COMSOL Multiphysics

Release Notes

5.2a

COMSOL

COMSOL Multiphysics Release Notes

Version: COMSOL 5.2a

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
- COMSOL Blog: www.comsol.com/blogs
- 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

Contents

Chapter 1: Release Notes

I

Release Notes

COMSOL Multiphysics® version 5.2a includes extended and improved versions of the Application Builder and the COMSOL Server™, as well as updates and improvements for all COMSOL Multiphysics® products. These *Release Notes* provide information regarding new functionality in version 5.2a for all COMSOL products.

COMSOL Multiphysics

General New Functionality

In Windows, the former Help Desk item on the Start menu is now called COMSOL Documentation and opens a standalone COMSOL Documentation window in a separate process.

New Functionality in the Application Builder

- A new **Application Builder** page is available in the **Preferences** dialog box. It contains a new preference to limit the maximum number of open editors. The **Desktop windows** section, previously located under General, has also been moved to the **Application Builder** page.
- There is now support for switching units in applications. The new **Unit Set** node under Declarations makes it possible to specify a list of values and display names, where each value represents a specific unit in a list of units. The Unit Set also has a table of different unit lists, and each list contains units with the same dimension. The Input Field and Slider form objects can use a specific unit list of a unit set to specify the display unit of the form object.
- You can now create shortcuts to form objects and menu, ribbon, and toolbar items. In the Settings windows for such objects, a **Create Shortcut** button is available. You can also use the keyboard shortcut Ctrl+K to create a shortcut declaration to the currently selected form object or menu item. Created shortcuts appear in the **Shortcuts** node under **Declarations** in the Application Builder tree. Shortcuts are automatically updated when objects are renamed, moved, copied, and duplicated. Shortcuts are available in application method code as read-only Java® variables.
- Most form objects and items for menus, ribbons, and toolbars have two new settings: Visible and Enabled. These settings determine the initial visibility and enabled states of a form object or item. Therefore, it is no longer necessary to use a startup event and the a `setFormObjectEnabled` and `setFormObjectVisible` methods to control the initial visibility and enabled states.
- An **About Dialog** section has been added to the **Main Window** node's **Settings** window. You can now customize the placement of the open About Application dialog action, choose to hide the default COMSOL Server information, and add a customized about text in the dialog box.

- When creating a command sequence for Animations with Target set to Player, an additional argument can now specify which graphics to draw the animation in.
- Events can now use explicit selections as a trigger source. The enlisted events are triggered when the selected entities of the source selection changes.
- Everything in methods and command sequences that affect the state of form objects now apply immediately and not after the method or command sequence has finished. It is possible to change a text label or switch a card in a card stack in a method and see the effect immediately, even if the method continues with some lengthy operation, such as solving. The only exception to this is generating a report. It is still done after the method or command sequence has finished. As a side effect, the `updateGraphics` method is now obsolete, and the corresponding command for command sequences is no longer available.

IMPROVEMENTS TO THE FORM EDITOR

- A new Hyperlink form object can be used for adding hyperlinks to an app.
- A new Video form object can be used to show a video of an embedded video file in apps.
- When showing a selection in a Graphics form object, the toolbar now includes the following buttons: **Zoom to Selection**, **Select Box**, and **Deselect Box**.
- The Progress Bar now has a setting to close dialog forms when the cancel button is clicked.
- Similar to the current Main Menu, Toolbar and Ribbon items, a new Toggle item is available, with toggle functionality.
- The minimum and maximum limits for Input Fields and Slider objects can now support global parameters in their expression. They also dynamically change when the parameter changes.
- Graphics form objects have a new Zoom to extents on first plot setting option. This setting makes the first plot that appears in the graphics canvas zoom to extents. This happens either when the app starts, or if the canvas is empty initially when something is plotted the first time.
- Font sizes assigned to form objects are now given in points (pt) and not pixels (px).

IMPROVEMENTS TO THE METHOD EDITOR AND APPLICATION LANGUAGE

- The form objects can now be modified from methods (see the *Application Programming Guide* for more information).

- A new document, the *Application Programming Guide*, available as a PDF, explains the structure and API for programming methods in the Application Builder and provides comprehensive information about available properties for form objects and other parts of the Application Builder that you can access from methods.
- There are new shortcut keys for brackets in code. Ctrl+M will jump the caret between matching brackets (parenthesis, square brackets, or curly braces) and Ctrl+Shift+M will additionally select the entire bracket range.
- A `toString` method with the option to remove trailing zeros has been added.
- A `getLicenseNumber` method retrieves the license number for the current session as a string.

New Functionality in COMSOL Server

- A new Power user permission role has been added, which is an intermediate role between a User and an Administrator. A Power user is a moderator of a set of groups and the members therein. These groups are selected by an Administrator or created by the Power user. A Power user is allowed to see the Monitor, User database, and Preference pages; edit the users in their moderated groups; and shut down the group's running apps.
- A new preference setting, Prelaunched application processes, has been added to the **Prelaunching** section on the **Preferences** page. This allows an administrator to manually set the number of prelaunched application processes that should be available for launching applications. The default number of prelaunched application processes is 1.
- Administrators can change the look and feel of COMSOL Server via a new Appearance section, found under Administration. Available customizations include colors for backgrounds and text, logos, login screen content, and custom HTML.
- Administrators and group moderators can now select an application to launch on login based on the group. There is also the ability to override the global launch on login settings. It is possible to select an app that is different from the global app to launch on login by group or disable launching on login for a specific group.
- The Monitor page has been improved with the possibility to show and hide parts of the information.
- There are general look-and-feel improvements to the COMSOL Server user interface.

New and Improved General Functionality in COMSOL Multiphysics

- A new **Add Multiphysics** window provides a list of available predefined multiphysics couplings for the selected physics interfaces (previously available only in the Model Wizard). Adding such a multiphysics coupling to a model components adds the corresponding multiphysics coupling nodes under the **Multiphysics** node. The Add Multiphysics window makes it easier to create multiphysics models incrementally.
- In parameters, variables, and most applicable settings in the physics interfaces, you can type Ctrl+Space to choose from a list of available and applicable parameters and variables (or start typing text for a list of possible variables). This functionality was previously only available for postprocessing. You can also click the **Insert Expression** button underneath tables of parameters and variables.
- You can now edit expressions in the **Equation View** nodes' Settings window for the main physics interface nodes and for coordinate system nodes.
- When saving COMSOL Multiphysics application files, you can choose to optimize for speed or file size when saving files. The speed option uses uncompressed MPH files.
- The Find tool for finding variables and expressions now also works for Equation View.
- You can now click on the column headers in the Variables table in an Equation View node to sort the table alphabetically.
- The **Range** dialog box now includes an option to create a logarithmic range, for example, for a frequency range.
- It is now possible to copy and duplicate **View** nodes in the model tree.
- The Duplicate action can now be performed using the keyboard shortcut Ctrl+Shift+D.
- With applicable time-explicit physics interfaces in the Acoustics Module, RF Module, and Wave Optics Module, a new **Absorbing Layer** node is available under Definitions. It adds an absorbing layer (sponge layer) to a time-explicit simulation. The absorbing layers work by combining three techniques: a scaling system, filtering, and simple nonreflecting conditions.
- It is now possible to override the automatic mapping of the orientation of the source and destination for periodic conditions. With this option, you can create more general periodic conditions.
- You can now double-click to add a geometric entity from the Selection List to a list of selected entities in a Settings window.

- Support for general numerical flux has been added to the Wave Form PDE interface.
- For models in user libraries added in the Application Libraries window, the **Open PDF Document** button will be available if there is a PDF file with the same name as the MPH file in the same directory.

New Geometry and Mesh Functionality

GEOMETRY FUNCTIONALITY

- A Partition Faces operation is now available in 3D geometries for partitioning faces using curve segments between vertices, extended edges, or a work plane.
- There is improved interactive drawing for 2D geometries, including coordinate display and snapping points indicated by a red circle. You can now also snap to the geometry for side midpoints and centerpoints of other geometry objects.
- A **Show in 3D** list is now available for geometry features in a plane geometry in a work plane. The selections will appear as selections for use within the 3D geometry sequence. In the **Work Plane** node's settings, you can choose to show selection from the plane geometry in the physics (or in instances for geometry parts).
- The geometry repair tolerance can now be specified as automatic, relative, or absolute.

MESHING FUNCTIONALITY

- For general unstructured meshing, several improvements are available:
    - In the **Tessellation** section, the **Method** list now has the options **Automatic**, **Delaunay**, and **Delaunay (legacy version)**. Automatic is the default in new models and means that COMSOL Multiphysics tries to determine the best algorithm to use for each domain. Delaunay is a more efficient implementation of the meshing algorithm that was introduced in version 5.2 (and was then called the alternative version). Delaunay (legacy version) is the algorithm used in earlier COMSOL versions. If you open an old model, it will use the legacy version by default.
    - In a new section, **Element Quality Optimization**, you can control how much effort COMSOL Multiphysics will put into optimizing element quality and tune the optimization for certain situations. The **Optimization level** setting can be **Basic** (the default), which makes basic optimizations aiming at a minimal element quality of 0.2; **Medium**, which makes more optimization and aims at a minimal element quality of 0.35; or **High**, which attempts all available optimization operations. If

the quality of the surface mesh is low (typically due to small details or narrow corners in the geometry), the high setting can take a significant amount of time.

  - If your geometry has fillets or other curved regions with a relatively coarse mesh, and you solve with a geometry shape order higher than one, you can select the Avoid inverted curved elements check box. This makes the optimization try to reduce the number of element that invert when they are curved but may also result in longer meshing times and often a slightly higher number of elements and lower element quality.
  - If your computation is sensitive to too large mesh elements, you can select the **Avoid too large elements** check box. For each element, there is a desired element size h, and if the element is larger than that, the COMSOL Multiphysics software tries to make it smaller. The cost for this operation is meshing time and element quality. If you evaluate the maximum of h on a sufficiently large mesh of uniform size, this value is typically decreased by 10 percent when this option is selected.

- For an explicit distribution of elements option in the **Distribution** node in meshing sequences, you can now choose to reverse the direction.
- The quality and robustness of the free quad mesh algorithm has been improved for circle-shaped domains in 2D and planar circle-shaped faces in 3D.
- In the **Mesh** node's **Statistics** window, the number of mesh vertices are now displayed.
- Progress information is now available during a mesh copy operation.

New and Updated Operators, Functions, and Definitions

- For Square and Triangle functions, the duty cycle can be set. The default value is 0.5. The duty cycle is the fraction of a period in which the function has the high value (for Square waveforms) or for which the function is rising (for Triangle waveforms).
- If you need to create several random functions that return independent random values, you can now select a **Use random seed** check box and make sure that the functions have different random seeds.
- You can use `'first'` or `'last'` as the first argument to the `at` and `with` operators to evaluate an expression at the first or last time of the simulation, respectively.
- Serendipity elements (rectangular shape elements with no interior nodes) are now available for the PDE interfaces and for structural mechanics. For hexahedral

elements, serendipity elements can significantly reduce the degrees of freedom while retaining a similar simulation accuracy.

- New dimensionless units: `[%]` or `[percent]` for percent (%) and `[ppm]` or `[partspermillion]` for parts per million (ppm).
- The extrapolation of unstructured interpolation function has been improved. The extrapolation takes place outside (an approximation) of the convex hull of the point cloud specified by the interpolation data points. Inside the convex hull, the evaluation is typically unchanged compared to earlier versions.

New Functionality in Studies and Solvers

STUDIES AND STUDY STEP FUNCTIONALITY

- The PML, Infinite Elements, and Scaling System nodes can now be enabled and disabled from a study step.
- The study step settings now include a list of multiphysics couplings in the study.

SOLVER FUNCTIONALITY

- The new smooth aggregations AMG solver is a multigrid solver that can handle a wider variety of problems than the classic AMG solver, such as saddle point forms of the Stokes and Navier-Stokes equations.
- Old solution settings are now kept in results features when clearing the solution.
- Initial conditions that depend on a parameter now respect the auxiliary sweep through an automatic synchronization of values of parameters to use for initial expressions. It also possible to pass local constants to solvers.
- Zero padding of input data is now available for the forward FFT solver. There are also new predefined variables that are available for user-defined window functions using an expression.
- The Domain Decomposition solver has been improved with better performance and support for restricted Schwarz methods. You can also use the new smooth aggregations AMG solver as the multigrid solver in domain decomposition.
- A new option makes it possible to name exported files by means of an index instead of parameter names and values. You can use this option for shorter filenames when the parameter names are long and many in a parametric sweep, for example.
- The MUMPS solver has been upgraded to version 5.0.1 with improved OpenMP support and a number of bug fixes.

- Support for symmetric dense matrices in the dense matrix format provides speed and memory improvement to applications using BEM.
- The frequency-domain modal solver now includes non-homogeneous Dirichlet conditions when performing a matrix export.
- The performance of the ODE integration has been improved.
- The job scheduler now shows progress information.
- A preference setting for the number of sockets in a multicore computer has been added.
- The `comsol batch` command supports two new options: `-paramfile` and `-error`.

New and Improved Results and Visualization Functionality

- There are new options for color tables, including: AuroraAustralis, AuroraBorealis, HeatCamera, HeatCameraLight, JupiterAuroraBorealis, and Twilight.
- It is now possible to evaluate several expressions in one single node under Derived Values of the following types: Global Evaluation, Point Evaluation, Average, Integration, Maximum, and Minimum.
- There is improved smoothing of plots, including:
  - Smoothing inside material domains.
  - Smoothing with respect to an expression.
  - A relative smoothing threshold is also available. No smoothing is performed where the discontinuity is greater than the threshold.

  These settings are available in the **Quality** section of the plot's Settings windows.
- Filled legends in contour and isosurface plots.
- A **Parameters** node is now available under **Results** for defining local parameters used in postprocessing.
- Plot annotations are now available in 1D plots and with extended formatting options, including LaTeX markup and the possibility to add a colored background rectangle for better contrast.
- Improved table surface plots are more flexible regarding the data formats they handle and support discrete plots in addition to continuous plots. A discrete plot with a height attribute results in a "skyscraper-like" plot.
- There is improved transparency, with a more reliable performance for a range of computers and graphics cards.

- Streamline plots can now allow for discontinuous streamlines at material boundaries by including the same smoothing settings as other plot types.
- There are more flexible plot titles, which support evaluating expressions and including the filename of the MPH file.
- Prefix and suffix strings can now be added in Automatic mode for graph plot legends.
- You can now control the axis and grid lines independently when exporting images from 1D plots.
- You can now save data from an **Export>Data** node in an unstructured VTK file (*.vtu).
- The default resolution for new plots can now be specified on the **Results** page in the **Preferences** dialog box.
- Two new options in the **Transformation** list in the **Global Matrix Evaluation** feature makes it possible to convert from a Maxwell capacitance matrix to a mutual capacitance matrix and vice versa.
- For reports in the Microsoft Word format, there is now support for custom themes.
- Mass Properties and Mass Contributions are now included in reports.
- You can now use global expressions that depend on the solution in many more places than before: The most important places are cut line positions, cut plane positions, arrow positions, and contour and isosurface levels.

General Backward Compatibility Considerations

COMSOL 5.2a is backward compatible with COMSOL versions 4.0–5.2. To open models created using COMSOL version 3.5a in version 5.2a, follow these steps:

1. Open and 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 5.2a.

Backward Compatibility with Version 5.2

FREE QUAD MESH

The new free quad meshing algorithm is used in new models, but for models created in earlier versions, the legacy algorithm is used instead. The default for the new `method` property is therefore `auto` in new models, and `legacy52` in migrated models.

GEOMINFO CHECK() METHOD

The return value of the `check()` method in GeomInfo has been removed. Now, `check()` throws an error if the geometry is invalid.

THE CONST PROPERTY IN SOLVERS

The `const` property available for many solvers has been removed in version 5.2a and replaced with the `cname` and `clist` properties, which are string arrays for the constant names and corresponding constant values, respectively. For the **Dependent Variables** node, the new property `initparametersmethod`, which can be set to `"auto"` (the default) or `"manual"`, has been added to control automatic synchronization of values of parameters to use for initial expressions. The **Dependent Variables** node now also includes the `cname` and `clist` properties.

Backward Compatibility with Version 5.1

CREATING SELECTIONS

In version 5.2, the `selresult` property replaced the `createselection` property. `createselection` is still supported for backward compatibility.

SELECTIONS IN PART INSTANCES

For backward compatibility for selections in part instances, there is a **Keep noncontributing selections** check box, cleared by default, in the **Selection Settings** section of the Settings window for **Part Instance** nodes. If you select the **Keep noncontributing selections** check box, the **Keep column** is disabled, and the selection is kept if the **Contribute to** value is **None**. In the COMSOL API, the default is an active Keep noncontributing selections setting.

MESH PARTS

For backward compatibility regarding STL/VRML import, and for COMSOL API compatibility, the old user interface for mesh import (with a specified file name and import properties) and the corresponding properties are available in the COMSOL API but not in the COMSOL Desktop, unless it is the active option. This means that the old user interface can only be reached if the filename is set in combination with the STL import type (the API type `stlvrml`), which cannot be done from the new user interface.

DORMAND-PRINCE 5 TIME-STEPPING SOLVER

The Dormand-Prince 5 Runge-Kutta solver in 5.1 does not use field norm scaling in version 5.1 when estimating errors. This means that the errors of, for instance, an ODE might become very small when solved together with a field with many DOFs, and the time steps taken might then be too large. In version 5.2a, the Runge-Kutta solver uses field norm scaling, which means that old models using the Dormand-Prince 5 might need tighter tolerances to produce results in earlier versions that are similar to those in 5.2a.

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 5.2a, 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 is 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.

MERGED MPH-FILE FORMAT

The MPHAPP-file and MPH-file formats have been merged since version 5.1, and all application files use the `.mph` file suffix. You can still open MPHAPP-files created in version 5.0.

Backward Compatibility with Version 4.4

COMSOL SERVER

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 setting 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 5.2a.

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 interface 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, the help variable `u0` (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 and 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 Multiphysics if you regenerate the solver sequence in 5.2a.

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 Multiphysics include a manifest file that contains a reference to the CDO library, which has been removed. To load such JAR files in version 5.2a, 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 this 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 first clicking on C2 may highlight C1. If you click for a second time without moving the cursor, it will highlight 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 5.2a:

- *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 have 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 of 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_lm` until the model is re-solved. In some cases, occurrences of `un_lm` in the solver sequence must be replaced manually. This is the case if `un_lm` 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 5.2a for postprocessing, or opened and re-saved in version 4.3 before being opened in version 5.2a.

AUTOMATICALLY CREATED IDENTITY/CONTACT PAIRS

The pairs created by the Form Assembly feature in the geometry can differ from their forms 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 5.2a

HYSTERESIS JILES-ATHERTON MATERIAL MODEL

The Jiles-Atherton material model for hysteresis captures important properties of ferromagnetic materials and allows for realistic modeling of devices like transformer cores, electrical motors, etc. It is available in the Magnetic Fields interface; the Magnetic Fields, No Currents interface; and the Rotating Machinery, Magnetic interface. It supports fully anisotropic (vector) hysteresis modeling.

MAGNETIC SHIELDING WITH SATURATION EFFECTS

The Magnetic Shielding boundary condition now supports magnetic saturation in the Magnetic Fields interface; the Magnetic Fields, No Currents interface; and the Magnetic and Electric Fields interface. The effect is important when designing thin, high-permeability shields for sensitive electronics, for example, photomultiplier tubes. Such shields easily saturate and, above the saturation limit, the shielding efficiency drops substantially.

MERGED, UPDATED, AND ENHANCED COIL FEATURES

The **Single-Turn Coil** and **Multi-Turn Coil** features in the Magnetic Fields interface and the Magnetic and Electric Fields interface have been merged into a single **Coil** feature. It provides a more streamlined workflow and better flexibility:

- The Coil Geometry Analysis preprocessing step can now handle single-conductor (old Single-Turn) 3D coils. This allows for the modeling of arbitrary shaped conductor as sources of excitation for the magnetic interfaces, with better convergence properties than the old Single-Turn Coil features.
- Coil Geometry Analysis now supports boundary coils in addition to domain coils.
- Single-conductor coils with voltage excitation can now be solved in time-dependent studies (in the Magnetic Fields interface).
- Single-conductor coils now apply their excitation as an external electric field, giving physically meaningful electric fields in the entire geometry.

DOMAIN TERMINAL

The **Terminal** feature in the Electric Currents and Electrostatics interface is now also available on the domain level. This is convenient for more geometrically complex electrodes that would involve the selection of a large number of boundaries when using a terminal at the boundary level. The unknowns for the electric potential inside the terminal's domain selection are not solved for but replaced by a variable. This is useful when modeling electrodes with a finite thickness that is respected by the geometry.

New Application in Version 5.2a

VECTOR HYSTERESIS MODELING

In this application, a 3D simulation employing the Jiles-Atherton vector hysteresis model is used to solve case 3 for the TEAM problem 32 published by the International Compumag Society. This case is especially designed for testing vector hysteresis modeling. It consists of a three-limbed ferromagnetic core and two coils placed on its external limbs. The coils are excited by AC voltage sources that are phase shifted by 90 degrees with respect to each other. This creates a magnetic field that is rotating in some regions of the core. A vector hysteresis model is necessary to accurately simulate the time-dependent field.

Backward Compatibility with Version 5.2 and older

COIL FEATURES

Multi-Turn Coil features in old models will be mapped to the new **Coil** feature using the Homogenized Multi-Turn Conductor model. **Single-Turn Coil** features in old models will import as is with a warning that the feature is obsolete and will be removed in future versions.

The **User Defined Coil Geometry** subnode, which is available for 3D **Coil** nodes with the User-Defined Coil type, is now a domain feature. You can now select coil input and output boundaries using the new **Input** and **Output** subnodes. The selection is set up correctly when opening old models.

REMOVED THE OLD INFINITE ELEMENTS FEATURES

The old **Infinite Elements** feature, which are obsolete since version 4.2, have now been discontinued and will be automatically removed when opening old models.

Backward Compatibility with Version 5.0 and older

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 of 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("stcd1").create("hp1",
   "CoilHarmonicPerturbation", -1);
```

Global features have no selections, so code that accesses 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 Physics Interfaces in Version 5.2a

CONVECTED WAVE EQUATION, TIME EXPLICIT

The Convected Wave Equation, Time Explicit is a completely new interface for modeling large acoustic problems in the time domain. The interface is based on the discontinuous Galerkin method (also known as DG-FEM or simply DG) and uses a time-explicit solver. This results in a very memory-efficient method.

The Convected Wave Equation, Time Explicit interface is found under the new Ultrasound subbranch. The interface is used to solve large transient linear acoustic problems containing many wavelengths in a stationary background flow. Application areas include ultrasound flow meters, ultrasound distance sensors, and other ultrasound sensors where time of flight is an important parameter. The applications are not restricted to ultrasound, but also include, for example, transient propagation of audio pulses in room acoustics or car cabins.

The interface is suited for time-dependent simulations with arbitrary time-dependent sources and fields. In general, it is suited for modeling the propagation of acoustic signals over large distances relative to the wavelength; linear ultrasound problems are example applications. The interface includes new **Absorbing Layer** nodes that are used to set up effective nonreflecting-like boundary conditions (sponge layers). The interface solves the linearized Euler equations assuming an adiabatic equation of state. The dependent variables are the acoustic pressure and the acoustic velocity perturbations. The background flow can be any stationary flow with small to moderate velocity gradients. No loss mechanisms are included in the interface. The physics interface exists in 3D and 2D geometries.

THERMOVISCOUS ACOUSTICS, BOUNDARY MODE

The Thermoviscous Acoustics, Boundary Mode interface is used to compute and identify propagating and nonpropagating modes in waveguides and ducts. The interface performs a boundary mode analysis on a boundary, inlet, or cross section of a waveguide or duct of small dimensions. The thermal and viscous loss effects that are important in the acoustic boundary layer near walls are included.

The interface solves for the acoustic variations in pressure, velocity, and temperature, as well as the out-of-plane wave number of the modes. As mentioned, near walls, viscous losses and thermal conduction become important because a boundary layer exists. The thickness of these layers is known as the viscous and thermal penetration depth. For this reason, it is necessary to include thermal conduction effects and viscous losses explicitly in the governing equations. The Thermoviscous Acoustics, Boundary Mode interface is, for example, used when setting up sources in systems with small ducts like hearing aids or mobile devices. It can also be used to identify the propagating wave number and characteristic impedance of a duct cross section and use that information in the homogenized **Narrow Region Acoustics** feature of the Pressure Acoustics, Frequency Domain Interface.

The physics interface exists in 3D and 2D axisymmetric geometries and is applied at boundaries. It solves the equations defined by the linearized Navier-Stokes equations (linearized continuity, momentum, and energy equations), in quiescent background conditions, searching for the out-of-plane wave numbers at a given frequency.

THERMOVISCOUS ACOUSTICS: NEW NAME FOR THERMOACOUSTICS

In this release of COMSOL Multiphysics, all interfaces that used the Thermoacoustics term have been renamed to use Thermoviscous Acoustics instead. The interfaces have always been dedicated for detailed modeling of thermal and viscous acoustic losses in problems with small geometrical dimensions. That is, in problems where the losses in the thermal and viscous acoustics boundary layers are important. This is the case when modeling microphones, mobile devices, hearing aids, miniature transducers, and much more. Since **Thermoacoustics** is already a branch of acoustics, dealing with cooling or heating using acoustic waves, a more descriptive and correct terminology is now used: **Thermoviscous Acoustics**.

The following interfaces have new names:

- Thermoviscous Acoustics, Frequency Domain
- Thermoviscous Acoustics, Boundary Mode (new interface)
- Acoustic-Thermoviscous Acoustic Interaction, Frequency Domain
- Thermoviscous Acoustic-Structure Interaction, Frequency Domain

New Functionality in Version 5.2a

BACKGROUND ACOUSTIC FIELDS IN THERMOVISCOUS ACOUSTICS (SCATTERED FIELD FORMULATION)

The Thermoviscous Acoustics interfaces are now defined in the so-called scattered field formulation. This enables users to add a background acoustic field to a model. The background field can be either User Defined or Plane Wave. The Plane Wave option defines a plane traveling wave with physically consistent viscous and thermal attenuation.

Using this new feature it is possible, for example, to create simple sources when modeling transmission problems where thermal and viscous losses are important or to study scattering of small objects (small as compared to the acoustic boundary layers). As an advanced application example, the feature can be used in conjunction with the Thermoviscous Acoustics, Boundary Model interface to create sources at the inlet of waveguides.

DIRECTIVITY PLOT

The new Directivity plot is a dedicated acoustics plot for the analysis of loudspeakers and other electroacoustic transducers. The plot allows audio engineers to depict the spatial response of a loudspeaker as a function of both frequency and spatial angle in a contour-like plot. Representing the spatial response in this manner is very common in the loudspeaker industry. Measured data is often represented in the same manner. The plot includes many options for formatting to achieve maximal insight into the modeled data. Some of the key formatting features are:

- Normalization: The input sound pressure level data can be normalized either with respect to a specific polar angle, with respect to the maximal value (at each frequency), or have no normalization.
- Evaluation: The evaluation circle can be defined at any location in space and it is possible to define a reference direction that sets the 0 deg. direction.
- Coloring and style: Format the data as filled surfaces, lines, and add labels. It is also simple to switch the layout of the axis and have the frequency on either the *x*- or *y*-axis.

BACKGROUND ACOUSTIC FIELDS IN LINEARIZED NAVIER-STOKES (SCATTERED FIELD FORMULATION)

The Linearized Navier-Stokes, Frequency Domain interface is now defined in a so-called scattered field formulation. This enables users to add a background acoustic field to a model. The background field can be entered as user-defined expressions for the pressure, acoustic velocity, and temperature variation. This can be an analytical expression defining a certain type of wave, but it can also be the solution of another acoustic model. An applications example could be to create simple sources when modeling transmission problems where thermal and viscous losses are important.

BACKGROUND ACOUSTIC FIELDS IN LINEARIZED EULER (SCATTERED FIELD FORMULATION)

The Linearized Euler, Frequency Domain interface is now defined in a so-called scattered field formulation. This enables users to add a background acoustic field to a model. The background field can be entered as user-defined expressions for the pressure, acoustic velocity, and temperature variation. This can be an analytical expression defining a certain type of wave, but it can also be the solution of another acoustic model. An applications example could be to create simple sources when modeling transmission problems where thermal and viscous losses are important.

The feature was present in the previous version of the Acoustics Module but the name has been changed from **Incident Acoustics Fields** to **Background Acoustics Fields**. The feature has been updated and improved.

RAY ACOUSTICS: COMPUTE RAY POWER AND SURFACE SOUND PRESSURE LEVEL (SPL)

New ray power computation functionality has been added to the Ray Acoustics interface. This means that four intensity computation options exist:

- Compute intensity
- Compute intensity and power
- Compute intensity in graded media
- Compute intensity and power in graded media

The options that existed in version 5.2 have been renamed. *Using principal curvatures* is now *Compute intensity*, and *Using curvature tensor* is now *Compute intensity in graded media*. Once one of the options that calculate the ray power is selected, it is possible to add a **Sound Pressure Level Calculation** subnode to the Wall conditions. This new feature sets up a predefined accumulator that calculates the surface sound pressure level, including the effects of surface properties, such as the absorption coefficient. The variables can easily be plotted in postprocessing.

RAY ACOUSTICS: MESHLESS RAY TRACING

Ray tracing no longer requires a computational domain mesh if the medium has homogeneous properties (nongraded media). In this case, the material parameters can be defined globally. The only requirement is that a model includes at least one boundary condition, such as a wall or material discontinuity. Rays can thus propagate over long distances in geometries that do not need to have their domains meshed. Rays can even be released and propagate outside the geometry. This could be used in underwater acoustics applications or in large concert hall simulations.

SPHERICAL AND CYLINDRICAL WAVES IN BACKGROUND PRESSURE FIELD AND INCIDENT FIELDS FOR PRESSURE ACOUSTICS

In the Pressure Acoustics, Frequency Domain interface, the **Background Pressure Field** and **Incident Pressure Field** (a subnode to the radiation conditions) features have been extended to include cylindrical and spherical waves. This makes it possible to set up more complex incident or background acoustic field more easily. Fields that are generated by exterior point sources or small vibrating bodies can be approximated by a monopole type source.

ELECTRO-ACOUSTIC COUPLINGS FOR LOUDSPEAKERS

In the AC/DC Module, new features have been added and improved that simplify and extend the electromagnetic couplings in transducers with, for example, coils. This is highly relevant when modeling loudspeaker drivers. The **Multi-Turn Coil** domain feature now supports the Velocity (Lorentz Terms) in 3D and in 2D axisymmetry. On the structural mechanics side, the Lorentz force contribution can be picked up automatically as a Body Load. This is showcased in the Loudspeaker Driver model in the Application Library.

LOGARITHMIC AND ISO PREFERRED FREQUENCIES OPTIONS FOR DEFINING FREQUENCIES IN THE STUDY SETTINGS

Two new entry methods have been added when selecting frequencies in the study:

- Logarithmic: A start and stop (frequency) and a number of frequencies per decade can be entered (in all products).
- Logarithmic and ISO preferred frequencies: This option is only available with the Acoustics Module and is activated by enabling the Advanced Study Options. For the ISO preferred frequencies, you can select a start and stop frequency and the desired interval (Octave, 1/3 Octave, 1/6 Octave, 1/12 Octave, and 1/24 Octave). The 1/3 octave preferred frequencies are defined based on ISO 266. The standard option is extended to frequencies based on the preferred numbers of ISO 3 (series R20, R40, and R80) to define 1/6 Octave, 1/12 Octave, and 1/24 Octave intervals.

UPDATES TO THE PERFECTLY MATCHED LAYERS

For users working with perfectly matched layers (PMLs) that want to perform advanced customization of the layer properties, several options have been added:

- Enable and disable PMLs in the solver. This is useful if modeling scattering problems where the source is a computed field.
- User-defined geometry types, which are available if the PML has a nonstandard geometry. This option can also be used if the automatic PML geometry detection fails (for example, for some imported CAD geometries).
- User-defined coordinate stretching functions for defining the scaling. This allows advanced users to tailor the scaling inside the PML, for example, to very efficiently absorb waves in specific physics configurations.

OTHER ENHANCEMENTS AND IMPORTANT BUG FIXES

- Updated the Far-Field plots with a new option to specify the reference direction that defines the 0-degree angle.
- A new option to calculate temperature and two new boundary conditions in the Compressible Potential Flow interface.
- Compressible Potential Flow has two new boundary conditions: Interior Wall (Slip Velocity) and Mean Flow Velocity Potential. Moreover, there is now an option to automatically calculate the temperature field in the flow; this option can be used when setting up the background flow for a linearized Potential Flow Model.
- Two new thermal boundary conditions have also been added to the Thermoviscous Acoustics, Frequency Domain interface:
  - A Heat Flux condition for when a pulsing heat source generates acoustic waves.
  - An Interior Temperature Variation condition for modeling, for example, a thin plate with a harmonic temperature variation possibly generated by resistive heating from a pulsing current.
- As a consequence of renaming Incident Acoustics Fields to now be called Background Acoustics Fields in the Linearized Euler interfaces, all variables have `_i` changed to `_b`.
- Time derivative variables have changed names in the Linearized Navier-Stokes and Linearized Euler interfaces. Names changed from: `p_t` to `pt`, `u_t` to `ut`, `rho_t` to `rhot`, and `T_t` to `Tt`.
- It is now possible to add two contributing point sources in the Acoustic Diffusion Equation interface.

New Models in the Application Library in Version 5.2a

GAUSSIAN PULSE IN 2D UNIFORM FLOW: CONVECTED WAVE EQUATION AND ABSORBING LAYERS

This tutorial model illustrates the use of the Convected Wave Equation interface to model the propagation of a transient Gaussian pulse in a 2D uniform flow. The model domain is truncated using the built-in absorbing layers. The results are compared to an analytical solution.

TRANSFER IMPEDANCE OF A PERFORATE

This model determines the acoustic properties of a perforate (a plate with small holes). Specifically, the transfer impedance, surface normal impedance, and absorption coefficients are determined as function of frequency. The transfer impedance is compared to a semi-analytical model.

The holes are located in an *xy*-matrix with given parameterized *x* and *y* distances. When the hole radius is small or the frequency is low, thermal and viscous losses need to be included in the model. This is achieved by using the Thermoviscous Acoustics, Frequency Domain interface.

ULTRASOUND FLOW METER WITH GENERIC TIME-OF-FLIGHT CONFIGURATION

This model illustrates how to simulate a generic wetted transient time-of-flight ultrasound flow meter in COMSOL Multiphysics. Wetted ultrasound flow meters have a dedicated signal tube for the ultrasound signal, and the whole device is therefore mounted into the tubing where the flow is measured. Two signals simultaneously traverse the flow upstream and downstream and use their difference in arrival times to estimate the speed of the main flow.

The model first uses the CFD Module to calculate the steady-state background flow in the flow meter. The Convected Wave Equation, Time Explicit interface is then used to model the ultrasound signal propagating in the device. The interface is well suited for modeling linear ultrasound problems, including many wavelengths, as it is very memory efficient.

This example requires the Acoustics Module and the CFD Module.

New Models and Apps in the Application Gallery in Version 5.2a

Several Acoustics Module models have been added to the Application Gallery (www.comsol.com/models/acoustics-module):

ACOUSTIC STREAMING IN A MICROCHANNEL CROSS SECTION

Recent advances in the fabrication of microfluidic systems require handling of live cells and other microparticles as well as mixing. This can, for example, be achieved using acoustic radiation forces and the viscous drag from the streaming flow.

- Streaming flow: Due to the nonlinear terms in the Navier-Stokes equations, harmonic perturbation of the flow will lead to a net time-averaged flow called acoustic streaming. Acoustic streaming is a second-order (nonlinear) acoustic effect. The effect can be simulated in two ways: direct simulation solving the nonlinear Navier-Stokes equations, or by the separation of time scales, as shown here.
- Radiation force: Due to nonlinear terms in the governing equations, momentum can be transferred from an acoustic field to particles. This results in a net force acting on the particles — the acoustic radiation force.

The trajectory of particles in devices will be governed by the balance between the viscous drag force (from the streaming flow) and the acoustic radiation force. This model shows how to include and model both using COMSOL Multiphysics.

POROELASTIC WAVES WITH THERMAL AND VISCOUS LOSSES (BIOT-ALLARD MODEL)

In applications where pressure waves and elastic waves propagate in porous materials filled with air, both thermal and viscous losses are important. This is typically the case in insulation materials for room acoustics or lining materials in car cabins. Another example is porous materials in mufflers in the automotive industry. In many cases, these materials can be modeled using the poroacoustic models (equivalent fluid models) implemented in Pressure Acoustics. The poroacoustic models do not capture all effects, so sometimes it is also necessary to include the elastic waves in the porous matrix. This is covered by the so-called Biot-Allard theory for modeling poroelastic waves. The present model shows how the Poroelastic Waves interface, relatively simply, can be tailored to include the thermal and viscous effects as described by the Biot-Allard theory.

SONIC CRYSTAL

Phononic and sonic crystals have generated rising scientific interest for very diverse technological applications. These crystals are made of periodic distributions of scatterers embedded in a matrix. Under certain conditions, acoustic band gaps can form. These are spectral bands where propagation of waves is forbidden. This model first analyzes a sonic crystal and determines its band structure. Secondly, the transmission loss through a finite-sized crystal is analyzed and the results are compared to the band structure.

ACOUSTIC-STRUCTURE INTERACTION AND AIR FLOW IN VIOLINS

One model applies acoustic-structure interaction to study how the air mode resonance is affected by the coupled vibrations in the violin body. The other uses a potential flow approximation to find out how the air flow through the sound holes relates to their shape.

ENERGY CONSERVATION WITH THERMOVISCOUS ACOUSTICS

This tutorial model studies energy conservation in a small conceptual test setup. The model has an inlet and outlet and a Helmholtz resonator with a very narrow neck. The acoustics in the narrow neck are modeled with the Thermoacoustic interface for a detailed analysis of the thermal and viscous losses. To study and verify energy conservation, the model compares the total dissipated energy in the acoustic boundary layer to the total input minus the output power of the system.

SUBCOMPONENT LUMPING IN ACOUSTICS USING THE IMPEDANCE BOUNDARY CONDITION

This application illustrates a modeling approach for deriving physically consistent simplified models in the Acoustics Module. The approach consists of converting complex subcomponents to an impedance boundary condition and otherwise using simple acoustics throughout the COMSOL Multiphysics model.

Backward Compatibility with Version 5.2

The default settings have changed for the Incident Pressure Field in the Pressure Acoustics interfaces. Add the following line to obtain the old behavior of this feature when `ipf1` is a subfeature to pwr1:

```
model.physics("acpr").feature("pwr1").feature("ipf1").
set("c", "acpr.c_c");
```

Backward Compatibility with Version 5.1

- Plane Wave Radiation has been removed from the Linearized Potential Flow, Frequency Domain and Linearized Potential Flow, Transient interfaces in 1D axisymmetry.
- A new Lagrange multiplier variable (`ta.lm_slip`) has been added to the slip conditions in the Thermoacoustics, Frequency Domain interface. If you run a model with a segregated solver, you will need to add this variable to the group containing the velocity degree of freedom. You can also regenerate the default solver. In the API, you need to add `comp1_ta_lm_slip` to the declaration of the segregated step; for example, adding:

  ```
  model.sol("sol1").feature("s1").feature("se1").feature("ss1").
  set("segvar", new String[]{"comp1_p", "comp1_u", "comp1_T",
  "comp1_ta_lm_slip"});
  ```
- The Waveguide end impedance option has been removed from 2D, 1D axisymmetric, and 1D as it had no physical meaning in these space dimensions.

Backward Compatibility with Version 4.3a and Older

- The old **Perfectly Matched Layer** (PML) node that is located under the physics node has been discontinued. From the next version on, only the **Perfectly Matched Layer** nodes defined under the **Definitions** node are supported.
- In order for the old **Perfectly Matched Layer** (PML) node to be backward compatible, the PML feature must be placed after any domain **Monopole Source**, domain **Dipole Source**, or **Background Pressure Field** nodes.

Backward Compatibility with Version 4.3

- The symbols for volume sources for the acoustics monopole (Q_m; was Q) and dipole ($\mathbf{q}_d$; was $\mathbf{q}$) volume sources for 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 5.2a

SINGLE PARTICLE BATTERY INTERFACE

A new Single Particle Battery interface has been added. The interface offers a simplified approach for modeling various kinds of batteries, including lithium-ion and nickel-metal hydride batteries. The governing equations describing the battery, which are typically valid for low and medium current levels, can be defined either globally (resulting in a small computational load) or locally in the geometry.

FAST ASSEMBLY OPTION

A new fast assembly option is available in the Lithium-Ion Battery and Battery with Binary Electrolyte interfaces. By enabling fast assembly in the particle dimension in the **Porous Electrode** node, the computational time for some battery models using particle intercalation may be significantly reduced. The effect is most pronounced in 1D models when the number of mesh elements in the battery elements are comparable to the number of elements in the particle dimension. When the option is used, it is, however, not possible to postprocess data from the solution along the particle dimension axis. Also, the use of varying material properties (such as the solid diffusion coefficient) in the particle dimension is not supported. The setting is found in the **Porous Electrode** node in the Lithium-Ion and Battery with Binary Electrolyte interfaces.

NERNST-PLANCK-POISSON EQUATIONS MULTIPHYSICS INTERFACE

A new Nernst-Planck-Poisson Equations predefined multiphysics interface is now available in the Model Wizard. The Nernst-Planck-Poisson Equations interface adds Electrostatics and Transport of Diluted Species interface to the model, together with new Multiphysics coupling nodes: **Potential Coupling** and **Space Charge Density Coupling**.

EXTERNAL SHORT BOUNDARY CONDITION

A new External Short boundary condition has been introduced that allows for short circuiting of electrode surfaces, porous electrodes, and electrodes through an external lumped resistance. The feature is available as a separate boundary node, applicable to **Electrode** and **Porous Electrode** domains, and as a boundary condition option in the **Electrode** surface node. The feature is available in the following physics interfaces: Primary Current Distribution; Secondary Current Distribution; Tertiary Current

Distribution, Nernst Planck;, Lithium-Ion Battery; Battery with Binary Electrode; and Lead-Acid Battery.

ELECTROCHEMICAL HEAT SOURCE

A new **Electrochemical Heat Source** is now available as a multiphysics coupling node. The **Electrochemical Heat Source** multiphysics node offers an alternate way of coupling the electrochemical heat sources from any Electrochemistry interface into a Heat Transfer interface.

THERMODYNAMIC EQUILIBRIUM IN ELECTRODE REACTIONS

The **Electrode Reactions** node now supports a new Thermodynamic Equilibrium (Primary Condition in the Secondary Current Distribution interface) electrode kinetics type that assumes zero overpotential (negligible voltage losses).

FAR-FIELD APPROXIMATION IN CURRENT DISTRIBUTION, BEM

The Current Distribution, BEM interface now supports a new far-field approximation assembly method. In combination with an iterative solver, the far-field approximation can significantly reduce the computational time and memory usage for large problems.

POROUS ELECTRODE AND EDGE ELECTRODE IMPROVEMENTS

The **Porous Electrode** and the **Edge Electrode** nodes now support the addition of film resistances and dissolving-depositing species (previously, this was only supported in the **Electrode Surface** feature).

POROUS MEDIA TRANSPORT PROPERTIES FEATURE

A **Porous Media Transport Properties** feature has been added the Transport of Concentrated Species interface. This feature includes models for computing effective transport properties dependent on the porosity of the material.

IMPROVED SOLVER DEFAULTS IN BATTERY INTERFACES

The solver defaults in the Lithium-Ion Battery and Battery with Binary Electrolyte interfaces have been improved. In 2D and 3D, the intercalating concentrations are now put into separate groups in a segregated solver. This change reduces the memory requirements for large problems and reduces the computational time.

PSEUDO TIME STEPPING FOR TRANSPORT OF CONCENTRATED SPECIES

Pseudo time stepping can now be applied for the Transport of Concentrated Species interface. It can be used to improve the convergence of stationary simulations when the flow is dominated by convection — for example, when the flow is turbulent.

IMPROVED NUMERICAL STABILITY FOR HIGH AND LOW SOCS

Now, there is improved numerical stability for high and low SOCs in the Lithium-Ion Battery and Battery with Binary Electrolyte interfaces. The numerical stability when using Lithium Insertion kinetics in the **Porous Electrode Reaction** node has been improved for SOC values close to 0 and 100%. The improved kinetics formulation is used by default for new models. In order to use the new kinetics expression in an old model, enable it in the **Advanced Insertion Kinetics Expression Settings** section in the **Porous Electrode Reaction** node (only shown when Advanced Physics Options is enabled).

BATTERIES & FUEL CELLS MATERIAL LIBRARY

Concentration- and temperature-dependent parameters have been added for the LiPF6 in EC:DEC (1:1 by weight) electrolyte in the Batteries & Fuel Cells Material Library.

New and Updated Models and Applications in Version 5.2a

INTERNAL SHORT CIRCUIT OF A LITHIUM-ION BATTERY

This new tutorial, Internal Short Circuit of a Lithium-Ion Battery, simulates the temperature rise due to an internal short circuit by dendrite formation.

MODELS WITH UPDATED ELECTROLYTE AND MATERIAL LINKS

The following models under Batteries, Lithium-Ion have been updated with new LiPF6 in EC:DEC (1:1 by weight) electrolyte and material links for easier exchange of domain materials: `li_battery_management`, `li_battery_internal_resistance`, `li_battery_drive_cycle`, and `li_battery_rate_capability`.

OTHER UPDATED MODELS

- The `li_battery_rate_capability` model now includes a comparison of the rate capability of two different lithium-ion battery chemistries (MCMB/LMO versus LTO/NMC).
- The Capacity Fade of a Lithium-Ion Battery model (`capacity_fade`) has been updated to use more recent aging data from the scientific literature and a timescale factor has also been introduced in order to reduce the simulation time for multiple cycles.
- The 2D Lithium-Ion Battery model (`li_battery_tutorial_2d`) now has a negative lithium metal instead of a porous carbon electrode (a half-cell system).

- The `capacity_fade`, `li_air_battery_1d`, and `zn_ago_battery_1d` models have been updated using the new **Dissolving-Depositing Species** feature, which removes the need for additional ODE-DAE interfaces to solve for concentration and porosity.
- The `li_battery_single_particle` model has been revised to use the new Single Particle Battery interface.
- The `diffuse_double_layer` model has been updated to use the new Nernst-Planck-Poisson Equations multiphysics interface.

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 Version 5.2a

The CAD file import functionality included with these products has been extended to support new versions for some of the supported file formats (see under Read from File, CAD on www.comsol.com/products/specifications/cad).

CAD IMPORT MODULE GEOMETRY KERNEL UPGRADE

The CAD Import Module, the Design Module, and the LiveLink™ products for CAD utilize the Parasolid® geometry kernel from Siemens PLM for solid modeling operations, geometry repair, and defeaturing. (Without these products, a COMSOL-native geometry modeling kernel is used.) The CAD Import Module released with COMSOL 5.2a includes an upgraded version of the Parasolid kernel. As a result, a number of stability issues have been fixed, which makes the import of CAD models and solid operations more robust.

New Functionality in LiveLink™ for AutoCAD® in Version 5.2a

- The LiveLink™ interface now supports AutoCAD® 2017.
- The LiveLink™ interface now keeps track of the filename and path of the synchronized document. The filename is detected during the first synchronization. For subsequent synchronizations, the same file is used as long as it is open in AutoCAD®.
- The LiveLink™ interface now automatically generates selections during synchronization based on material assignments in the AutoCAD® file. It detects materials assigned to layers, solid and surface objects, and even to faces of an object. In the COMSOL model, domain or face selections are then generated based on whether the material is assigned to faces and surface objects, or solid objects, respectively.
- With a COMSOL Client installation of LiveLink™ *for* AutoCAD®, it is now possible to launch a COMSOL Server interface in a separate window by using the newly added **Connect to Server** button. From the COMSOL Server interface, it is possible to browse, and run apps, including those that are based on geometry synchronized from AutoCAD® using the LiveLink™ interface.

New Functionality in LiveLink™ for Inventor® in Version 5.2a

- The LiveLink™ interface now supports Inventor® 2017.
- The LiveLink™ interface now keeps track of the file name and path; and the table record of the synchronized iPart or iAssembly; as well as the view representation, positional representation, and level of detail representation, of the synchronized document. These properties are detected during the first synchronization. For subsequent synchronizations, the same properties are used, as long as the specified file is open in Inventor®, and the specified table record and representations exist.
- For assemblies, the LiveLink™ interface now automatically synchronizes parameters that are selected in the component files. Previously the parameters in the components had to be selected from the synchronized assembly file.
- With a COMSOL Client installation of LiveLink™ *for* Inventor®, it is now possible to launch a COMSOL Server interface in a window embedded into the Inventor user interface by using the newly added **Connect to Server** button. From the COMSOL Server interface, it is possible to browse, and run apps, including those that are based on geometry synchronized from Inventor® using the LiveLink™ interface.

New Functionality in LiveLink™ for PTC® Creo® Parametric™ in Version 5.2a

- The LiveLink™ interface now keeps track of the file name and path, as well as the family table instance, and simplified representation, of the synchronized document. These properties are detected during the first synchronization. For subsequent synchronizations, the same properties are used, as long as the specified file is open in PTC® Creo® Parametric™, and the specified instance and simplified representation exist.
- With a COMSOL Client installation of LiveLink™ *for* PTC® Creo® Parametric™, it is now possible to launch a COMSOL Server interface in a separate window by using the newly added **Connect to Server** button. From the COMSOL Server interface, it is possible to browse, and run apps, including those that are based on geometry synchronized from PTC® Creo® Parametric™ using the LiveLink™ interface.

New Functionality in LiveLink™ for PTC® Pro/ENGINEER® in Version 5.2a

The LiveLink™ interface now keeps track of the file name and path of the synchronized document. The filename is detected during the first synchronization. For subsequent synchronizations, the same file is used as long as it is open in PTC® Pro/ENGINEER®.

New Functionality in LiveLink™ for Revit® in Version 5.2a

- The LiveLink™ interface now supports Revit® 2017.
- The LiveLink™ interface now keeps track of the file name and path, as well as the configuration of the synchronized document. These properties are detected during the first synchronization. For subsequent synchronizations, the same properties are used, as long as the specified file is open in Revit®, and the specified configuration exists.
- With a COMSOL Client installation of LiveLink™ *for* Revit®, it is now possible to launch a COMSOL Server interface in a window embedded into the Revit user interface by using the newly added **Connect to Server** button. From the COMSOL Server interface, it is possible to browse and run apps, including those that are based on geometry synchronized from Revit® using the LiveLink™ interface.

LiveLink™ for Solid Edge®

NEW FUNCTIONALITY IN VERSION 5.2A

- The LiveLink™ interface now keeps track of the file name and path of the synchronized document. The filename is detected during the first synchronization. For subsequent synchronizations, the same file is used as long as it is open in Solid Edge®.
- With a COMSOL Client installation of LiveLink™ *for* Solid Edge®, it is now possible to launch a COMSOL Server interface in a separate window by using the newly added **Connect to Server** button. From the COMSOL Server interface, it is possible to browse and run apps, including those that are based on geometry synchronized from Solid Edge® using the LiveLink™ interface.

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`.

LiveLink™ for SOLIDWORKS®

NEW FUNCTIONALITY IN VERSION 5.2A

- The LiveLink™ interface now keeps track of the file name and path, as well as the configuration, and display state, of the synchronized document. These properties are detected during the first synchronization. For subsequent synchronizations, the same properties are used, as long as the specified file is open in SOLIDWORKS® and the specified configuration and display state exist.
- For assemblies, the LiveLink™ interface now automatically synchronizes parameters that are selected in the component files. Previously, the parameters in the components had to be selected from the synchronized assembly file.
- With a COMSOL Client installation of LiveLink™ *for* SOLIDWORKS®, it is now possible to launch a COMSOL Server interface in a window embedded into the SOLIDWORKS® user interface by using the newly added **Connect to Server** button. From the COMSOL Server interface, it is possible to browse, and run apps, including those that are based on geometry synchronized from SOLIDWORKS® using the LiveLink™ interface.

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`.

BACKWARD COMPATIBILITY WITH VERSION 5.2

Geometry Repair Tolerances

The geometry repair tolerance can be automatic, relative, or absolute in version 5.2a. In previous versions, the repair tolerance was always a relative tolerance.

- If you have not set the relative repair tolerance `repairtol`, you will get the new default repair tolerance type: `auto`. This will result in a valid geometry in almost all cases. In extreme cases, the result can have a different topology than in 5.2.
- In rare cases, if you have set the `repairtol` tolerance, version 5.2a can fail when building the geometry. If so, set `repairtoltype` to `auto` (select **Automatic** from the **Repair tolerance** list) to get the 5.2 behavior.

CFD Module

New Functionality in Version 5.2a

WEAKLY COMPRESSIBLE FLOW OPTION

There is a new compressibility option in the fluid flow interfaces. The new Weakly compressible flow option is suitable for nonisothermal flow, when density variations with respect to pressure are negligible. This is the case for the majority of flows at Mach numbers less than 0.3. The advantage of using this option instead of the Compressible flow (Ma < 0.3) option is that pressure waves, traveling at the speed of sound, need not be resolved in time-dependent simulations.

GRAVITY PROPERTY AND FEATURE

A gravity property is now available for the Single-Phase Flow and Non-Isothermal Flow branches (except Rotating Machinery) and can be selected at the interface level. When the gravity property is activated, a volume force due to gravity is added on all domains where the fluid flow interface is active, and a **Gravity** feature appears in the model tree. The gravity property adds an option to compensate for the hydrostatic pressure where a pressure-type boundary condition, such as an Outlet, Open Boundary, or Boundary Stress condition, is specified. This compensation is exact for incompressible flow, and for the other two compressibility options, it gives a good approximation of the pressure profile on the outlet. There is also an option to use the reduced pressure — that is, to automatically include the hydrostatic pressure in the definition of the dependent variable for pressure.

BOUSSINESQ APPROXIMATION IN NONISOTHERMAL FLOW

When incompressible nonisothermal flow is simulated using a material for which the coefficient of thermal expansion is defined, a buoyancy force with a linear temperature dependence is added, while all fluid properties (such as density, viscosity, and thermal conductivity) remain constant. This is generally referred to as *Boussinesq approximation*, and it is a popular modeling approach in nonisothermal flow applications. An alternative to specifying the linearization point using a built-in material is to define the reference density and coefficient of thermal expansion explicitly in the **Multiphysics** node.

SWIRL FLOW FOR THE FAN FEATURE

The flow direction now includes a swirl option for fans applied on inlets. The swirl on the downstream side of the fan is determined by setting the rotation rate and a swirl coefficient defined as the ratio of the angular velocity of the flow to that of the blades.

NEW TWO-PHASE FLOW, LEVEL SET AND TWO-PHASE FLOW, PHASE FIELD MULTIPHYSICS INTERFACES

New Two-Phase Flow, Level Set and Two-Phase Flow, Phase Field multiphysics interfaces, replacing the previous versions, are available in the Model Wizard. By selecting a Two-Phase Flow interface under the Multiphase Flow branch, a laminar or turbulent Single-Phase Flow interface and a Level Set or Phase Field interface are added to the Model Builder. In addition, a multiphysics coupling feature, **Two-Phase Flow, Level Set** or **Two-Phase Flow, Phase Field**, is added under the **Multiphysics** node. The Two-Phase Flow multiphysics coupling feature has predefined controls and couplings between the separate interfaces. In addition, a **Wetted Wall** boundary condition feature can be added in the **Multiphysics** node. Opening an application from an earlier version, the previous Two-Phase Flow interfaces are used.

UPDATED CROSSWIND STABILIZATION

The consistent crosswind stabilization has been improved.

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 5.2a 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 Model 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 names 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 5.2a 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 5.2a 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 5.2a 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 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 converge. However, you can still activate the penalty diffusion in the **Advanced** section of the Mixture Model physics interface settings.

EULER-EULER MODEL, LAMINAR FLOW

The results from models using the Euler-Euler Model, Laminar Flow interface may differ in version 5.2a 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 ways:

- 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 clear their time-dependent solutions when opened in 5.2a. 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 5.2a.

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 interface 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 $-Q_{\mathrm{br}}\cdot\mathbf{u}/\varepsilon_{\mathrm{p}}^2$ in the right-hand side previously lacked the factor $1/\varepsilon_{\mathrm{p}}^2$, 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 have 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
- The 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 Functions 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 of the other boundary conditions for the dependent velocity variables. The previously used Lagrange multiplier `un_lm` has been removed.

When you open models saved in version 4.3, they include `un_lm` until the model is re-solved. In some cases, occurrences of `un_lm` in the solver sequence must be replaced manually. This is the case if `un_lm` 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 5.2a for postprocessing, or opened and re-saved in version 4.3 before being opened in version 5.2a.

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 5.2a than in version 4.3 and the results 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 between versions 4.3 and 5.2a 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 5.2a

NEW REACTING FLOW MULTIPHYSICS INTERFACES

New Reacting Flow multiphysics interfaces, replacing the previous versions, are available in the Model Wizard. By selecting a Reacting Flow interface under the Chemical Species Transport branch, one of the Single-Phase Flow interfaces and a Transport of Concentrated Species interface are added to the Model Builder. In addition, a multiphysics coupling feature, **Reacting Flow**, is added under the **Multiphysics** node. The **Reacting Flow** feature predefines and controls the couplings between the separate interfaces. Opening an application from an earlier version, the previous Reacting Flow interfaces are used.

POROUS MEDIA TRANSPORT PROPERTIES FEATURE

A **Porous Media Transport Properties** feature has been added the Transport of Concentrated Species interface. This feature includes models for computing effective transport properties dependent on the porosity of the material and can be used to study species transport in a fluid mixture propagating through a porous medium.

NEW AND RENAMED REACTION FEATURES

A new **Reaction** feature is available in the Transport of Concentrated Species interface. It defines species source terms from a single chemical reaction, using an Arrhenius or user-defined rate expression. When licensed to the CFD Module, the feature includes the possibility to model the turbulent reaction rate using the eddy-dissipation model.

The **Reaction Sources** feature corresponds to the previous **Reactions** feature in the Transport of Concentrated Species interface and the concentrated species version of the Reacting Flow in Porous Media interface. The feature in Transport of Concentrated Species has also been extended with a **Reacting Volume** section, which can be used to specify the basis for the reaction source terms on porous media domains.

PSEUDO TIME STEPPING FOR TRANSPORT OF CONCENTRATED SPECIES

Pseudo time stepping can now be applied for the Transport of Concentrated Species interface. It can be used to improve the convergence of stationary simulations when the flow is dominated by convection — for example, when the flow is turbulent.

REACTIVE PELLET BED: SURFACE SPECIES AND SURFACE REACTIONS

Surface species functionality has been added to the **Reactive Pellet Bed** feature (in the Transport of Diluted Species interface and the Transport of Diluted Species in Porous Media interface). Surface species existing within the porous pellet, attached to the solid matrix-fluid interface, can be defined in the **Surface Species** section of the feature. Adding surface species, the **Reactions** subfeature includes the possibility to model surface reactions and corresponding heat sources.

Surface reactions, defined in the Reaction Engineering interface, can now be exported to occur within porous pellets in a space-dependent model, using the **Generate Space-Dependent Model** feature. Selecting to export surface reactions to porous pellets, a **Reactive Pellet Bed** feature is added (in a Transport of Diluted Species interface or Transport of Diluted Species in Porous Media interface) and automatically set up to include the defined reaction kinetics.

Backward Compatibility with Version 5.0 and Earlier

REACTION ENGINEERING AND CHEMISTRY INTERFACES

The API syntax for creating surface species has changed. Running Java® files from previous versions may fail due to the new default name indexes, including **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 5.2 and later:

```
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 recomputing 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 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 (in 5.0 and earlier) from

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

to (in 5.2 and later)

```
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 (in 5.0 and earlier) from

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

to (in 5.2 and later)

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

Java® API Syntax Changes

The 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);
```

For version 5.2 and later, change this to:

```
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 described in the Transport of Concentrated Species Interface section above, holds true for the Surface Reactions interface in version 5.2a.

Java® API Syntax Changes

The API syntax for setting the diffusion tensor of surface species, the molar mass and density of the bulk species, and 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 Model Java® files from previous versions, apply the syntax changes as shown 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"});
```

For version 5.2 and later, change this to:

```
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 shown 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");
```

For version 5.2 and later, change this to:

```
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 shown 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");
```

For version 5.2 and later, change this to:

```
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 5.2a

SACRIFICIAL EDGE ANODE

Th new **Sacrificial Edge Anode** node is a customized node for modeling slender sacrificial anodes along geometric edges. It is available in the Secondary Current Distribution interface. The **Sacrificial Edge Anode** node makes it possible to model changing cathodic protection properties of the anodes as they dissolve in time-dependent simulations.

NERNST-PLANCK-POISSON EQUATIONS MULTIPHYSICS INTERFACE

A new Nernst-Planck-Poisson Equations predefined multiphysics interface is now available in the Model Wizard. The Nernst-Planck-Poisson Equations interface adds the Electrostatics and Transport of Diluted Species interfaces to the model, together with new Multiphysics coupling nodes: **Potential Coupling** and **Space Charge Density Coupling**.

EXTERNAL SHORT BOUNDARY CONDITION

A new External Short boundary condition has been introduced that allows for short circuiting of electrode surfaces, porous electrodes, and electrodes through an external lumped resistance. The feature is available as a separate boundary node, applicable to **Electrode** and **Porous Electrode** domain nodes, and as a Boundary Condition option in the **Electrode** surface node. The feature is available in the following physics interfaces: Primary Current Distribution; Secondary Current Distribution; Tertiary Current Distribution, Nernst Planck;, Lithium-Ion Battery; Battery with Binary Electrode; and Lead-Acid Battery.

ELECTROCHEMICAL HEAT SOURCE

A new **Electrochemical Heat Source** node is now available as a Multiphysics coupling node. The **Electrochemical Heat Source** multiphysics node offers an alternate way of coupling the electrochemical heat sources from any Electrochemistry interface into a Heat Transfer interface.

THERMODYNAMIC EQUILIBRIUM IN ELECTRODE REACTIONS

The **Electrode Reactions** node now supports a new Thermodynamic Equilibrium (Primary Condition in the Secondary Current Distribution interface) electrode kinetics type that assumes zero overpotential (negligible voltage losses).

POROUS ELECTRODE AND EDGE ELECTRODE IMPROVEMENTS

The **Porous Electrode** and **Edge Electrode** nodes now support the addition of film resistances and dissolving-depositing species (previously this was only supported in the **Electrode Surface** feature).

FAR-FIELD APPROXIMATION IN CURRENT DISTRIBUTION, BEM

The Current Distribution, BEM interface now supports a new far-field approximation assembly method. In combination with an iterative solver, the far-field approximation can significantly reduce the computational time and memory usage for large problems.

IMPROVED SOLVER DEFAULTS

An automatic predictor is now used in auxiliary sweeps for Stationary study steps in the Electrochemistry interfaces. The automatic predictor lowers computational time in, for instance, polarization sweeps.

New and Updated Models in Version 5.2a

MONOPILE WITH DISSOLVING SACRIFICIAL ANODES TUTORIAL

The new Monopile tutorial demonstrates how the **Sacrificial Edge Anode** node can be used in cathodic protection simulations.

UPDATED MODELS

- The Cathodic Protection of Steel in Reinforced Concrete model has been updated to use the new Primary Condition (Thermodynamic Equilibrium) electrode kinetics.
- The Diffuse Double Layer model has been updated to use the new Nernst-Planck-Poisson Equations multiphysics interface.

Backward Compatibility with Earlier Versions

SURFACE REACTIONS INTERFACE

See Chemical Reaction Engineering Module.

ECAD Import Module

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 the types *Metal* and *Dielectric* are initialized with the string `on` in the `importlayer` string array. Previously, this also included layers of the type *Drill*.

Electrochemistry Module

New Functionality in Version 5.2a

NERNST-PLANCK-POISSON EQUATIONS MULTIPHYSICS INTERFACE

A new Nernst-Planck-Poisson Equations predefined multiphysics interface is now available in the Model Wizard. The Nernst-Planck-Poisson Equations interface adds the Electrostatics and Transport of Diluted Species interfaces to the model, together with new multiphysics coupling nodes: **Potential Coupling** and **Space Charge Density Coupling**.

EXTERNAL SHORT BOUNDARY CONDITION

A new External Short boundary condition has been introduced that allows for short circuiting of electrode surfaces, porous electrodes, and electrodes through an external lumped resistance. The feature is available as a separate boundary node, applicable to **Electrode** and **Porous Electrode** domain nodes, and as a Boundary Condition option in the **Electrode** surface node. The feature is available in the following physics interfaces: Primary Current Distribution; Secondary Current Distribution; Tertiary Current Distribution, Nernst Planck; Lithium-Ion Battery; Battery with Binary Electrode; and Lead-Acid Battery.

ELECTROCHEMICAL HEAT SOURCE

A new **Electrochemical Heat Source** node is now available as a multiphysics coupling node. The **Electrochemical Heat Source** multiphysics node offers an alternate way of coupling the electrochemical heat sources from any Electrochemistry interface into a Heat Transfer interface.

THERMODYNAMIC EQUILIBRIUM IN ELECTRODE REACTIONS

The **Electrode Reactions** node now supports a new Thermodynamic Equilibrium (Primary Condition in the Secondary Current Distribution interface) electrode kinetics type that assumes zero overpotential (negligible voltage losses).

POROUS ELECTRODE AND EDGE ELECTRODE IMPROVEMENTS

The **Porous Electrode** and **Edge Electrode** nodes now support the addition of film resistances and dissolving-depositing species (previously this was only supported in the **Electrode Surface** feature).

FAR-FIELD APPROXIMATION IN CURRENT DISTRIBUTION, BEM

The Current Distribution, BEM interface now supports a new far-field approximation assembly method. In combination with an iterative solver, the far-field approximation can significantly reduce the computational time and memory usage for large problems.

IMPROVED SOLVER DEFAULTS

An automatic predictor is now used in auxiliary sweeps for Stationary study steps in the Electrochemistry interfaces. The automatic predictor lowers computational time in, for instance, polarization sweeps.

Updated Models in Version 5.2a

The Diffuse Double Layer and Diffuse Double Layer with Charge Transfer models have been updated to use the new Nernst-Planck-Poisson Equations multiphysics interface.

Backward Compatibility with Previous Versions

SURFACE REACTIONS INTERFACE

See Chemical Reaction Engineering Module.

Electrodeposition Module

New Functionality in Version 5.2a

NERNST-PLANCK-POISSON EQUATIONS MULTIPHYSICS INTERFACE

A new Nernst-Planck-Poisson Equations predefined multiphysics interface is now available in the Model Wizard. The Nernst-Planck-Poisson Equations interface adds the Electrostatics and Transport of Diluted Species interfaces to the model, together with new multiphysics coupling nodes: **Potential Coupling** and **Space Charge Density Coupling**.

EXTERNAL SHORT BOUNDARY CONDITION

A new External Short boundary condition has been introduced that allows for short circuiting of electrode surfaces, porous electrodes, and electrodes through an external lumped resistance. The feature is available as a separate boundary node, applicable to **Electrode** and **Porous Electrode** domain nodes, and as a Boundary Condition option in the **Electrode** surface node. The feature is available in the following physics interfaces: Primary Current Distribution; Secondary Current Distribution; Tertiary Current Distribution, Nernst Planck;, Lithium-Ion Battery; Battery with Binary Electrode; and Lead-Acid Battery.

ELECTROCHEMICAL HEAT SOURCE

A new **Electrochemical Heat Source** is now available as a multiphysics coupling node. The **Electrochemical Heat Source** multiphysics node offers an alternate way of coupling the electrochemical heat sources from any Electrochemistry interface into a Heat Transfer interface.

THERMODYNAMIC EQUILIBRIUM IN ELECTRODE REACTIONS

The **Electrode Reactions** node now support a new Thermodynamic Equilibrium (Primary Condition in the Secondary Current Distribution interface) electrode kinetics type that assumes zero overpotential (negligible voltage losses).

POROUS ELECTRODE AND EDGE ELECTRODE IMPROVEMENTS

The **Porous Electrode** and **Edge Electrode** nodes now support the addition of film resistances and dissolving-depositing species (previously this was only supported in the **Electrode Surface** feature).

Far-Field Approximation in Current Distribution, BEM

The Current Distribution, BEM interface now supports a new far-field approximation assembly method. In combination with an iterative solver, the far-field approximation can significantly reduce the computational time and memory usage for large problems.

Improved Solver Defaults

An automatic predictor is now used in auxiliary sweeps for Stationary study steps in the Electrochemistry interfaces. The automatic predictor lowers computational time in, for instance, polarization sweeps.

Updated Model in Version 5.2a

The Diffuse Double Layer model has been updated to use the new Nernst-Planck-Poisson Equations multiphysics interface.

Backward Compatibility with Previous Versions

Surface Reactions Interface

See Chemical Reaction Engineering Module.

Fatigue Module

New Functionality in Version 5.2a

NEW FATIGUE MODEL FOR FATIGUE PREDICTION IN COMPRESSIVE LOAD CYCLES: DANG VAN

The Dang Van model has been added to the Stress-Based fatigue model family. The new model incorporates, via the hydrostatic stress, the sensitivity of the compressive state on the fatigue prediction. The Dang Van model is therefore suitable for evaluating compressive load cycles and can be used in, for example, contact fatigue. The model can predict fatigue development prior to spalling that is observed in bearing and rail applications.

FATIGUE MODEL PARAMETERIZATION

Most of the fatigue model setting options have been parameterized. This improvement implies that parameters and variables can be assigned as fatigue model settings and can be evaluated in a parametric study. Only two options cannot be parametrized: the Number of mean value bins and Number of range bins in the **Cumulative Damage** feature.

New and Updated Models in Version 5.2a

ROLLING CONTACT FATIGUE IN A LINEAR GUIDE

A rolling load initiates spalling on the rail raceway of a linear guide. The Dang Van model is used to simulate the rolling contact fatigue. This new model demonstrates how to efficiently mesh when modeling applications where high compressive loads are transferred through a small contact area.

STANDING CONTACT FATIGUE

In this new example, a surface-hardened material is subjected to standing contact fatigue. As a result, both the material properties and the residual stress vary through the depth. The fatigue prediction is based on the Dang Van model.

FATIGUE ANALYSIS OF A NON-PROPORTIONALLY LOADED SHAFT WITH A FILLET

This example has been extended with the fatigue prediction based on the Dang Van model. The differences between the results of the three stress-based models, Findley, Matake, and Dang Van, are discussed.

BRACKET — FATIGUE EVALUATION

The fatigue evaluation of the tutorial bracket model is extended with a parametric study. The model demonstrates how to perform a parametric study of a fatigue analysis.

Geomechanics Module

New Functionality in Version 5.2a

MIXED ISOTROPIC AND KINEMATIC HARDENING

The plasticity model has been updated so that isotropic and kinematic hardening models can be used simultaneously.

TIME DERIVATIVES OF PLASTIC STRAINS

Time derivatives of the components of the plastic strain are now available, so it is possible to use expressions like `d(solid.epe,TIME)` to create strain-rate-dependent plasticity models.

Heat Transfer Module

New and Improved Functionality in Version 5.2a

TIME-DEPENDENT CLIMATE DATA FOR 6000+ STATIONS (ASHRAE 2013)

A new section named **Ambient Settings** is now available in the settings of the heat transfer interfaces, for the definition of ambient variables (temperature, relative humidity, absolute pressure, wind velocity, and solar irradiance). These variables are defined once and available as inputs from several features for all of the interfaces in the Heat Transfer Module.

By default, the ambient variables are **User defined**. With the Heat Transfer license, the ambient variables can also be defined from typical time-dependent meteorological data (Ashrae, 2013). Several settings for the choice of location (among 6000+ stations worldwide), the time, and the ambient conditions are available.

The ambient variables are available as inputs in several features. For example, in the **Heat Flux** feature, the ambient temperature, ambient absolute pressure, and wind velocity can be used in the correlations defining the heat transfer coefficient of external forced convection over a plate.

HEAT AND MOISTURE MULTIPHYSICS COUPLING

A set of new interfaces and features is available for modeling coupled heat and moisture transport in building materials, by taking into account heat and moisture storage, latent heat effects, and liquid and convective transport of moisture. The heat and moisture coupling can model different moisture variation phenomena in building components, such as drying of initial construction moisture, condensation due to migration of moisture from outside to inside in summer, and moisture accumulation by interstitial condensation due to diffusion in winter.

The Heat and Moisture Transport interface is found under the **Heat Transfer** branch in the model wizard. It adds the Heat Transfer in Building Materials interface, the Moisture Transport interface, and the **Heat and Moisture** multiphysics node.

BUILDING MATERIAL MODEL

The **Building Material** model is the default domain feature of the Heat Transfer in Building Materials interface. It is also available in any heat transfer interface when the **Heat transfer in porous media** property is selected.

This feature models porous media containing water and moist air following the partial derivative equations derived from *EN 15026, Hygrothermal performance of building components and building elements - Assessment of moisture transfer by numerical, CEN, 2007*.

The effective thermal properties are determined from the dry material properties and water content. In addition, a heat source or sink is determined from the moisture transfer and the latent heat of evaporation.

MOISTURE TRANSPORT INTERFACE

The Moisture Transport interface models moisture transfer. The default domain feature, **Porous Medium**, accounts for the moisture storage, capillary suction forces, and convective transport of vapor. Similarly to the **Building Material** feature, it implements partial derivative equations derived from *EN 15026*.

The Moisture Transport interface also provides features to define **Moisture Source**, **Thin Moisture Barrier**, **Moisture Content**, and **Moisture Flux**.

HEAT TRANSFER IN THIN FILMS AND FRACTURE

Two new versions of the Heat Transfer in Thin Shells interface are available for the modeling of thin films and fractures.

The Heat Transfer in Thin Film interface and the Heat Transfer in Fractures interface are available under the **Thin Structures** group of the **Heat Transfer** branch in the model wizard.

They have the **Thin Film** and **Fracture** features as default boundary feature and have the same features available as the Heat Transfer in Thin Shells interface.

GENERAL FORMULATION FOR THIN FILM

A new option, the **General** thin film model, is available in the **Thin Film** feature. It provides a discretization of the temperature field through the film thickness. This new option defines an extra dimension to account for the temperature variations through the film thickness.

This formulation may be useful for applications such as bearings modeling or, more generally, when an accurate representation of the temperature profile in a film is needed, particularly in the presence of large heat sources or temperature differences at the film surfaces.

In addition, the former default option, a lumped model that accounts only for tangential gradients, is renamed **Thermally thin approximation**.

SECTOR SYMMETRY CONDITION FOR SURFACE-TO-SURFACE RADIATION

A new option, **Sector of symmetry**, is now available for 2D and 3D models in the **Symmetry for Surface-to-Surface Radiation** feature. It supports an arbitrary number of sectors and provides an option to define a reflection plane in each sector.

This option improves the computational efficiency by computing and storing view factors on only a sector of a geometry with symmetry. Consequently, the number of degrees of freedom of all variables should decrease accordingly.

REFACTORED OPACITY FEATURE

The **Opaque** subfeature (used for surface-to-surface radiation) available in previous versions under a restricted list of domain features — **Solid** (formerly **Heat Transfer in Solids**), **Porous Medium** (formerly **Heat Transfer in Porous Media**), and **Biological Tissue** — has been replaced by the **Opacity** feature under all main domain features. This includes **Fluid** (formerly **Heat Transfer in Fluids**), **Phase Change Material** (formerly **Heat Transfer with Phase Change**), **Building Material**, and **Isothermal Domain**.

The opacity is set by selecting either **Transparent** or **Opaque** in the settings of the feature.

GRAVITY FEATURE

The **Gravity** property is available in the Single Phase Flow interface. When selected, it adds the **Gravity** feature to the application where it is possible to edit the **Acceleration of the gravity** definition. The **Gravity** feature defines a volume force due to the gravity in all the domains where the interface is active.

Two formulations are available to implement the fluid flow equation with gravity: the relative pressure formulation (default) and the reduced pressure formulation. When the relative pressure formulation is selected, features that use an external pressure or external total stress get an option to account for the hydrostatic pressure (incompressible flow) or for the hydrostatic pressure approximation (weakly compressible and compressible flow). When the reduced pressure option is selected, the flow equations are defined using the reduced pressure as a dependent variable. This is suitable in cases where the buoyancy force variations are small compared to the absolute value of the buoyancy forces.

SINGLE-PHASE FLOW COMPRESSIBILITY

The single-phase flow options have been revised. In COMSOL Multiphysics 5.2a, three compressibility options are now available: **Incompressible flow**, **Weakly compressible flow**, and **Compressible (Ma<0.3)**.

When the **Incompressible flow** option is selected, the incompressible version of the flow equation, assuming that the velocity field is divergence free, is used. With this option, the density is evaluated at the **Reference pressure level** and at the **Reference temperature** defined in the fluid flow interface, which guarantee that the density is correct in most cases. The dynamic viscosity is also evaluated at the reference temperature. In addition, when a **Non-Isothermal Flow** multiphysics feature couples a heat transfer interface and a flow interface using the **Incompressible flow** option, the thermal material properties are also evaluated at the reference temperature.

When the **Weakly compressible flow** option is selected, the compressible version of the flow equation is used. However, the density is evaluated at the **Reference pressure level** defined in the fluid flow interface. The temperature dependency of the density is considered. This option is recommended in cases where the density has a very small dependency on the pressure, which does not significantly affect the results but may significantly increase the computational cost.

The **Compressible flow (Ma<0.3)** corresponds to the general formulation of the fluid flow equations. It does not assume any hypothesis on the density property.

FEATURES AND OPTIONS NAME CHANGE

The names of the following domain features have been updated:

- **Heat Transfer in Solids** is renamed **Solid**
- **Heat Transfer in Fluids** is renamed **Fluid**
- **Heat Transfer in Porous Media** is renamed **Porous Medium**
- **Heat Transfer with Phase Change** is renamed **Phase Change Material**

Note: The names of the interfaces have not been changed.

The names of the following options have been updated:

- The **Conductive** option of **Thin Layer** is renamed **Thermally thin approximation**
- The **Resistive** option of **Thin Layer** is renamed **Thermally thick approximation**
- The **Overall heat transfer rate** option in several flux and source conditions has been renamed **Heat rate**

MISCELLANEOUS

- The **Non-Isothermal Flow** multiphysics coupling is now compatible with the **Phase Change Material**.
- The **Pressure Work** subfeature is now available for heat transfer in porous media.
- When **Solar position** is selected in **External Radiation Source**, the new option, **Update time from solver**, automatically updates the time used to evaluate the solar position in transient studies. The time from the user input corresponds to the time for $t = 0$.
- The **Heat Source** subfeature under **Pair Thin Film** and **Pair Fracture** has been changed into a pair feature, named **Pair Heat Source**. This allows you to define its selection from a pair and ensures that the **Heat Rate** option works as expected when the source and destination boundaries of the pair are not in contact on their entire surfaces.
- The **Discrete ordinate method** with **Constant** discretization level is now supported for all kinds of mesh elements: Support for pyramidal elements has been added. In addition, the GUI has been updated to hide the **Constant** discretization level when **P1 approximation** is selected, since it was not a valid choice.
- There is added support for side temperatures when **Diffuse Surface**, **Diffuse Mirror**, and **Prescribed Radiosity** features contribute to **Thin Layered Shell**.
- There are new postprocessing variables for dew point temperature and equivalent temperature when **Moist Air** is used as the **Fluid type** or when the **Building Material** feature is active.
- The implementation of the **Thin Rod** feature has been revised in order to increase its numerical stability.
- The following functions, `fpsat`, `fc`, `fxvap`, `feb`, and fIb have been updated to be compatible with all unit systems.

- The Bioheat material database has been updated. The following materials have been added:
  - Liver (porcine)
  - Lung
  - Myocardium (human)
  - Myocardium (porcine)
  - Renal cortex
  - Renal medulla
  - Spleen
    For almost all of these materials, temperature-dependent properties are provided through linear or polynomial expressions. If not, constant values are given.

  In addition, the properties for the Prostate material have been updated.

New Applications in Version 5.2a

INLINE INDUCTION HEATER

This app computes the efficiency of a magnetic induction apparatus for the heating of liquid food flowing in a set of ferritic stainless steel pipes. A circular electromagnetic coil is wound around a set of pipes in which a fluid flows. Various configurations are available for the set of pipes and for the coil to optimize the heat exchange with the fluid, while ensuring a homogeneous temperature within it for a given flow rate.

EVAPORATION IN POROUS MEDIA LARGE RATE

This tutorial describes a drying case with laminar airflow through an unsaturated porous medium. The air is dry at the inlet and its moisture content increases as air flows through the porous medium. The water saturation in the porous medium is computed over time. It is supposed that water leaves the porous medium only as vapor.

Updated Application in Version 5.2a

Several applications have been updated to benefit from the new **Opacity**, **Gravity**, **Ambient Settings**, and **Flow Compressibility** features.

Backward Compatibility with Version 5.2

Several enhancements or bug fixes may require an update of the existing applications or induce a change in the results. Here is the list of the changes:

- When **Solar position** is selected in **External Radiation Source**, the new option **Update time from solver** is selected by default. When loading an MPH file that was created before version 5.2a this option will be cleared. However, for the API, for time-dependent studies, it will be necessary to update the code to get the same behavior as in previous versions. In most cases, removing the *t* term in the Second field is the best option. Alternatively, clearing the **Update time from solver** option will ensure that the same behavior as before is obtained.
- The crosswind stabilization of the heat transfer and fluid flow interfaces has been updated. In general, the update reduces the smoothing induced by the stabilization compared to previous version. In addition, when you use the **Non-Isothermal Flow** multiphysics coupling and the fluid is defined as incompressible, the uncoupled versions of the streamlines and crosswind stabilization are used.
- Since **Pair Thin Film** and **Pair Fracture** are now pair features, their selection needs to be redefined.
- The definition of the `ht.`*`feature`*`.fc(RH.T,pA)` function when **Moist air** fluid type is used in **Fluid** feature has been corrected. This change can affect the results in simulations where it is used.
- The frame support has been enhanced to support moving mesh combined with deformed geometry in the same component.
- The **Prescribed Radiosity** feature has been updated so that it no longer ignores the **Temperature** model input.
- In surface-to-surface boundary features, the ambient temperature definition has been fixed when the **Opacity Controlled** and **Defined ambient temperature on each side** options are used simultaneously.

Backward Compatibility with Version 5.1

CHANGES IN VARIABLE NAMES

The names of the heat capacity variables have changed in several features for consistency:

- In the **Bioheat** feature, `ht.cb` has become `ht.cp_b`.
- In the **Thin Layer** feature, `ht.cs` has become `ht.cp_s`.

- In the **Thin Rod** feature, ht.cl has become ht.cp_l.
- In the **Thin Layered Shell** feature, htsh.cs1 has become htsh.cp_s1, and so on.
- In the **Heat Transfer in Porous Media** feature, ht.c_pp has become ht.cp_p, ht.c_pp1 has become ht.cp_p1, and so on.
- In the **Immobile Fluids** feature, ht.c_pg1 has become ht.cp_g1, and so on.
- In the **Fracture** feature, ht.c_pfr has become ht.cp_fr.

The names of the thermal resistance and absolute thermal resistance variables have changed in the **Isothermal Domain Interface** feature:

- ht.idi1.Rs has become ht.idi1.R_t
- ht.idi1.Rsth has become ht.idi1.R_tth

The name of the fluid viscosity variable has changed in the **Local Thermal Non-Equilibrium** multiphysics coupling: ltne1.mu_f has become ltne1.mu.

Backward Compatibility with Version 5.0

OBSOLETE FEATURES

A number of features are now obsolete because they have been merged or replaced by a more general features. However, when models from COMSOL 5.0 or before contain these features, some of the obsolete features are visible in version 5.2a 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 5.2a, but the feature will be missing.

Updating the models is recommended. Use the new feature and remove the obsolete feature to make sure that the model will remain compatible with future versions of COMSOL Multiphysics.

The table below summarizes the changes:

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

OBSOLETE FEATURE	REPLACE IN 5.2A BY
Convective Heat Flux	**Heat Flux** with the **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

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

OBSOLETE FEATURE	REPLACE IN 5.2A BY
Highly Conductive Layer	**Thin Layer** with **Layer type** set to **Conductive**
Thin Thermally Resistive Layer	**Thin Layer** with **Layer type** set to **Resistive**

TABLE 1-2: OBSOLETE FEATURES THAT ARE NO LONGER AVAILABLE

OBSOLETE FEATURE	REPLACE IN 5.2A BY
Infinite Element	**Infinite Element** is no longer defined in the physic interface. Infinite elements 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 to **Radiation in Participating Media**
Porous Matrix	**Heat Transfer in Porous Media**

Note that it is not possible to add any of the obsolete features from the COMSOL Desktop in version 5.2a. 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 nonisothermal 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 `Ptot`, `Pbtot`, `Pltot`, and `Pstot` become `P0`, `Pb`, `Pl`, and `Ps`, 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:

- `Qfric`, `QfricPow`, and `Qrate` become `Qb`, `Pb`, and `Qbpart`, respectively
- `Qfric0`, `Qrate0`, `Econtact0`, `hrad0`, `hgap0`, and `hconstr0` are removed and migrated to existing variables `Qb`, `Qbpart`, `Econtact`, `hrad`, `hgap`, and `hconstr`, respectively, in old models.

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

The Heat Transfer in Thin Shells 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 3D geometries and thickness or area parameters are always available.

API Users

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 for the following features:

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

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

- If you specified the thickness parameter twice (there were two instances of specifying the parameter).
- If you first specified the thickness and in a second command the total power value.
- If you first specified the thickness and then activated 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 a heat source example (2D and 1D axisymmetric 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 used since COMSOL Multiphysics 5.0, and `dz_entr_44_value` is the value used in previous versions.

In 1D geometries, the following update should be performed: `Ptot_50 = Ptot_44_value*ht.d/Ac_input_44_value` ,
where `Ac_input_44_value` is the value 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.trlflux`
- 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. The variables `chflux` and `rflux` may have opposite signs 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 model results if this boundary condition was previously applied to a boundary adjacent to a nonfluid domain.

HEAT TRANSFER WITH PHASE CHANGE NOW USES A VOLUMETRIC FORMULATION

The variables `ht.theta`*`i`* 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 change 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 definitions, physics interfaces, or postprocessing show different results in versions 5.0 and later 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.

Backward Compatibility with Version 4.3a

HEAT TRANSFER IN POROUS MEDIA

Old models using the **Porous** feature will be supported in the COMSOL Multiphysics interface. However, it is recommended to replace this feature with 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 a 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 variable names, previously named `I1` and so on, became `ht.I1` (when there are fewer 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 a 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 the S2 quadrature. However, the theoretical error is not different and the 2D and 3D implementations are now consistent.

Backward Compatibility with Version 4.3

THE DEFAULT MODEL LIST HAS BEEN REMOVED

Heat transfer physics 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 heat flux/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 MPH-files created in version 4.3 are converted to appear and behave the same way as in 5.0 or a 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 versions 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 have been removed:

- **Porous Matrix** (replaced by **Heat Transfer in Porous Media**)

- **Infinite Element** (replaced by **Infinite Element Domain** declared in the **Definitions** node)
- **Electrochemical Heat Flux** (replaced by **Boundary Heat Source**)
- **Reaction Heat Flux** (replaced by **Boundary Heat Source**)
- **Heat Transfer in Participating Media** (replaced by **Heat Transfer with Radiation in Participating Media**)

When models created in older versions are loaded in version 5.2a, these features are removed from the model tree and the model can still be opened. The new corresponding feature can be added to replace the removed one. In the particular case of a feature defined as the default feature of an interface in the model, the model cannot be opened.

The table below contains a list of variables that have been removed and replaced by new corresponding expressions.

TABLE 1-3: REMOVED VARIABLES AND CORRESPONDING EXPRESSIONS

REMOVED 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 5.2a

PIEZORESISTIVITY MULTIPHYSICS COUPLINGS

The three dedicated physics interfaces for the piezoresistive effect, namely, Piezoresistivity, Domain Currents; Piezoresistivity, Boundary Currents; and Piezoresistivity, Shell, have all been upgraded to corresponding multiphysics couplings. The old dedicated physics interfaces will be deprecated in a future version.

The new multiphysics couplings allow the flexibility to easily enable and disable each constituent physics interface and the couplings between the physics.

ELECTRIC CURRENTS, SHELL

The Electric Currents, Shell interface is now included with the MEMS Module for the Piezoresistivity, Shell multiphysics coupling.

MAGNETOSTRICTION INTERFACE

A new Magnetostriction interface has been introduced. When added, a Solid Mechanics interface, a Magnetic Fields interface, and a **Magnetostriction** multiphysics coupling are created.

In the Solid Mechanics interface, a new material model, **Magnetostrictive Material**, has been added. This material has three different formulations: **Linear**, **Nonlinear Isotropic**, and **Nonlinear cubic crystal**.

In the Magnetic Fields interface, the new **Ampere's law, Magnetostrictive** feature is used when modeling a magnetostrictive material.

The magnetostrictive coupling requires the AC/DC Module together with either the Structural Mechanics Module, Acoustics Module, or MEMS Module.

SERENDIPITY ELEMENTS

In the Solid Mechanics interface, elements with so-called *serendipity* shape functions have been added. The serendipity elements have fewer degrees of freedom than the standard Lagrange elements. The default element type in these interfaces has been changed to **Quadratic serendipity**. The element type in existing models is not changed when the model is opened, so if you want to employ the serendipity elements, you must change the element type in the **Discretization** section of the physics interface settings.

The serendipity shape functions affect hexahedral, prism, pyramid, and quadrilateral elements. For a model with only hexahedral elements, the solution time is typically decreased by a factor of two when serendipity elements replace Lagrange elements in the same mesh.

ADHESION AND DECOHESION

The **Contact** node has a new **Adhesion** subnode. When using adhesion, the contacting boundaries will stick together when a certain criterion has been fulfilled. This criterion can be either a contact pressure, gap distance, or arbitrary user-defined expression. Using adhesion requires a penalty contact formulation.

Two boundaries that are joined by adhesion can separate again if a decohesion law is specified. There are three different decohesion laws: **Linear**, **Polynomial**, and **Multilinear**. The decohesion laws allow mixed mode decohesion with independent properties for the normal and tangential directions.

NEW AND MODIFIED VARIABLES FOR CONTACT ANALYSIS

Total forces are now computed for each **Contact** node as well as summed over all **Contact** nodes.

In general, the postprocessing variables for contact analysis have been restructured so that they are available for both the individual and aggregated **Contact** nodes.

The friction slip velocity is no longer defined as a dependent variable. When running a model using the COMSOL API, you will need to remove the reference to this variable in the solver settings. The friction slip velocity is usually defined as `<comp>_<solid>_vslip_<pairname>`, where `<comp>` is the tag of the component, `<solid>` is the tag of the Solid Mechanics physics interface where the **Friction** node is defined, and `<pairname>` is the name of the contact pair.

NEW METHODS FOR ENTERING THERMAL EXPANSION DATA

There are now three different ways on which thermal expansion data can be entered:

- As a secant coefficient of thermal expansion. This is the default, and the only method available in previous versions.
- As a tangent (‘thermodynamical’) coefficient of thermal expansion.
- By explicitly specifying the thermal strain as a function of temperature.

By selecting the appropriate option, you can use different types of measured data without conversions. This new functionality is available in the Solid Mechanics interface.

THERMAL EXPANSION OF CONSTRAINTS

Constraints like **Fixed Constraint** and **Prescribed Displacement** can now be augmented with a **Thermal Expansion** subnode. This makes it possible to relieve the stresses induced by constraints when the surrounding structure, idealized by the constraints, is not held at a fixed temperature.

Similarly, a **Thermal Expansion** subnode has been added to the **Rigid Domain** and the **Rigid Connector**. This allows for a thermal expansion of these otherwise rigid objects.

HARMONIC PERTURBATION FOR PRESCRIBED VELOCITY AND ACCELERATION

The **Prescribed Velocity** and **Prescribed Acceleration** features have been augmented with a **Harmonic Perturbation** subnode. These boundary conditions can thus be used as a fixed constraint in a stationary study step, and then provide a harmonic vibration in a subsequent prestressed frequency domain study. This new functionality is available in the Solid Mechanics interface.

SMITH PLOTS

Smith plots are now included in the MEMS Module.

Backward Compatibility with Version 5.2

The friction slip velocity is no longer defined as a dependent variable. When running a model using the COMSOL API, you will need to remove the reference to this variable in the solver settings. The friction slip velocity is usually defined as `<comp>_<solid>_vslip_<pairname>`, where `<comp>` is the tag of the component, `<solid>` is the tag of the Solid Mechanics physics interface where the **Friction** node is defined, and `<pairname>` is the name of the contact pair.

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

New Functionality in Version 5.2a

It is now possible to reverse the orientation of the surface normal in the **Fluid-Fluid Interface** feature in the **Laminar Two-Phase Flow, Moving Mesh** interface. This avoids the ambiguity in the definition of the contact angle in the **Wall-Fluid Interface** subfeature.

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 $-Q_{\mathrm{br}}\cdot\mathbf{u}/\varepsilon_{\mathrm{p}}^{2}$ on the right-hand side previously lacked the factor $1/\varepsilon_{\mathrm{p}}^{2}$, where ε_{p} is the porosity.

MOBILITY

In physics interfaces modeling migration (for example, 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 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 5.2a

UPDATED PARTS IN THE MIXER MODULE PARTS LIBRARY

The axial, radial, and C-shaped impellers have been updated to include virtual surfaces used for calculations of the axial and radial flow numbers. The inclusion of upper, lower, and side control surfaces is enabled with a single Boolean parameter in the respective part.

Updated Application in Version 5.2a

MIXER APP

The Mixer app has been updated with a plot of the shear rate and evaluations of the radial and axial flow numbers for each impeller.

Molecular Flow Module

Backward Compatibility with Version 5.0 and Earlier

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 should be changed from `fmf.N` to `fmf.N_G`.
- The pressure should be changed from `p` to `fmf.p_G`.
- The reconstructed number density should be changed from `fmf.N` 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, and so on:

- 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 5.2a

NEW GEAR FUNCTIONALITY

New functionality is introduced in the Multibody Dynamics Module to model a system of gears for power transmission. There are several types of gears available to connect and transfer power from one shaft to another shaft. The results obtained from the gear dynamics can be further used for the fatigue analysis of a gear, or for an acoustic analysis to compute the noise generated by the system.

GEAR NODES

The following nodes are added in the Multibody Dynamics interface to model different types of gears and racks:

- **Spur Gear**
- **Helical Gear**
- **Bevel Gear**
- **Worm Gear**
- **Spur Rack**
- **Helical Rack**

GEAR PAIR NODES

The following nodes are added in the Multibody Dynamics interface to connect different types of gears and racks:

- **Gear Pair**
- **Worm and Wheel**
- **Rack and Pinion**

GEAR PAIR SUBNODES

The following subnodes are added on gear pair nodes to incorporate the effects of gear mesh elasticity, transmission error, backlash, and friction in the gear dynamics:

- **Gear Elasticity**
- **Transmission Error**

- **Backlash**
- **Friction**

GEAR PARTS LIBRARY

A new gear parts library is added in the Multibody Dynamics Module to create geometries of different types of gear. The gear geometries are available for 3D and 2D models. All of the geometries are parametrized and the input parameters can be varied to customize the gear tooth or gear blank shape.

The following gear geometries are available in the parts library:

- Spur gear (external and internal)
- Helical gear (external and internal)
- Bevel gear
- Worm gear
- Spur rack
- Helical rack
- Gear tooth (external and internal)
- Rack tooth

New Applications in Version 5.2a

DIFFERENTIAL GEAR

This example illustrates the modeling of a differential gear mechanism, commonly used in automobiles. The differential gear allows the outer drive wheel to rotate faster than the inner drive wheel during a turn. This is necessary when the vehicle turns, making the wheel that is traveling around the outside of the turning curve roll farther and faster than the other.

In this model, five sets of straight bevel gears are used to model the differential mechanism. The speed ratio of the two output shafts is varied to simulate the motion of a vehicle on a straight as well as a curved path.

GEAR TRAIN

This example illustrates the modeling of a gear train where shafts are mounted on an elastic housing. A transient analysis is performed to compute the angular velocity and acceleration of all of the gears, contact forces on the gear pairs, and the normal acceleration of the housing.

The gear mesh is assumed elastic and the detailed contact model is set up for the computation of gear mesh stiffness. A parametric analysis is performed to compute the gear mesh stiffness as a function of gear rotation in one mesh cycle.

BEVEL GEAR PAIR

This is a verification example where the obtained results are compared with the results from the reference.

This example illustrates the modeling of a straight conical bevel gear pair. The gears are modeled as rigid. One of the gears is fixed and the other is mounted on a rigid bar, which is free to rotate about the center of the fixed gear. A transient analysis is performed to compute the forces and moments experienced by the fixed gear.

HELICAL GEAR PAIR

This is a tutorial example that illustrates the modeling of a helical gear pair. The gear mesh is assumed rigid and elastic in two different cases. A transient study is performed to analyze the effect of gear mesh stiffness as well as the joint stiffness on the contact force and the speed of the gears. An eigenfrequency analysis is also performed to compute the natural frequencies and mode shapes of a gear pair for both rigid and elastic gear mesh.

Nonlinear Structural Materials Module

New Functionality in Version 5.2a

LARGE STRAIN VISCOELASTICITY

It is now possible to add **Viscoelasticity** to a **Hyperelastic Material** to simulate large strain viscoelastic applications.

MIXED ISOTROPIC AND KINEMATIC HARDENING

The plasticity models have been updated so that isotropic and kinematic hardening models can be used simultaneously.

NEW ISOTROPIC HARDENING MODELS

The set of isotropic hardening models in plasticity has been extended with four new built-in models: **Ludwik**, **Swift**, **Voce**, and **Hockett-Sherby**.

NEW KINEMATIC HARDENING MODELS

The kinematic hardening in plasticity has been extended with the **Armstrong-Frederick** and **Chaboche** models.

TIME DERIVATIVES OF PLASTIC STRAINS

Time derivatives of the components of plastic strain are now available, so it is possible to use expressions like `d(solid.epe,TIME)` to create strain-rate-dependent plasticity models.

NEW VARIABLES FOR HYPERELASTIC STRAIN ENERGIES

New variables have been added for incompressible hyperelastic material models: the volumetric and isochoric strain energy densities. The volumetric strain energy density is usually defined as `<solid>.Wsvol`, where `<solid>` is the tag of the Solid Mechanics interface where the **Hyperelastic material** node is defined. The isochoric strain energy density is defined as `<solid>.Wsiso.`

When using the **Nearly incompressible** formulation, the new variable `<solid>.Uvol` is defined as `Wsvol` was defined in previous versions. The variable `Wsiso` is now defined in terms of the auxiliary pressure `<solid>.pw`.

New variables for the isochoric second Piola-Kirchhoff stress tensor components have been added. These variables are derived from the isochoric strain energy density, and used in the definitions for large strain viscoelastic material models.

Optimization Module

New Functionality in Version 5.2a

CONSTRAINT METHOD SETTING IN POINTWISE INEQUALITY CONSTRAINTS

In version 5.2, the method used for assembling pointwise inequality constraints was silently changed from elemental to nodal. This, in practice, removed a lot of redundant constraints. Including each constraint once instead of several times in an optimization problem does not change its solutions, but the SNOPT solver's termination criterion is sensitive to the number of constraints. Therefore, in order to allow direct comparisons between solutions in different versions, a Constraint method setting has been added to the **Pointwise Inequality Constraint** features. The default method is still Nodal. Choose Elemental to recreate the behavior from versions older than 5.2.

DEFAULTS CHANGED IN GLOBAL LEAST-SQUARES OBJECTIVE

When the experimental Data source in a **Global Least-Squares Objective** feature is set to Result table, no table will be selected by default. In previous versions, the first table in the list under **Results>Tables** was arbitrarily used as default.

Similarly, the default for the Time column and Parameter column settings are now None instead of the first column in the selected table.

These changes lead to more predictable behavior when deleting tables and table columns. Some Java® and MATLAB® code recorded in previous versions may be affected, producing an error message when executed. The solution in these cases is to insert lines explicitly setting the desired table and column. For example:

```
model.physics("opt").feature("glsobj1").set("table", "tbl1");
model.physics("opt").feature("glsobj1").set("parameterColumn",
"1");
```

TIME-DEPENDENT LEAST-SQUARES OPTIMIZATION

In time-dependent least-squares optimization problems, time lists are modified while solving if they contain time values that are outside the time range defined by the time list of the corresponding Time Dependent study. The modified time lists do not contain time values that are outside of the time range defined by the corresponding Time Dependent study. Such modified time lists are used in the optimization iterations when least-squares objectives are evaluated. Before, the modified least-squares time lists were used only in the last iteration. Hence, running such a model can now return a solution that differs in this way. It is, however, possible to use the unmodified least-squares-defined time list by setting the new property of Time-Dependent Solver, Exclude times outside General parameter value lists, to off.

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 Multiphysics where you have selected that check box.

Particle Tracing Module

New Functionality in Version 5.2a

RESTRUCTURED PARTICLE TRACING FOR FLUID FLOW CONTEXT MENU

The context menu for the Particle Tracing for Fluid Flow interface has been reorganized to improve readability.

All forces are now included in a separate **Forces** submenu. Similarly, if the **Compute particle temperature** check box is selected, all heat sources are included in a separate **Thermal** submenu.

FIRST-ORDER NEWTONIAN FORMULATION

It is now possible to select **Newtonian, first order** from the **Formulation** list in the settings window for any of the particle tracing physics interfaces. Whereas the **Newtonian** formulation defines second-order equations for the components of each particle's position vector, the first-order Newtonian formulation defines coupled first-order equations for the particle position and velocity components.

The first-order Newtonian formulation supports all of the same physics features as the second-order formulation but is easier to combine with explicit time-stepping methods. When the first-order Newtonian formulation is chosen, the default time-stepping method is a high-order Runge-Kutta method instead of the typical second-order implicit method. This can lead to faster performance in some types of nonstiff problems.

STORE EXTRA TIME STEPS FOR WALL INTERACTIONS

It is now possible to automatically store extra time steps in the solution, in addition to the steps that are usually specified in the study settings. Usually these extra time steps are automatically selected to be close to times at which particle-wall collisions occur. This often enables the **Particle Trajectories** plot to convey information about particle-wall interactions more clearly. The storage of extra time steps is activated by selecting the **Store extra time steps for wall interactions** check box in the settings window for any of the particle tracing physics interfaces.

PARTICLE BEAM IMPROVEMENTS

The **Particle Beam** feature for the Charged Particle Tracing interface contains several new options in version 5.2a.

Four options are now available from the **Sampling from phase space ellipse** list — **Gaussian**, **KV**, **Waterbag**, and **Parabolic**.

New options are available for the **Longitudinal velocity distribution** — **None**, **Normal**, **Uniform**, and **List of values**. In addition, the options **Normal**, **Uniform**, and **List of values** multiply the number of particles released.

Due to some changes in the algorithm for initializing particle positions, the initial positions of particles released by the **Particle Beam** feature are no longer mesh dependent, so it is no longer necessary to create an extremely fine mesh on the releasing boundaries.

NEW TURBULENT DISPERSION OPTIONS

New options are available to account for the effects of turbulence when tracing particles in a fluid. In the setting for the **Drag Force** feature for the Particle Tracing for Fluid Flow interface, the following options are now available in the Turbulent dispersion model list — **None**, **Discrete random walk**, **Discrete random walk**, **variable time step**, and **Continuous random walk**. The options **Discrete random walk**, **variable time step**, and **Continuous random walk** take the average eddy lifetime into account when seeding random velocity perturbations to produce better statistical convergence of the particle distribution when the time step taken by the solver is sufficiently small.

DROPLET BREAKUP FEATURE

It is now possible to model the breakup of liquid droplets in a flow. The **Droplet Breakup feature** is available when the **Compute particle mass** check box is selected in the settings window for the Particle Tracing for Fluid Flow interface.

Backward Compatibility with Version 5.2

TURBULENT DISPERSION CHANGES

In the settings window for the **Drag Force** for the Particle Tracing for Fluid Flow interface, the **Turbulent dispersion** check box has been replaced by the **Turbulent dispersion model** list. If the check box is selected in a model created in version 5.2 or earlier, then after opening the model in version 5.2a, the option **Discrete random walk** will be selected from the list. If the check box is cleared, then **None** will be selected from the list.

PARTICLE DENSITY SPECIFICATIONS MERGED

In version 5.2, when specifying particle properties using the **Particle Properties** node or **Override Properties** node in the Particle Tracing for Fluid Flow interface, it is possible to specify two different values of the particle density, depending on whether the **Compute particle mass** check box is selected. In version 5.2a, these two values have been merged into a single parameter value; that is, the value or expression for the **Particle density** will be retained when selecting or clearing the **Compute particle mass** check box in the physics interface **Advanced Settings** section.

PARTICLE BEAM CHANGES

The names of several settings have been changed in the **Particle Beam** feature for the Charged Particle Tracing interface.

In the **Sampling from phase space distribution** list, **Uniform** has been renamed **KV**.

In the **Longitudinal velocity distribution** list, **Uniform** has been renamed **None** and **Gaussian** has been renamed **Normal**.

The parameters **Position refinement factor** and **Release distribution accuracy order** have been removed. Due to some changes in the algorithm for initializing particle positions, in version 5.2a, the distribution of particle positions no longer depends on the finite element mesh and may differ from the distribution in version 5.2.

Backward Compatibility with Version 5.1

RENAMED PHYSICS INTERFACE SETTINGS

The **Release type** list in the settings windows for the Charged Particle Tracing and Particle Tracing for Fluid Flow interfaces in earlier versions has been renamed **Particle release specification**. For the Charged Particle Tracing interface, the options **Static** and **Transient** have been renamed to **Specify current** and **Specify release times** (the default), respectively. For the Particle Tracing for Fluid Flow interface, the options **Static** and **Transient** have been renamed to **Specify mass flow rate** and **Specify release times** (the default), respectively.

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 or later, 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 5.2a

NEW OPTION FOR PUMP FEATURE FOR THE PIPE FLOW INTERFACE

An additional option for the **Pump** feature allows to specify a pump curve (head vs volumetric flow rate). The pump curve can be entered either as an expression or by using a table. Pump curve data can be imported from a text file.

NEW OPTIONS FOR INLET FEATURE FOR THE PIPE FLOW INTERFACE

The inlet feature is enhanced with the following options:

- A pump option is available for using pump curves at inlets. The pump curve can be entered either as an expression, or using a table.
- The reservoir option allows you to account for the loss of energy that occurs when a liquid enters a pipe from a large tank or reservoir.

PHYSICS SYMBOLS FOR BOUNDARY CONDITIONS

For the Pipe Flow interface, the physics symbols are available for internal point boundary conditions such as Bend, Valve, Contraction/Expansion, Pump, T-Junction, Y-Junction, and N-Way Junction.

NEW OPTION FOR HEAT SOURCE FEATURE FOR THE HEAT TRANSFER IN PIPES INTERFACE

A new option for defining the heat source is now available. This options allows you to specify the heat rate.

Backward Compatibility with Version 5.1 and older

REVISED FORMULATION OF THE PIPE FLOW INTERFACE

The Pipe Flow interface has been amended in version 5.2 with an inertial term. The stabilization and element shape functions have also been changed. Models created in previous versions will be opened using the old interface with a notification. For new models, only the new interface is available in the physics interface list.

Plasma Module

New Functionality in Version 5.2a

The DC Discharge and Capacitively Coupled Plasma interfaces have been replaced and consolidated into a new interface called Plasma. The Plasma interface includes all the functionality of the DC Discharge and Capacitively Coupled Plasma interfaces. Upon opening a model containing a DC Discharge and Capacitively Coupled Plasma interface, it is converted to the new Plasma interface. The original interface Label, Name, and Tag are retained for backward compatibility.

The Inductively Coupled Plasma and Microwave Plasma interfaces have been reimplemented as multiphysics interfaces combining the Plasma and Magnetic Fields interfaces and Plasma and Electromagnetic Waves, Frequency Domain interfaces, respectively. A table summarizing these changes is as follows:

TABLE 1-4: SUMMARY OF CHANGES TO THE PHYSICS INTERFACES IN THE PLASMA MODULE

OLD INTERFACE NAME (PRIOR TO V5.2A)	NEW INTERFACE NAME(S)
DC Discharge	Plasma
Capacitively Coupled Plasma	Plasma
Microwave Plasma	Plasma Electromagnetic Waves, Frequency Domain
Inductively Coupled Plasma	Plasma Magnetic Fields

Coupling between the high-frequency fields and plasma for the Inductively Coupled Plasma and Microwave Plasma interfaces is now done using multiphysics coupling features. The **Plasma Conductivity Coupling** feature computes the electric conductivity required by the Magnetic Fields or Electromagnetic Waves, Frequency Domain interfaces based on the electron density, collision frequency, and angular frequency, using the cold plasma approximation. The **Electron Heat Source** feature computes the collisional heating of electrons based on the plasma conductivity and electric field.

Backward Compatibility with Version 5.2 and Earlier

The old Inductively Coupled Plasma and Microwave Plasma interfaces can still be opened, edited, and solved. When such a model is opened, a warning is given indicating that the interface is obsolete and will eventually be removed.

Backward Compatibility with Version 5.1

Models using the Reduced Electric Fields study will interpret the unit in the edit field for the Reduced electric fields as Townsend. For example, when opening a model created in version 5.1, if the value for the Reduced electric fields was 10, this will now appear as `10[Td]` in the study settings.

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 5.2a

RAY TRACING OUTSIDE THE GEOMETRY

It is now possible to trace rays outside the model geometry and in domains that are not included in the selection for the Geometrical Optics physics interface, as long as all such domains have the same homogeneous refractive index. In this case, domains outside the selection of the Geometrical Optics interface do not need to be meshed. However, any boundaries that the rays interact with must still be meshed.

Because it is now possible to release and trace rays outside the geometry, the Wall condition is no longer the default boundary condition on exterior boundaries. Instead, the Material Discontinuity condition is used.

To revert to the old (version 5.2 and earlier) behavior, in which rays can only propagate through meshed domains in the selection of the physics interface, select **Advanced Physics Options** in the Model Builder and then clear the **Allow propagation outside selected domains** check box (which is selected by default) in the settings window for the Geometrical Optics interface.

DEDICATED POSTPROCESSING FEATURES FOR MONOCHROMATIC ABERRATIONS

The **Optical Aberration** plot can be used to compute monochromatic aberrations in an imaging system by expanding the optical path difference among rays in terms of a basis of Zernike polynomials. Similarly, the **Aberration Evaluation** feature can be used to create a table of Zernike coefficients.

EXTRA POINTS IN THE RAY TRAJECTORIES PLOT

A number of additional data points are now automatically stored in the solution, in addition to the solution times or optical path length increments that are specified in the study settings. These additional solution times typically correspond to points at which rays are reflected or refracted at boundaries.

As a result, plots that are created in version 5.2a may look slightly different than plots that were created in version 5.2 or earlier. In version 5.2a, the rays are more likely to extend all the way to the reflecting or refracting surface, even if the exact instant of reflection or refraction would not otherwise coincide with a stored solution time.

NEW OPTIONS FOR CONE-BASED RELEASE

When releasing a conical distribution of rays, it is now possible to select an option from the **Conical distribution** list — **Uniform density**, **Specify polar and azimuthal distributions**, **Marginal rays only**, or **Marginal and axial rays only**. These options affect the distribution of the released rays in the wave vector space.

Backward Compatibility with Version 5.2

ANISOTROPIC REFRACTIVE INDICES CAN NO LONGER BE SPECIFIED

In version 5.2 and earlier, it is possible to specify an anisotropic refractive index in the settings window for the **Medium Properties** node, despite the fact that the anisotropy of the refractive index is not taken into account when modeling ray propagation. In version 5.2a, only isotropic refractive indices can be specified in the settings window for the **Medium Properties** node. If an anisotropic refractive index was specified in an older version, then when opening the model in version 5.2a, all of the diagonal terms will be populated using the first diagonal value and all off-diagonal terms will be set to zero.

RENAMED INTENSITY COMPUTATION METHODS

The options in the **Intensity Computation** list in the settings window for the Geometrical Optics interface have been renamed.

- **Using principal curvatures** has been renamed to **Compute intensity**.
- **Using principal curvatures and ray power** has been renamed to **Compute intensity and power**.
- **Using curvature tensor** has been renamed to **Compute intensity in graded media**.
- **Using curvature tensor and ray power** has been renamed to **Compute intensity and power in graded media**.

RENAMED POLARIZATION SETTING

In the **Initial polarization type** list, available for most ray release features in the Geometrical Optics interface, the option **Partially coherent** has been renamed to **Partially polarized**.

CHANGES TO PERTURBATIONS FOR SOLAR RADIATION

The treatment of perturbations due to sunshape and surface roughness has been changed in version 5.2a. This change affects the **Illuminated Surface** and **Solar Radiation** features for the Geometrical Optics interface.

The **Use corrections for finite source diameter** check box has been replaced by the **Corrections for finite source diameter** list. If the check box was selected, **Create light cones at release points** will be selected from the list after opening the model in version 5.2a. If the check box was cleared, **None** will be selected from the list.

For the **Illuminated Surface**, the **Standard deviation of the angle of incidence** parameter has been renamed **Surface slope error** and is now used to define a Rayleigh distribution for perturbations to the surface normal, rather than the angle of incidence of radiation. To account for the different effect that the surface slope error has on the model, any value or expression entered for the **Standard deviation of the angle of incidence** in version 5.2 or earlier will be divided by 2, before being used to specify the **Surface slope error** in version 5.2a.

The **Number of rays in wave vector** space $N_{w,r}$ has been removed. The effect of surface roughness can no longer multiply the number of degrees of freedom created by a release feature at each release point.

ACTIVATION CONDITIONS ON OPTICAL COMPONENTS

The following boundary conditions require some form of ray intensity computation in order to have any effect on the solution:

- **Linear Polarizer**
- **Linear Wave Retarder**
- **Circular Wave Retarder**
- **Ideal Depolarizer**
- **Mueller Matrix**

If any of these features is present in a model created in version 5.2 or earlier, and **None** is selected from the **Intensity computation** list in the settings window for the Geometrical Optics interface, then the feature will be automatically disabled when the model is opened in version 5.2a. To enable these features, select any option other than **None** from the **Intensity Computation** list.

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 5.2a unless this operator is applied.

THIN DIELECTRIC FILM SETTINGS

In version 5.0, if the ray intensity is computed, the option to add **Thin Dielectric Film** nodes to a **Material Discontinuity** node is always available. In version 5.2a, 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 5.2a, 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 corresponds 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 5.2a, 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 5.2a

POSTPROCESSING FAR-FIELD VARIABLES FOR BISTATIC RADAR CROSS SECTION

Postprocessing variables are added for the bistatic radar cross section (RCS) calculation. You can use these in far-field plots to visualize the radar cross section that is a measure of the size of a scatterer seen by a radar. The bistatic RCS variable `bRCS3D` describes the RCS measured through a transmitter and receiver that are located separately, and you can also plot monostatic RCS. For 2D models, you can model bistatic radar cross section per unit length with the operator `bRCS2D`.

The Application Library path for the example plotting the bistatic RCS is:

`RF_Module/Verification_Examples/rcs_sphere`

The Application Library path for the example plotting the monostatic RCS is:

`RF_Module/Scattering_and_RCS/radar_cross_section`

TWO-PORT NETWORK SYSTEMS

The two-port network feature characterizes the response of a two-port network system, such as reflection and transmission, using S-parameters. Just like the lumped port feature, a two-port network can only be applied on boundaries that extend between two metallic boundaries — that is, boundaries where Perfect Electric Conductor, Impedance Boundary, or Transition Boundary conditions apply — separated by a distance much smaller than the wavelength. A pair of two-port network port subnodes are added by default to the two-port network node and are used to select boundaries corresponding to port 1 and port 2 in the S-parameter input, respectively.

FAST MODELING APPROACH FOR BANDPASS FILTER TYPE DEVICES

Two powerful simulation methods new to the RF Module have been implemented in existing Application Library examples for designing bandpass-filter-type high-Q devices. When designing RF devices using the finite element method (FEM) in the frequency domain, it is easy to encounter a situation requiring many frequency samples to describe the passband more accurately. The simulation time is directly proportional to the number of frequencies included during the simulation of a microwave device;

the finer the frequency resolution the study settings use, the longer it takes to finish the simulation. The new asymptotic waveform evaluation (AWE) and Frequency-Domain Modal methods provide a simulation speed orders of magnitude faster than the conventional frequency sweeping.

The Application Library path for the example using the asymptotic waveform evaluation (AWE) method is:

```
RF_Module/Passive_devices/cylindrical_cavity_filter_evanescent
```

The Application Library path for the examples using the Frequency-Domain Modal method is:

```
RF_Module/Passive_devices/cascaded_cavity_filter
```

```
RF_Module/Passive_devices/coupled_line filter
```

```
RF_Module/Passive_devices/cpw_bandpass_filter
```

New and Updated Models and Applications in Version 5.2a

LOG-PERIODIC ANTENNA FOR EMI/EMC TESTING

A log-periodic antenna is known to be a wideband or frequency-independent antenna. One of the most popular forms of such an antenna is the coplanar dipole array type, which is commonly used for UHF-range antenna measurements in an anechoic chamber. This example computes the voltage-standing-wave-ratio (VSWR) and far-field radiation pattern between 250 MHz and 750 MHz.

SIGNAL INTEGRITY (SI) AND TIME-DOMAIN REFLECTOMETRY (TDR) ANALYSIS OF ADJACENT MICROSTRIP LINES

A signal integrity (SI) analysis gives an overview of the quality of an electrical signal transmitted through electrical circuits, such as high-speed interconnects, cables, and printed circuit boards. The quality of the received signal can be distorted by noise from outside the circuit and can be degraded by impedance mismatch, insertion loss, and crosstalk. In practice, EMC/EMI analyses are run to estimate the susceptibility of a device or a network to an undesired coupling. In this example tutorial, we examine the crosstalk effect between two adjacent microstrip lines on a microwave substrate. The simulated results provide the time-domain reflectometry (TDR) response at the coupled ports and show increased distortion of a signal at higher data rates.

PLASMONIC WIRE GRATING SIMULATOR

The upgraded plasmonic wire grating simulator app, which computes coefficients of refraction, specular reflection, and first-order diffraction for a wire grating on a dielectric substrate, shows the enhanced view of electric field magnitude for multiple grating periods for a selected angle of incidence, now in 3D.

Backward Compatibility with Version 4.3a

The following models have been moved from the RF Module Application Library to the Wave Optics Module Application 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 5.2a

ENHANCED PERFORMANCE FOR THE IDEAL SCHOTTKY BOUNDARY CONDITION AT METAL CONTACTS

In version 5.2 and prior, a constant extrapolation scheme is used at metal contacts for the ideal Schottky boundary condition. This requires a very refined mesh at the boundary to produce results with acceptable accuracy. In version 5.2a, a high-order extrapolation scheme has been added to achieve much better accuracy without the need for an extremely dense mesh at the boundary.

IMPROVED PERFORMANCE FOR THE THERMIONIC EMISSION BOUNDARY CONDITION AT HETEROJUNCTIONS

In version 5.2 and prior, a constant extrapolation scheme is used at heterojunctions for the Thermionic Emission boundary condition. This requires a very refined mesh at the boundary to produce results with acceptable accuracy. In version 5.2a, a high-order extrapolation scheme has been added to achieve much better accuracy without the need for an extremely dense mesh at the boundary.

ENHANCED CAPABILITY FOR THE CONTINUOUS QUASI FERMI LEVEL BOUNDARY CONDITION AT HETEROJUNCTIONS

Fermi-Dirac statistics are now supported for heterojunctions with continuous quasi-Fermi-level boundary conditions.

MORE ACCURATE FORMULATION FOR ELECTROSTATICS OF ADJACENT CHARGE CONSERVATION DOMAINS

The electrostatics formulation for adjacent charge conservation domains is improved to obtain more accurate results.

OPTIMIZED STUDY SETTINGS TO SPEED UP COMPUTATION FOR THE BIPOLAR TRANSISTOR MODELS

The Study settings for the bipolar transistor tutorial models have been optimized to speed up computation. In version 5.2a, the 3D model takes hours rather than days. The 2D model takes minutes rather than over an hour.

New Application in Version 5.2a

SI SOLAR CELL WITH RAY OPTICS

This app combines the Ray Optics Module and the Semiconductor Module to illustrate the operation of a silicon solar cell at a location specified by the user. The Ray Optics Module computes the average illumination over a day of the year and a location chosen by the user. The Semiconductor Module computes the normalized output characteristics of a solar cell with design parameters specified by the user. The normalized output characteristics are then multiplied by the computed average illumination to obtain the output characteristics of the cell at the specified date and location, assuming a simple linear relationship between the output and the illumination.

Structural Mechanics Module

New Functionality in Version 5.2a

MAGNETOSTRICTION INTERFACE

A new Magnetostriction interface has been introduced. When added, a Solid Mechanics interface, a Magnetic Fields interface, and a **Magnetostriction** multiphysics coupling are created.

In the Solid Mechanics interface, a new material model, **Magnetostrictive Material**, has been added. This material has three different formulations: **Linear**, **Nonlinear Isotropic**, and **Nonlinear cubic crystal**.

In the Magnetic Fields interface, the new **Ampere's law, Magnetostrictive** feature is used when modeling a magnetostrictive material.

The magnetostrictive coupling requires the AC/DC Module together with either the Structural Mechanics Module, Acoustics Module, or MEMS Module.

NEW POROELASTICITY INTERFACE

A new **Poroelasticity** multiphysics coupling between Solid Mechanics and Darcy's law has been introduced. When adding a Poroelasticity interface in version 5.2a, the two separate physics interfaces and the multiphysics coupling will be created. This will give access to all functionality available in the constituent interfaces. As an example, it is now possible to model poroplasticity.

The Poroelasticity interface existing in previous versions is now obsolete. Old models containing this interface can still be opened and used. In the future, it will be removed.

SERENDIPITY ELEMENTS

In the Solid Mechanics and Membrane interfaces, elements with so called *serendipity* shape functions have been added. The serendipity elements have fewer degrees of freedom than the standard Lagrange elements. The default element type in these interfaces has been changed to **Quadratic serendipity**. The element type in existing models is not changed when the model is opened, so if you want to employ the serendipity elements, you must change the element type in the **Discretization** section of the physics interface settings.

The serendipity shape functions affect hexahedral, prism, pyramid, and quadrilateral elements. For a model with only hexahedral elements, the solution time is typically decreased by a factor of two when serendipity elements replace Lagrange elements in the same mesh.

PERIODIC CONDITIONS FOR SHELLS

A **Periodic Condition** boundary condition node has been added to the Shell interface. This feature, which is similar to the one in Solid Mechanics, allows for efficient modeling of periodic structures by coupling corresponding edges. There are five different selections for the type of periodicity: **Continuity**, **Antiperiodicity**, **Floquet periodicity**, **Cyclic symmetry**, and **User defined**.

ADHESION AND DECOHESION

The **Contact** node has a new **Adhesion** subnode. When using adhesion, the contacting boundaries will stick together when a certain criterion has been fulfilled. This criterion can be either a contact pressure, a gap distance, or an arbitrary user-defined expression. Using adhesion requires a penalty contact formulation.

Two boundaries that are joined by adhesion can separate again if a decohesion law is specified. There are three different decohesion laws: **Linear**, **Polynomial**, and **Multilinear**. The decohesion laws allow mixed mode decohesion with independent properties for the normal and tangential directions.

The Application Libraries example Mixed-Mode Debonding of a Laminated Composite has been revised to make use this new built-in functionality.

NEW AND MODIFIED VARIABLES FOR CONTACT ANALYSIS

Total forces are now computed for each **Contact** node as well as summed over all **Contact** nodes.

In general, the postprocessing variables for contact analysis have been restructured so that they are available both for the individual **Contact** node, and aggregated.

The friction slip velocity is no longer defined as a dependent variable. When running a model using the COMSOL API, you will need to remove the reference to this variable in the solver settings. The friction slip velocity is usually defined as `<comp>_<solid>_vslip_<pairname>`, where `<comp>` is the tag of the component, `<solid>` is the tag of the Solid Mechanics physics interface where the **Friction** node is defined, and `<pairname>` is the name of the contact pair.

NEW METHODS FOR ENTERING THERMAL EXPANSION DATA

There are now three different ways in which thermal expansion data can be entered:

- As a secant coefficient of thermal expansion. This is the default, and the only method available in previous versions.
- As a tangent ('thermodynamical') coefficient of thermal expansion.
- By explicitly specifying the thermal strain as a function of temperature.

By selecting the appropriate option, you can use different types of measured data without conversions. The new options are available in the Solid Mechanics, Membrane, and Truss interfaces.

THERMAL EXPANSION OF CONSTRAINTS

Constraints like **Fixed Constraint** and **Prescribed Displacement** can now be augmented with a **Thermal Expansion** subnode. This makes it possible to relieve the stresses induced by constraints when the surrounding structure idealized by the constraints is not held at a fixed temperature.

Similarly, a **Thermal Expansion** subnode has been added to the **Rigid Connector** and **Attachment** nodes. This allows for a thermal expansion of these otherwise rigid objects.

SHELL COORDINATE SYSTEM

The local coordinate system used in the Shell interface is now defined in a subnode **Shell Local System** under the **Linear Elastic Material**. This means that it is easier to use different material orientations with the same material data.

A new node, **Shell Local System**, is also created under **Definitions** when a Shell interface is added. This system contains the local directions for all boundaries on which the Shell interface is active, and can be referenced, for example, when setting up multiphysics couplings.

NEW DEFAULT SETTING FOR STRAIGHT EDGE CONSTRAINT

The **Straight Edge Constraint** in the Truss interface has a new control **Disable for linear discretization order**. The default state is selected. This gives a significant performance improvement for the common case when these constraints are not needed.

The default mesh in the Truss interface uses one element per edge, and then the straight edge constraints are not needed. Thus, there will not be any difference in behavior unless the mesh settings have been changed, in which case you may have to deselect **Disable for linear discretization order**.

HARMONIC PERTURBATION FOR PRESCRIBED VELOCITY AND ACCELERATION

The **Prescribed Velocity** and **Prescribed Acceleration** features have been augmented with a **Harmonic Perturbation** subnode. These boundary conditions can thus be used as a fixed constraint in a stationary study step, and then provide a harmonic vibration in a subsequent prestressed frequency domain study. This new functionality is available in the Solid Mechanics, Membrane, and Truss interfaces.

Backward Compatibility with Version 5.2

The friction slip velocity is no longer defined as a dependent variable. When running a model using the COMSOL API, you will need to remove the reference to this variable in the solver settings. The friction slip velocity is usually defined as `<comp>_<solid>_vslip_<pairname>`, where `<comp>` is the tag of the component, `<solid>` is the tag of the Solid Mechanics physics interface where the **Friction** node is defined and `<pairname>` is the name of the contact pair.

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

- When using version 5.2a, in the Solid Mechanics interface, the property `ControlsSpatialFrame` from version 4.3a no longer exists, and COMSOL Multiphysics 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 is set to the expression used in the model.

Subsurface Flow Module

New Functionality in Version 5.2a

NEW POROELASTICITY INTERFACE

A new **Poroelasticity** multiphysics coupling between Solid Mechanics and Darcy's law has been introduced. When adding a Poroelasticity interface in version 5.2a, the two separate physics interfaces and the multiphysics coupling will be created. This will give access to all functionality available in the constituent interfaces. As an example, it is now possible to model poroplasticity.

The Poroelasticity interface existing in previous versions is now obsolete. Old models containing this interface can still be opened and used. In the future, it will be removed.

GRAVITY FEATURE

A new **Gravity** feature is available in the Darcy's Law, Richards Equation, and Fracture Flow interfaces. The **Gravity** feature defines a volume force on the fluid due to the gravity in all the domains where the interface is active.

Two formulations are available to specify Darcy's velocity field under gravity: the **Acceleration** (default) and **Elevation** formulations. When the **Acceleration** formulation is selected, define the components of the gravity acceleration vector, while the **Elevation** formulation defines a reference elevation for the water table.

FLOWNET PLOT

The Richards' Equation interface in 2D and 2D axisymmetric geometries now generates a new 2D Plot Group called **Flownet**. This plot combines a streamline plot for Darcy's velocity field and a contour plot for the pressure head variable.

Backward Compatibility with Version 5.2

POROELASTICITY

The Poroelasticity interface has been replaced by the new multiphysics coupling Poroelasticity.

New Applications in Version 5.2a

PERFORATED WELL

Perforated wells are commonly used for oil or gas production, because of their advantage of a better control of the well by sealing perforations individually. To model the nonuniform fluid flow of such a perforated well, a full 3D analysis is required.

This application allows you to simulate the Darcy flow to a perforated wellbore for different well, reservoir, and fluid properties to predict the fluid intake at each single perforation.

TWO-PHASE FLOW COLUMN

A multistep outflow experiment for two-phase immiscible flow through porous media gives information about the flow and transport characteristics of the fluid soil system.

This application computes the capillary pressure and effective saturation for different fluid phases and soil combinations in different experimental setups. The relationship between capillary pressure, effective saturation, and relative permeability are based on the van Genuchten retention model.

Wave Optics Module

New Functionality in Version 5.2a

POLARIZATION DOMAIN FEATURE

The new **Polarization** domain feature facilitates coupling different frequency-domain interfaces together. This new feature simplifies simulations of nonlinear frequency mixing, like sum- and difference-frequency generation and nonlinear parametric processes. The **Polarization** feature is available for the Electromagnetic Waves, Frequency Domain and the Electromagnetic Waves, Beam Envelopes interfaces.

USER DEFINED WAVE VECTOR SPECIFICATION SECTION FOR THE ELECTROMAGNETIC WAVES, BEAM ENVELOPES INTERFACE

In the settings for the Electromagnetic Waves, Beam Envelopes interface, there is a new section called **User Defined Wave Vector Specification**. This section appears when the Type of phase specification parameter is set to User defined. Normally, the values for the wave vectors should be the default settings. However, this new section allows the specification of special settings in, for instance, perfectly matched layer (PML) domains.

ABSORBING LAYER FEATURE

The Electromagnetic Waves, Time Explicit interface now include an **Absorbing Layer** feature that is used to set up effective nonreflecting-like boundary conditions. The layers work by combining three techniques: a scaling system, filtering, and a simple nonreflecting boundary condition.

The layers are set up by adding the **Absorbing Layer** feature available under the **Definitions** node.

DIFFRACTION ORDER WITH ORTHOGONAL POLARIZATION IN 3D

For 3D models, the Components parameter for the **Diffraction Order** feature now also have the option Orthogonal polarization, for the mode numbers $m = n = 0$. When you choose Orthogonal polarization, the mode gets a polarization that is orthogonal to the polarization of the incident wave.

When the **Compute Diffraction Orders** button in the **Port Mode Settings** section for periodic ports is clicked, a mode with Orthogonal polarization for the mode numbers

$m = n = 0$ will be added if that mode is not evanescent for the other specified parameters in the **Port Mode Settings** section.

SURFACE HEATING SPLIT UP IN RESISTIVE HEATING AND MAGNETIC LOSS

For the Transition boundary condition and the Impedance boundary condition, the surface loss is divided into Surface resistive losses (the variable `ewfd.Qsrh`) and Surface magnetic losses (`ewfd.Qsml`).

New Models and Applications in Version 5.2a

SECOND HARMONIC GENERATION IN THE FREQUENCY DOMAIN

This is a proof of principle example, describing the second harmonic generation (SHG) process using two Electromagnetic Waves, Frequency Domain interfaces: one for the fundamental wave and one for the second harmonic. The coupling between the waves is implemented using a domain **Polarization** feature for each interface. The results are compared against the analytical solution from the Slowly Varying Envelope Approximation (SVEA).

SINGLE-BIT HOLOGRAM

This model simulates a bit-by-bit holographic data storage system, including data recording and retrieval. In the bit-by-bit holographic data storage system, the data is 1 bit, so the profiles of the object beam and the reference beam are identical in this model. In the recording process, the two beams intersect each other and make an interference fringe pattern, which is the hologram carrying the 1-bit data. During the retrieval process, the object beam is turned off and then the reference beam is diffracted by the hologram and creates the object beam.

POLARIZING BEAM SPLITTER

Polarizing beam splitter cubes consist of two right-angled prisms. A dielectric coating is evaporated on the hypotenuse prism surface. The dielectric coating is designed to transmit the part of the incident wave with the electric field polarized in the plane of incidence (p-polarization) and to reflect (in an orthogonal direction) the part of the incident wave with the polarization orthogonal to the plane of incidence (s-polarization). Two advantages of the cube design, compared to a plate design, are that ghost images are avoided, since there is only one reflecting surface here, and the translation of the transmitted output beam is very small, compared to the input beam.

The latter advantage simplifies the alignment of optical systems using polarizing beam splitter cubes.

This app demonstrates the basic MacNeille design. This design consists of pairs of layers having consecutively high and low refractive indexes. In the internal dielectric stack, the waves hit the layer boundary at the Brewster angle. Thus, the s-polarization is reflected, whereas the p-polarization is transmitted at each internal layer boundary. A common design is to use zinc sulphide (ZnS) for the high refractive index layers (n = 2.3) and magnesium fluoride (MgF2) for the low refractive index layers (n = 1.38), and SF5 glass prisms (n = 1.673).

The app demonstrates how the phase functions required by the Electromagnetic Waves, Beam Envelopes interface can be defined using variables and global equations from the Global ODEs and DAEs interface. The variable definitions and the global equations are modified dynamically, as the number of layers in the dielectric stack is changed.

Backward Compatibility with Version 5.2 and Earlier

PERFECTLY MATCHED LAYERS PHYSICS FEATURE NO LONGER AVAILABLE

The **Perfectly Matched Layer** physics feature that for version 5.2 and earlier versions was marked deprecated is no longer available. Thus, models using the **Perfectly Matched Layer** physics feature, must first be rebuilt in a version earlier than 5.2a to use the **Perfectly Matched Layer** feature from under the **Definitions** node, before being loaded in 5.2a.

NEW DEFAULT VALUE FOR GAUSSIAN BEAM BACKGROUND FIELD SPOT RADIUS

The Gaussian beam background field is based on the paraxial approximate solution to Helmholtz' equation. Since this approximation is less accurate for small spot radii, the default value for the spot radius has been changed from one wavelength to ten wavelengths.

PORT PHASE PARAMETER RENAMED AND RELOCATED

The parameter Port phase, which was previously located in the **Port Properties** section, is now called Mode phase and is located in the **Port Mode Settings** section. The API, however, has not changed.

BOUNDARY MODE STUDY SEQUENCES NO LONGER AVAILABLE IN THE MODEL WIZARD

Previously, when the Laser Heating multiphysics was selected, the Boundary Mode, Frequency-Stationary and Boundary Mode, Frequency-Transient study sequences were available from the Model Wizard. Now, these two study sequences are removed. Instead, a Boundary Mode Analysis step can be added manually to the Frequency-Stationary and Frequency-Transient studies.

Material Library

New and Updated Material Data in Version 5.2a

Data has been added for the following materials: C/C SiC composite; olive oil, apple pulp, tomato paste, meat, starch, and sodium caseinate; LaMnO3, La0.825Sr0.175MnO3, and La0.3Sr0.7MnO3; Al-Li and Zn-Li alloys; 1Cr-0.5Mo, H11 and Modified 9Cr-1Mo; Ga2Mg5 and Tl2Mg5; W-Ni-Fe-Cu heavy alloys; and L-605, Haynes 25, P9, P91, Mod. 9Cr-1Mo, and CaCO3.

The expansion data for 316 stainless was extended up to 1000 °C. The previous data above 20 °C was changed by less than 2%. Data below 20 °C was not affected.

Also, the thermal conductivity data for 1030 steel was approximately 40% higher than similar materials. The data and reference have been changed and are now in good agreement with similar materials. This change affects the thermal conductivity for the following materials: 1030, A36, A106, A201, A302, and A333.

LiveLink™ for MATLAB®

New Functionality in Version 5.2a

LiveLink™ *for* MATLAB® now supports MATLAB® R2016a and R2015b.

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

MPHEVAL, MPHGLOBAL, AND MPHINTERP FUNCTIONS

The property `'complexout'` is now `'on'` by default, which means that any complex results will now be returned by default. If you wish to have the previous behavior, you should set `'complexout'` to `'off'`.

MPHPLOT FUNCTION

Support is added for the following plot types: Directivity Plots, Optical Aberration Plots, and Whirl Plots.

These plot types have been improved: 2D Tube plots, wireframe rendering for Volume Plots. Surfaces and lines are ordered such that lines appear on top of surfaces. Support for filled legend types have been added.

The new color tables AuroraAustralis, AuroraBorealis, HeatCamera, HeatCameraLight, JupiterAuroraBorealis, and Twilight have been added.

To improve usability, the error message returned after specifying the wrong plot group tag now includes a list of valid tags.

MPHEVALUATE FUNCTION

The `mphevaluate` function now accepts expressions as a vector matrix. Results are returned as an array of the same size as the expressions.

A new option `'asvalue'` has been added to return results as an array of structs, or a cell array of strings.

MPHEVALGLOBALMATRIX FUNCTION

The option `'trans'` now supports additional values for matrix transformation. New valid values are `'maxwellmutual'` and `'mutualmaxwell'` in addition to `'none'`, `'inverse'`, `'sy'`, `'sz'`, `'ys'`, `'yz'`, `'zs'`, and `'zy'`.

MPHSTATE AND MPHMATRIX FUNCTION

It is now possible to specify at what step in the solution sequence the matrices should be extracted. By using the new property `'extractafter'`, it is possible to extract the initial load vector before running the solver. The matrices can then be manipulated and reinserted with mphinputmatrix.

MPHNAVIGATOR FUNCTION

Error and warning nodes as well as icons for all nodes are now displayed in the model tree.

The new buttons **Copy set** and **Copy get** have been added for copying text to the clipboard on how to set or get information from settings and parameters in the model.

MPHSHOWERRORS FUNCTION

Multiple levels of errors are extracted from models. An extra column is added to the output. This column contains a cell array of the model tree nodes that contain the error information. Having this information makes it easier to extract error information for automated processing of error and warning conditions.

Backward Compatibility with Version 5.2

The functions `mpheval`, `mphglobal`, and `mphinterp`, which extract numerical data, now support complex values by default; that is, the property `'complexout'` is `'on'` by default. If you want to have the previous behavior, set `'complexout'` to `'off'`.

Backward Compatibility with Version 5.0

- The `mphint2` command: For data sets other than Solution, Particle, Cut Line, Cut Plane, 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 solution 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®

Supported Versions

Excel® 2007 will no longer be supported from the next version of COMSOL Multiphysics.

New Functionality in Version 5.2a

CONNECT TO SERVER AND SERVER INTERACTIONS

Connecting to a server, other than what is automatically opened, is now more stringent with COMSOL Multiphysics. The user can select **Connect to Server** and specify connection settings in the **Connect to Server** window. If the client in LiveLink™ *for* Excel® is connected to a server and the user selects **Open**, the model will be opened on the connected server. If no server is connected and the user selects **Open** an automatic server will be launched and the model will be loaded on the started server, similar to the behavior in version 5.2. It is also possible to import an application from the server and remove applications from the server.

SUPPORT FOR LAUNCHING APPS

It is now possible to run an application and launch the COMSOL Server interface from LiveLink™ *for* Excel®. The COMSOL Server interface will be displayed in a separate window.

PARAMETERS

- Parameter values that are imported into a worksheet may now include the values in addition to the expressions that define the parameters. The values can be updated on the fly in order to evaluate a set of parameters before starting a simulation.
- A new preference setting, called Automatic update, has been introduced for parameters. When this setting is selected, editing the cells for the imported parameters will automatically update the model, and the values for the imported parameters.

CHANGES TO THE INTERPRETATION OF CELL RANGES

When using a cell range, such as "A1:B20", this cell range is now used to decide exactly what cells are used for the import. The old method just used the first cell and continued until an empty row was found. The old method is still supported if only a single cell is specified ("C5", for example).

The COMSOL API for Use with Java®

COMSOL 5.2a API Changes

AC/DC MODULE CHANGES

New Coil Feature

In the Magnetic Fields, Magnetic and Electric Fields, and Rotating Machinery, Magnetic interfaces, a new unified **Coil** feature has replaced the **Multi-Turn Coil** feature. The functionality of the old **Multi-Turn Coil** feature can be replicated by using the new **Coil** feature with the Conductor model set to Homogenized Multi-Turn, but all of the other parameters are unchanged. The only change required is to the create statements such as:

```
model.physics("mf").feature().create("mtcd1",
"MultiTurnCoilDomain", 3);
```

which should be modified to:

```
model.physics("mf").feature().create("mtcd1", "Coil", 3);
model.physics("mf").feature("mtcd1").set("ConductorModel",
"Multi");
```

User-Defined Coil Geometry Subfeature Selection

In 3D models, when using a user-defined geometry for a **Coil** feature, the **User Defined Coil Geometry** subfeature has been transformed from a boundary feature to a domain feature. New **Input** and **Output** subnodes are used to select the coil input or output boundaries instead. Any code setting the selection of this feature should be modified to set the selection on the **Input** subfeature instead.

Relative Permeability for Magnetic Shielding in Magnetic Fields, No Current

The relative permeability parameter `mur` in the **Magnetic Shielding** feature in the Magnetic Fields, No Currents interface is now a scalar quantity. Setting a tensor quantity will not cause exception, but only the first component of the tensor will be used as the value.

ACOUSTICS MODULE CHANGES

The default settings have changed for the Incident Pressure Field in the Pressure Acoustics interfaces. Add the following line to obtain the old behavior of this feature when `ipf1` is a subfeature to `pwr1`:

```
model.physics("acpr").feature("pwr1").feature("ipf1").
set("c", "acpr.c_c");
```

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 setting 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("reac1").set("R_0", new
String[]{"R1"});
model.physics("tcdee").feature("reac1").set("R_1", new
String[]{"R2"});
model.physics("tcdee").feature("reac1").set("R_2", new
String[]{"R3"});
```

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

```
model.physics("tcdee").feature("reac1").set("R_c1", new
String[]{"R1"});
model.physics("tcdee").feature("reac1").set("R_c2", new
String[]{"R2"});
model.physics("tcdee").feature("reac1").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("Convect
iveTerm", "noncons", 0);
```

For information about API backward compatibility for the Surface Reactions interface, see the Chemical Reaction Engineering Module release 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 the types *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 were deprecated in COMSOL 5.0:

TABLE 1-5: 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 1-5: DEPRECATED METHODS AND THEIR REPLACEMENTS

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

TABLE 1-5: DEPRECATED METHODS AND THEIR REPLACEMENTS

DEPRECATED METHOD	NEW METHOD
`com.comsol.model.ParameterEntity.set(String, int, String)`	Use the 0-based version (`setIndex`) instead.
`com.comsol.model.ParameterEntity.set(String, int, String[]`	Use the 0-based version (`setIndex`) 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 and later. 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 API 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 5.2 and later: `mod1_solid_contact_old_p1`, `mod1_solid_cm_old_p1`.
  - In version earlier than 5.0: `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 the Pressure Acoustics, Frequency Domain and Pressure Acoustics, Transient interfaces. 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 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 Multiphysics version 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_lm` now create a vectorial Lagrange multiplier field with default name `model.u_lm`. The default component names in 3D are `model.u_lm`, `model.v_lm`, and `model.w_lm`, 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 *<phtag>*`.locCFL` to the built-in variable `CFLCMP`, which in turn triggers a PID regulator via the automatic solver suggestions. Here, *<phtag>* 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("CFLNumbExpr"
  , 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();
```

COMSOL 4.2a API Changes

- Class files compiled with COMSOL Multiphysics version 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.relEx`). Instead, the scattered field is only available in the interface scope as, for example, `mod1.emw.relEx`.
- In the Solid Mechanics interfaces (`SolidMechanics`, `PiezoelectricDevices`, `AcousticStructure`, `Poroelasticity`, `ThermallyInducedStresses`, `JouleHeatingThermalExpansion`, `TransientAcousticStructure`, and `FluidStructureInteraction`), tensors in local coordinate systems (`el`, `eel`, `Sl`, `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, `elX2` is replaced by `el22` and `elx2x3` is replaced by `el23`. 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("dl").feature("smm1").set(chif_mat,userdef);
  model.physics("dl").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("dl").feature("smm1").set(chif_mat,userdef);
model.physics("dl").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 Multiphysics, 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 versions 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("dl").feature("dlm1").set(kappa_mat,userdef);
model.physics("dl").feature("dlm1").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 versions 4.0 and 4.0a:

- The units of the load face parameter (`Fph`) of the **Phase** feature in `AcousticStructure`, `TransientAcousticStructure`, `ThermallyInducedStresses`, `SolidMechanics`, `JouleHeatingThermalExpansion`, `Poroelasticity`, and `FluidStructureInteraction` have 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 5.2aa.

Index