNDARY CONDITION

This new subfeature for the Transition Boundary condition is a one-sided surface current source.

HEXAGONAL PERIODIC STRUCTURES

Hexagonal periodic structures are now analyzed using periodic ports. The user provides the incident wave direction to the sides of the hexagonal cell, whereas the correct primitive vectors for the periodic structure (which are different from the sides of the hexagonal cell) are calculated internally. The periodic ports have also been improved to handle partitioned port boundaries.

DAMPED-DRIVEN POLARIZATION

For the Electromagnetic Waves, Transient interface, you can now select the **Drude-Lorentz dispersion model** from the **Electric displacement field** list in the **Wave Equation, Electric** node's **Settings** window. With this selection, **Drude-Lorentz Polarization** features can be added as subnodes to the **Wave Equation, Electric** node. The **Drude-Lorentz Polarization** nodes add domain ODEs for the polarization that are solved together with the time-dependent wave equation for the magnetic vector potential.

S-PARAMETERS SET TO ZERO FOR EVANESCENT MODES

For port modes that are not propagating (that is, the waves are evanescent), the S-parameter is now set to zero. This setting simplifies the use of the S-parameter variables during postprocessing.

New Applications in Version 5.1

The following new applications are available in the RF Module Application Library:

- Axisymmetric Cavity Resonator
- Modeling a Biconical Antenna for EMI/EMC Testing
- Simulating Antenna Crosstalk on an Airplane's Fuselage
- Modeling a Conical Dielectric Probe for Skin Cancer Diagnosis
- Fast Numerical Modeling of a Conical Horn Lens Antenna
- Designing a Waveguide Diplexer for the 5G Mobile Network
- Modeling of a Mobile Device Antenna
- Simulating Wireless Power Transfer in Circular Loop Antennas
- Numerical Modeling of a UHF RFID Tag for Animal Identification
- Thermal Drift in a Microwave Filter Cavity
- Hexagonal Grating
- Time-Domain Modeling of Dispersive Drude-Lorentz Media

- Plasmonic Wire Grating Analyzer (runnable application available in the Demo Application folder)
- Corrugated Circular Horn Antenna Simulator (runnable application available in the Demo Application folder)

Backward Compatibility with Version 4.3a

The following models have been moved from the RF Module Application Library to the Wave Optics Module Model Library:

- Beam Splitter
- Photonic Crystal
- Scattering on Substrate
- Optical Scattering Off of a Gold Nanosphere
- Dielectric Slab Waveguide
- Fabry-Perot Cavity

Semiconductor Module

New Functionality in Version 5.1

INDIRECT OPTICAL TRANSITIONS

Optical absorption in silicon and other indirect band-gap materials can now be included using the new **Indirect Optical Transition** feature. The photogeneration rate in silicon can be calculated automatically using an empirical model—making it quick and convenient to simulate silicon photovoltaic devices. Alternately, for other materials, a user-defined option enables the photogeneration rate to be directly specified using either the refractive index or a value for the absorption coefficient. This can be used as a standalone feature within the Semiconductor interface, or can be coupled with the Electromagnetic Waves, Frequency Domain or Electromagnetic Waves, Beam Envelopes interfaces (requires the Wave Optics Module).

DIAMOND MATERIAL ADDED TO SEMICONDUCTOR MATERIAL LIBRARY

Diamond is now available as a material within the Semiconductor Material Library.

POSTPROCESSING VARIABLES IMPROVED FOR SPONTANEOUS EMISSION

New postprocessing variables have been added to enable the spontaneous emission spectrum to be plotted as a function of photon energy, wavelength, and frequency. Also, it is now possible to directly access the photon energy, wavelength, and frequency variables throughout the extra dimension, where previously these quantities needed to be calculated using an expression in terms of the angular frequency. The new variables are:

- Emitted power per unit volume and wavelength: `semi.ot1.dP_dLambda`
- Emitted power per unit volume and frequency: `semi.ot1.dP_df`
- Emitted power per unit volume and energy: `semi.ot1.dp_dE`
- Photon wavelength: `semi.ot1.lambda`
- Photon frequency: `semi.ot1.freq`
- Photon energy: `semi.ot1.energy`

DEMO APPLICATION: WAVELENGTH TUNABLE LED

The new application simulates the emission properties of a GaN-based light emitting diode (LED) device. The device has a double heterostructure design, with an optically active InGaN layer placed between two layers of GaN. The indium composition of the InGaN layer can be varied in order to control the emission wavelength. The device current, emission intensity, and efficiency are calculated either for a single voltage or as a function of voltage over a user-defined range. The emission spectrum is calculated, and, when the peak emission falls within the viable range, the corresponding RGB value is computed to display the emission color.

Structural Mechanics Module

New Functionality in Version 5.1

VISCOUS DAMPING

In the Solid Mechanics and Membrane interfaces, it is now possible to specify a viscous damping with the **Linear Elastic Material** node. When using viscous damping, an additional stress proportional to the strain rate will be generated.

Viscous damping can be used both in the time domain and frequency domain.

EXTERNAL STRESS

A new subnode called **External Stress** has been added to the **Linear Elastic Material** node. It is used for adding an extra stress contribution to the stress computed by the constitutive model. This is somewhat similar to entering a stress in the existing **Initial Stress and Strain** node, but with the following differences:

- You can choose to only use the external stress as a load contribution, and not add it to the stress tensor. This is typically used when there is a pore pressure in a porous elastic medium.
- Stress tensors computed by other physics interfaces are visible, and can be selected.
- A pore pressure can be directly selected.
- There can be several external stress contributions to the same domain.

MULTIPHYSICS COUPLING FOR HYGROSCOPIC SWELLING

When Solid Mechanics is combined with the Transport of Diluted Species or Transport of Diluted Species in Porous Media interfaces, there is a new multiphysics coupling called **Hygroscopic Swelling**. It has the same types of settings as the **Hygroscopic Swelling** subnode in the material models. The purpose of this feature is to transfer a moisture concentration computed in one of the transport interfaces into a hygroscopic swelling strain.

PHYSICS-BASED MESH IN TRUSS INTERFACE

The default mesh for the Truss interface now includes one element per edge. This resolution is sufficient for all truss structures, so it is only when modeling cables that you need to add more elements. The advantage is that with a default mesh, the models will be much smaller.

CHANGED DEFAULT DISCRETIZATION IN TRUSS INTERFACE

The default shape function in the Truss interface is now linear. In previous versions, it was quadratic. Because all pure truss applications can be solved exactly with a linear shape function, the need to use a quadratic shape function appears only when combining truss elements with elements from another physics interface on the same edge. The advantage is a reduction in model size. Combined with the new meshing scheme, there will be no internal nodes on edges in a truss structure, so the computationally expensive **Straight Edge Constraint** is no longer needed for such cases.

COMPUTATION OF MASS PROPERTIES

The structural mechanics physics interfaces (Solid Mechanics, Membrane, Shell, Plate, Truss, Beam, and Multibody Dynamics) can now return complete mass properties to the **Mass properties** node under **Definitions**. All types of mass contributions from the physics interfaces are accounted for:

- Mass density in all material models
- Added mass
- Point mass and inertia
- Mass and inertia on rigid domains and rigid connectors
- Inertia with respect to thickness in beams and shells
- Inertia with respect to rotation around the axis of a beam.

It is also possible to compute the mass properties in the deformed configuration by selecting **Spatial frame** in the settings for **Mass Properties**.

PART LIBRARIES

In the new Part Libraries window, two families of geometries have been added for structural mechanics use.

Bolts

In the Bolts folder, you can find geometries for bolts, nuts, and washers. The bolts are prepared for easy meshing and for use with the **Bolt Pre-Tension** feature.

Beams

In the Beams folder, you can find 2D geometries for common beam cross sections. The main purpose is for use with the Beam Cross Section interface, but it is also possible to extrude the sections for detailed 3D modeling.

The cross sections are divided into the following categories:

- Generic — containing fully parametrized cross sections for C, I, L, and T profiles.
- European standard — containing profiles like HEA, IPE, and so on.
- US Standard — containing profiles such as C, M, and W in both metric and imperial units.

New Applications in Version 5.1

PRESSURE SENSOR HYGROSCOPIC SWELLING

This example studies the moisture diffusion across the mold compound that protects a MEMS pressure sensor. The hygroscopic swelling induced by the moisture diffusion induces a drift in the measured characteristics of the sensor.

VIBRATION ANALYSIS OF A THICK BEAM

In this example, the free and forced vibrations of a deep beam are studied. The solutions for eigenfrequency, frequency-response, and transient analyses are computed using a Timoshenko beam formulation and are compared with analytical results.

Updated Applications in Version 5.1

PRESTRESSED BOLTS IN A TUBE CONNECTION

In this example, the bolts are now taken from the Part Libraries instead of being drawn manually.

VISCOELASTIC STRUCTURAL DAMPER

A synthetization of time history from the harmonic results using the inverse Fourier transform has been added.

Backward Compatibility with Version 4.3b

- If an old model using the Beam interface is opened, the Euler-Bernoulli formulation is used. The Beam Formulation selection is still shown but cannot be changed from Euler-Bernoulli. If Advanced Physics Options is enabled, then the new section **Backward Compatibility** is shown. If you clear the **Use pre 4.4 formulation** check box, then the new formulation is used. Doing this enables the use of Timoshenko beams, but you must manually handle solver settings like segregation and scaling. This

legacy option does not support beams mixed with solids or shells in cases where the same names for degrees of freedom were used in both physics interfaces.

- For models using mechanical contact, plot expressions are not updated when you open an old model where the pair names have been edited manually. Also, opening an old model where a pair name has been manually edited might affect the solver configuration. It is recommended that you regenerate a new default solver configuration.

Backward Compatibility with Version 4.3a

- In the Solid Mechanics interface (SolidMechanics), the property `ControlsSpatialFrame` no longer exists, and COMSOL now always assumes that the displacements control the spatial frame. Remove all lines setting the value of `ControlsSpatialFrame`. See Support Knowledge Base article 1156.
- When opening old models that include contact, the penalty factor control is set to **User defined** and the contact normal penalty factor to the expression used in the model.

Subsurface Flow Module

Backward Compatibility with Version 4.4

SOLUTE TRANSPORT

The Solute Transport interface has been replaced by the new interface Transport of Diluted Species in Porous Media.

Wave Optics Module

New Functionality in Version 5.1

POSTPROCESSING WAVE VECTOR VARIABLES

Postprocessing variables have been added for the wave vectors for the incident wave and the various diffraction orders (including the reflected wave) in periodic ports and diffraction order ports. These variables can be used in arrow plots for visualization of the directions the different waves are propagating in.

IMPROVED SCATTERING BOUNDARY CONDITION IN 2D AXISYMMETRY

The Scattering boundary condition in 2D axisymmetry now includes a plane wave option for the scattered wave type. This means that you now, for example, can set up the Scattering boundary condition to absorb a wave propagating along a coaxial waveguide. Furthermore, it is also possible to enter the field of the incident wave, assumed to propagate along the symmetry axis. This is useful for exciting and absorbing waves propagating along coaxial waveguides, for example, but also for propagating Gaussian beams in free space.

LOSS TANGENT, LOSS ANGLE AND LOSS TANGENT, DISSIPATION FACTOR

The old loss tangent model is renamed as Loss tangent, Loss angle, and a new electric displacement field model, the Loss tangent, dissipation factor model has been added, from which it is possible to directly enter a value for the dissipation factor.

SURFACE CURRENT DENSITY ON TRANSITION BOUNDARY CONDITION

This new subfeature for the Transition Boundary condition is a one-sided surface current source.

WAVELENGTH DOMAIN STUDY

With the Wavelength Domain study step you sweep the vacuum wavelength instead of sweeping the frequency as is done for the Frequency Domain study. The Wavelength Domain study creates the variables `root.lambda0` and `phys.lambda0` (where `phys` is the tag of the physics interface) representing the vacuum wavelength. The frequency is still the driving parameter for the Electromagnetic Waves, Frequency Domain and the Electromagnetic Waves, Beam Envelopes interfaces, but now `root.freq` is defined as `c_const/root.lambda0`. When plotting global parameters like the S-parameters against the sweep parameter, the wavelength automatically appears on the x -axis.

HEXAGONAL PERIODIC STRUCTURES

Hexagonal periodic structures are now correctly analyzed using periodic ports. The user provides the incident wave direction to the sides of the hexagonal cell, whereas the correct primitive vectors for the periodic structure (which are different from the sides of the hexagonal cell) are calculated internally. The periodic ports have also been improved to handle partitioned port boundaries.

DAMPED-DRIVEN POLARIZATION

For the Electromagnetic Waves, Transient interface, you can now select the **Drude-Lorentz dispersion model** from the **Electric displacement field** list in the **Wave Equation, Electric** node's **Settings** window. With this selection, **Drude-Lorentz Polarization** features can be added as subnodes to the **Wave Equation, Electric** node. The **Drude-Lorentz Polarization** nodes add domain ODEs for the polarization that are solved together with the time-dependent wave equation for the magnetic vector potential.

FIELD CONTINUITY IN THE UNIDIRECTIONAL BEAM ENVELOPES INTERFACE

To model ring resonators, for example, using the unidirectional formulation of the Electromagnetic Waves, Beam Envelopes interface, you can now use the Field Continuity feature. This feature forces the tangential components of the electric and magnetic fields to be continuous across boundaries even if there is a discontinuity in the prescribed phase approximation at the boundary. This boundary condition is only available for unidirectional propagation and on interior boundaries. It is normally hidden but becomes available if **Advanced Physics Options** is selected in the **Show** menu in the Model Builder toolbar.

OPTICAL MATERIALS DATABASE

The new Optical Materials Database is available in the Wave Optics Module and the Ray Optics Module. It contains data for a large number of materials for the dispersion of the real and the imaginary parts of the refractive index. Among the available materials are a large number of glasses used for lenses, nonlinear optical materials (the linear refractive index only), and semiconductor materials.

The following models now use the Optical Materials Database: `scattering_nanosphere`, `plasmonic_wire_grating`, and `scatterer_on_substrate`.

S-PARAMETERS SET TO ZERO FOR EVANESCENT MODES

For port modes that are not propagating (that is, the waves are evanescent), the S-parameter is now set to zero. This setting simplifies the use of the S-parameter variables during postprocessing.

New Applications in Version 5.1

The following new applications are included in the Wave Optics Module Application Library:

- Hexagonal Grating
- Optical Ring Resonator Notch Filter
- Time-Domain Modeling of Dispersive Drude-Lorentz Media
- Plasmonic Wire Grating Analyzer (runnable application in the Demo Applications folder)

Material Library

New Materials and Material Data in Version 5.1

- A number of Boehler alloys have been added.
- The viscosity of liquid has been revised and added for the following materials: Cd, Co, Ga, In, Si, Tl, and Zn.
- Stress-strain, stress-rupture, and S-N data has been added for PWA 1484.
- Stress-rupture data has been added for 2.25Cr-Mo and P122 steel.
- Stress-strain and yield strength data has been added for Fe-3Si.

LiveLink for MATLAB

New Functionality in Version 5.1

The following sections list new and improved functionality in LiveLink™ for MATLAB®:

- `mphopen` can now open a new window to assist with opening and searching for model files; for example, it can display recently opened files.
- `mphload` and `mphopen` now support the use of passwords for protected files.
- The window opened by `mphnavigator` has been updated with a new column that displays allowed property values. This makes it easier to set properties in a model. At the command line you can now also use `mphgetproperties` to check allowed property values.
- The new wrapper function, `mphevalpointmatrix`, can evaluate matrix quantities at specified points in the geometry.
- `mphevalglobalmatrix` supports new arguments to perform matrix transformation.
- Geometry display by `mphplot` now shows domain labels, and has improved support for 1D geometry.
- `mphinterp` now supports the arguments `differential` and `evalmethod`.
- The `mphapplicationlibraries` function replaces the `mphmodellibrary` function for showing the Application Libraries window.
- `mphmeshstats` supports a new argument called `qualityhistogram` for controlling the number of bins in the quality distribution histogram. Additional new arguments support the use of selections: `entity`, `selection`, and `type`.
- `mphmatrix` has been enriched by these new arguments: `complexfun`, `matherr`, `symmetry`, `nullfun`.
- In the `mphinterp` command, the automatic resolution setting now takes into account the number of sectors. This can affect the solution and give slightly less accurate results but faster evaluation time. To retrieve a solution as in previous versions, set the resolution setting manually.

Backward Compatibility with Version 5.0

- The `mphint2` command: For data sets other than Solution, Particle, Cut, Time Integral, Time Average, Surface, and Line, the integration order now corresponds

to the element refinement. The default value is set to 4 instead of 1. This change can affect the solutions in cases where the problem is not resolved well enough.

- The `mphapplicationlibraries` function replaces the `mphmodellibrary` function for showing the Application Libraries window.

LiveLink for Excel

New Functionality in Version 5.1

The following major new functionality is now available in LiveLink™ for Excel®:

- It is now possible to save a model as Visual Basic for Applications (VBA) code. Expanding on previously available functionality for saving models as Java files and M-files, saving model VBA-files makes it much easier to create subroutines to control COMSOL simulations from Microsoft Excel. The COMSOL API in VBA can be used to control all aspects of modeling, including setting up models, changing settings, solving models, and performing postprocessing and extracting results to Excel sheets.
- Support has been added for loading and saving Excel files even if Excel is not installed on the computer. This makes it possible to access Excel files from a wider range of computers and operating systems and provides access to Excel files from an application created in the Application Builder.

The COMSOL API for Use with Java®

COMSOL 5.1 API Changes

ACOUSTICS MODULE CHANGES

The ratio of specific heats is now a material parameter. If the default value was used in a model, add the following line to the Java file to obtain the old behavior:

```
model.physics("cpf").feature("cpf1").set("gamma_mat", "userdef");
```

ELECTROCHEMISTRY MODULES CHANGES

There are changes for the Tertiary Current Distribution, Nernst-Planck interface. The Java API syntax for creating and accessing vectors and tensors has changed, as well as the syntax for settings physics properties. See the following examples:

- Specifying the diffusion tensor.

Old syntax:

```
model.physics("tcdee").feature("ice1").set("D_0", new  
String[]{"D1", "0", "0", "0", "D1", "0", "0", "0", "D1"});  
model.physics("tcdee").feature("ice1").set("D_1", new  
String[]{"D2", "0", "0", "0", "D2", "0", "0", "0", "D2"});  
model.physics("tcdee").feature("ice1").set("D_2", new  
String[]{"D3", "0", "0", "0", "D3", "0", "0", "0", "D3"});
```

New syntax (includes the species name in the first argument of the set method):

```
model.physics("tcdee").feature("ice1").set("D_c1", new  
String[]{"D1", "0", "0", "0", "D1", "0", "0", "0", "D1"});  
model.physics("tcdee").feature("ice1").set("D_c2", new  
String[]{"D2", "0", "0", "0", "D2", "0", "0", "0", "D2"});  
model.physics("tcdee").feature("ice1").set("D_c3", new  
String[]{"D3", "0", "0", "0", "D3", "0", "0", "0", "D3"});
```

- Accessing tensor components (in the definitions of other variables, for example).

Old syntax:

```
Dxx_c1, Dxy_c1, Dxz_c1...
```

New syntax (component index after the species name):

```
D_c1xx, D_c1xy, D_c1xz...
```

- Specifying the reaction rate vector.

Old syntax:

```

model.physics("tcdee").feature("react1").set("R_0", new
String[]{"R1"});
model.physics("tcdee").feature("react1").set("R_1", new
String[]{"R2"});
model.physics("tcdee").feature("react1").set("R_2", new
String[]{"R3"});

```

New syntax (includes the species name in the first argument of the set method):

```

model.physics("tcdee").feature("react1").set("R_c1", new
String[]{"R1"});
model.physics("tcdee").feature("react1").set("R_c2", new
String[]{"R2"});
model.physics("tcdee").feature("react1").set("R_c3", new
String[]{"R3"});

```

- Accessing vector components.

Old syntax:

```
tfluxx_c1, tfluxy_c1, tfluxz_c1
```

New syntax (component index after the species name):

```
tflux_c1x, tflux_c1y, tflux_c1z
```

- Specifying initial values:

Old syntax:

```

model.physics("tcdee").feature("init1").setIndex("c2", "c20", 0);
model.physics("tcdee").feature("init1").setIndex("c3", "c30", 0);
model.physics("tcdee").feature("init1").setIndex("V", "V0", 0);

```

New syntax:

```

model.physics("tcdee").feature("init1").setIndex("initc", "c20",
1);
model.physics("tcdee").feature("init1").setIndex("initc", "c30",
2);
model.physics("tcdee").feature("init1").setIndex("initphil",
"V0");

```

- Selecting properties.

Old syntax:

```

model.physics("tcdee").prop("Convection").set("Convection", 1,
"0");
model.physics("tcdee").prop("ConvectiveTerm").setIndex("Convectiv
eTerm", "noncons", 0);

```

New syntax (the property name corresponds to the section in the user interface):

```

model.physics("tcdee").prop("TransportMechanism").set("Convection
", 1, "0");

```

```
model.physics("tcdee").prop("AdvancedSettings").setIndex("ConvectiveTerm", "noncons", 0);
```

For information about API backward compatibility for the Surface Reactions interface, see the [Chemical Reaction Engineering Module](#) notes.

ECAD IMPORT MODULE CHANGES

- The default value for the grouping property of the Import function is now set to layer. Previously the default was all.
- For ODB++® and ODB++(X) files (the type property is odb) new rules apply for the initialization of the importlayer property of the Import function. Now only layers of type *Metal* and *Dielectric* are initialized with the string on in the importlayer string array. Previously this also included layers of type *Drill*.

LIVELINK FOR SOLID EDGE CHANGES

The default value for the keepfree property of the LiveLinkSolidEdge function is now set to on. Previously the default was set to off.

LIVELINK FOR SOLIDWORKS CHANGES

The default value for the keepfree property of the LiveLinkSOLIDWORKS function is now set to on. Previously the default was set to off.

COMSOL 5.0 API Changes

DOCUMENTATION

The *COMSOL Multiphysics Programming Reference Manual* replaces the *COMSOL API for use with Java® Reference Manual*.

DEPRECATED METHODS

The following methods are deprecated in COMSOL 5.0:

TABLE I-5: DEPRECATED METHODS AND THEIR REPLACEMENTS

DEPRECATED METHOD	NEW METHOD
com.comsol.model.OptFeature.field()	OptFeature.comp()
com.comsol.model.OptFeature.field(String)	OptFeature.comp(String)
com.comsol.model.Group.identifier()	Group.paramName()
com.comsol.model.Material.identifier()	ModelEntity.tag()
com.comsol.model.MaterialModel.identifier()	ModelEntity.tag()

TABLE I-5: DEPRECATED METHODS AND THEIR REPLACEMENTS

DEPRECATED METHOD	NEW METHOD
<code>com.comsol.model.ModelNode.identifier()</code>	<code>ModelEntity.tag()</code>
<code>com.comsol.model.physics.Physics.identifier()</code>	Use <code>tag()</code> instead.
<code>com.comsol.model.physics.MultiphysicsCoupling.identifier()</code>	<code>ModelEntity.tag()</code>
<code>com.comsol.model.Group.identifier(String)</code>	<code>Group.paramName(String)</code>
<code>com.comsol.model.Material.identifier(String)</code>	<code>ModelEntity.tag(String)</code>
<code>com.comsol.model.MaterialModel.identifier(String)</code>	<code>ModelEntity.tag(String)</code>
<code>com.comsol.model.ModelNode.identifier(String)</code>	<code>ModelEntity.tag(String)</code>
<code>com.comsol.model.physics.Physics.identifier(String)</code>	Use <code>tag(string)</code> instead.
<code>com.comsol.model.physics.MultiphysicsCoupling.identifier(String)</code>	<code>ModelEntity.tag(String)</code>
<code>com.comsol.model.ModelEntity.name()</code>	<code>ModelEntity.label()</code>
<code>com.comsol.model.ModelEntity.name(String)</code>	<code>ModelEntity.label(String)</code>
<code>com.comsol.model.ParameterEntity.set(String, int, double)</code>	Use the 0-based version (<code>setIndex</code>) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, double)</code>	Use the 0-based version (<code>setIndex</code>) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, double[])</code>	Use the 0-based version (<code>setIndex</code>) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, int)</code>	Use the 0-based version (<code>setIndex</code>) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, int[])</code>	Use the 0-based version (<code>setIndex</code>) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, int, double)</code>	Use the 0-based version (<code>setIndex</code>) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, int, int)</code>	Use the 0-based version (<code>setIndex</code>) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, int, String)</code>	Use the 0-based version (<code>setIndex</code>) instead.

TABLE 1-5: DEPRECATED METHODS AND THEIR REPLACEMENTS

DEPRECATED METHOD	NEW METHOD
<code>com.comsol.model.ParameterEntity.set(String, int, String)</code>	Use the 0-based version (<code>setIndex</code>) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, String[])</code>	Use the 0-based version (<code>setIndex</code>) instead.

COMSOL 4.4 API Changes

PHYSICS INTERFACES

- The default Initial Value features in the following interfaces have been extended:
 - Transport of Concentrated Species
 - Reacting Flow in Porous Media (rfcs)
 - Reacting Flow
 - Rotating Machinery, Reacting Flow

Previously, only the initial mass fractions could be specified. Now the initial values can be specified in terms of mass fractions, mole fractions, molar concentrations, number densities, or densities. When opening an old Model MPH-file, *Mixture specification* is set to *Mass fractions*, and the mass fractions specified are entered in the respective text fields. In old model files for Java, commands specifying the initial value of a mass fraction return an error in version 5.0. To specify the initial mass fraction of a species `w2`, which is the second in the list of species (seen in the Dependent Variables section when clicking the interface), change the COMSOL Java command

```
model.physics("chcs").feature("init1").set("w2", 1, "0.1");
```

to

```
model.physics("chcs").feature("init1").set("w0", 2, "0.1");
```

- The Pressure Acoustics, Frequency Domain interface and the Boundary Mode Acoustics interface have a new default feature. If the default feature has been edited in a Java file, the following line should be added to the Java file to obtain the old behavior for Pressure Acoustics, Frequency Domain:

```
model.physics("acpr").feature().create("pam1",
"PressureAcousticsModel").selection().all();
```

or

```
model.physics("acbm").feature().create("pam1",  
"PressureAcousticsModel").selection().all();
```

for Boundary Mode Acoustics.

This line should be added directly after the line where the physics interface is created.

- For models using mechanical contact, the variable field name in the solver settings for the old contact variables is different than in earlier versions. For example:
 - In version 5.0: `mod1_solid_contact_old_p1`, `mod1_solid_cm_old_p1`.
 - In earlier versions: `mod1_solid_contact_p1_old`, `mod1_solid_cm_p1_old`.
- The parameter form in the property `EquationForm` can no longer be set to `ModeAnalysis` in 3D and 2D axisymmetric calculations for Pressure Acoustics, Frequency Domain, and Pressure Acoustics, Transient. In these cases, the parameter `modeFreq` has been removed.

COMSOL 4.3b API Changes

PHYSICS INTERFACES

- The *Crosswind diffusion* formulation has been updated for all physics user interfaces supporting Crosswind diffusion. Models solved with the new formulation can give different results than models solved in versions earlier than 4.3b. Java files can be modified to retain old crosswind formulations. Please contact COMSOL Support for details.
- The default settings have changed for the Background Pressure Field in the Pressure Acoustics interfaces. Add the following line to obtain the old behavior of this feature:

```
model.physics("acpr").feature("bpf1").set("c", 1, "acpr.c_c");
```
- In the Acoustics Module, the Far-Field Calculation feature is no longer available in 1D and 1D axisymmetric models.
- The shape function property `border` has been deprecated and replaced by `order`.

COMSOL 4.3a API Changes

GENERAL API CHANGES

- Class files compiled with COMSOL 4.3 or earlier need to be recompiled if they contain code that changes the contents of `model.selection()`.

- The `solutionintitle` property governs whether to show solution information in plot titles. But since solution information has never been shown for 1D plots, regardless of the content of the property, the property has been removed from 1D plot groups.

FLUID FLOW INTERFACES API

- Fluid flow features in the CFD Module and Microfluidics Module that previously created a scalar Lagrange multiplier field with default name `model.un_1m` now create a vectorial Lagrange multiplier field with default name `model.u_1m`. The default component names in 3D are `model.u_1m`, `model.v_1m`, and `model.w_1m`, respectively. Java files must be updated accordingly.
- Weak constraints for the Interior Wall feature are no longer available. Any references to its weak constraint parameter (`weakConstraints`) or Lagrange multipliers must be removed.
- The **Automatic** setting for the **Pseudo time stepping** property now sets the variable `<phitag>.locCFL` to the built-in variable `CFLCMP`, which in turn triggers a PID regulator via the automatic solver suggestions. Here, `<phitag>` is the physics interface tag. Java files where pseudo time stepping is active and the local CFL number is set to **Automatic** must be modified by adding the command


```
model.physics(<tag>).prop("PseudoTimeProperty").set("CFLNumExpr", 1, "Manual");
```

 before calling a solver.

COMSOL 4.3 API Changes

- In the Acoustics Module, the far-field variables `pfar` and `Lp_far` have new names with full scope. They are now referred to as `acpr.ffc1.pfar` and `acpr.ffc1.Lp_far`, respectively. Any model files for Java that use the old variable names in expressions (such as expressions used for plotting or evaluation that include such old variable names) require a manual update.
- The following methods


```
model.physics(<tag>).feature(<ftag>).params();
model.physics(<tag>).prop(propname).params();
```

 are deprecated and replaced by the methods


```
model.physics(<tag>).feature(<ftag>).param();
model.physics(<tag>).prop(propname).param();
```

- Class files compiled with COMSOL versions 4.1 or earlier need to be recompiled.
- The far-field feature in the Electromagnetic Waves interface has changed from being a boundary feature with a boundary selection only, to a domain feature with a domain selection. It also has a subfeature—a boundary selection.
- The dependent variable associated with gauge fixing in the Magnetic Fields, Magnetic and Electric Fields, Electromagnetic Waves, and Transient Electromagnetic Waves interfaces is now unique to each interface. It is no longer available in the model scope, for example, `mod1.psi`. Instead the gauge fixing field is only available in the interface scope, for example, as `mod1.mf.psi`.
- In the scattered field formulation in the Electromagnetic Waves interface, the scattered field is no longer available in the model scope (for example, `mod1.re1Ex`). Instead the scattered field is only available in the interface scope as, for example, `mod1.emw.re1Ex`.
- In the Solid Mechanics interfaces (`SolidMechanics`, `PiezoelectricDevices`, `AcousticStructure`, `Poroelasticity`, `ThermallyInducedStresses`, `JouleHeatingThermalExpansion`, `TransientAcousticStructure`, `FluidStructureInteraction`), tensors in local coordinate systems (`e1`, `ee1`, `S1`, `si`, and `ei`) have new names. The coordinates of the local system (for example `x1`, `x2`, and `x3`) are no longer used. Instead 1, 2, and 3 are used together with double indexing for all components. As an example, `e1x2` is replaced by `e122` and `e1x2x3` is replaced by `e123`. The tensors `si` and `ei` are now called `Sil` and `eil`.
- In the Darcy's Law interface and the Richards' Equation interface in the Subsurface Flow Module, fluid compressibility is now a material parameter and no longer has a default value. If the default value was used, you now have to set the value. The following example sets the permeability to the old default value:

```
model.physics("d1").feature("smm1").set(chif_mat,userdef);
model.physics("d1").feature("smm1").set(kappa,4e-10);
```
- In the Poroelasticity interface in the Subsurface Flow Module, the fluid compressibility and the Biot-Willis coefficient are now material parameters and no longer have default values. If the default values were used you now have to set the value. The following example sets the permeability to the old default value:

```
model.physics("d1").feature("smm1").set(chif_mat,userdef);
model.physics("d1").feature("smm1").set(kappa,4e-10);
```

- The Level Set and Phase Field interfaces now include the Initial Interface feature by default. If you have a model that was created in an earlier version of COMSOL, it will fail to create a feature with the same tag name.

COMSOL 4.1 API Changes

The following changes were made to the COMSOL API between version 4.0a and 4.1:

- In version 4.0a the property `nonlin` in the stationary solver could have the values `auto`, `on`, `off`, and `linearized`. The default in most cases was `auto`, but in some cases, specifically in a frequency-domain study, the default was `linearized`. In version 4.1 the possible values are `auto`, `on`, `off`, and `linper`, with `auto` as default, except in the *Frequency Domain*, *Linearized* study step where the default is `linper`. The value `linearized` can still be set, but this is treated as `auto`. If the problem is linear and the linearization point is zero, `auto` should give the same solution as `linearized`. For nonlinear problems where the linearization point is zero, the solution given by `auto` might be slightly different, and probably more accurate, while changing to `off` should give exactly the same solution as `linearized`. When there is a nonzero linearization point, it is relevant to use the `linper` option. This option works like `linearized` except that source terms that contribute to the linearized problem must be enclosed by the `linper` operator.
- In the Darcy's Law interface, Brinkman interface, Richards' Equation interface, and Poroelasticity interface (in the CFD Module and Subsurface Flow Module), the permeability and porosity are now material parameters and no longer have default values. If the default values were used in a model file for Java, you now have to set them. The following example sets the permeability to the old default value:


```
model.physics("d1").feature("d1m1").set(kappa_mat,userdef);
model.physics("d1").feature("d1m1").set(kappa,3e-11);
```
- The pressure variable in solid mechanics, typically `solid.pw`, now only gets allocated degrees of freedom for incompressible materials. If you have referred to it, for example, during segregation in the solver, the code will have to be changed to remove the field.
- The solution to eigenvalue and eigenfrequency problems may now appear in a different order.
- The Batteries & Fuel Cells Module's boundary feature, `BoundaryReactionCoefficient`, is tagged by `rc` instead of `brc` by default. This means that files that explicitly refer to the old default tag name must be modified to refer to the new tag name.

COMSOL 4.0a Java API Changes

The following changes were made to the COMSOL API between version 4.0 and 4.0a:

- The load face parameter (Fph) of the *Phase* feature in `AcousticStructure`, `TransientAcousticStructure`, `ThermallyInducedStresses`, `SolidMechanics`, `JouleHeatingThermalExpansion`, `Poroelasticity`, and `FluidStructureInteraction` has changed from degrees to radians.
- The `physics.field()` operator for `Beam` and `Truss` are not backward compatible with 4.0. Scalar fields are now vector fields.
- The variables for strain and stress in the `Shell` interface have been renamed. Variable names that began with a lowercase `s` now begin with an uppercase `S`. For variables names that ended in lowercase `l` (`L`), the letter was moved to the third position. For example, `emXXl` is now `emlXX`.
- Force loads on solids and trusses have been moved from the material to the spatial frame. This means that the index notation of some variables has changed from uppercase `XYZ` to lowercase `xyz`.
- The acoustics interfaces (`PressureAcoustics`, `TransientPressureAcoustics`, `AcousticStructure`, and `TransientAcousticStructure`) have been moved from the material to the spatial frame. This means that the index notation of vector variables has changed from uppercase `XYZ` to lowercase `xyz`.

This concludes the release notes for COMSOL Multiphysics version 5.1.