COMSOL Multiphysics

Release Notes

5.3

COMSOL

COMSOL Multiphysics Release Notes

Version: COMSOL 5.3

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

1

Release Notes

COMSOL Multiphysics® version 5.3 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 5.3 for all COMSOL products.

COMSOL Multiphysics

General New Functionality

IMPROVED PERFORMANCE AND USER INTERFACE

The performance for many operations in the COMSOL Desktop has been improved, and the undo/redo-functionality now works in almost all cases. Specifically, operations in the geometry and meshing sequences now fully support undo and redo.

PREFERENCES TAKEN FROM PREVIOUS VERSION

By default, the installation of version 5.3 reuses most preference settings from the previous installation, so that you do not need to redo changes to the factory settings for your specific preferences.

New Functionality in the Application Builder

GENERAL NEW FUNCTIONALITY

- The *Application Builder Reference Manual*, which contains reference documentation previously available as context help only, is now available as a PDF file from the **Documentation** window, which you open from the COMSOL Desktop.
- The **Editor Tools** window now includes toolbar buttons for expanding and collapsing all nodes of the displayed tree.
- Explicit selections now appear in the **Editor Tools** window and on the **Graphics** tab in the **New Form Wizard**. Adding an explicit selection to a form will place a **Graphics** form object in that form window and set the explicit selection as its source. It is also possible to add a **Selection Input** form object by clicking the **Input** button in the **Editor Tools** window while having an explicit selection node selected.
- Export features can now be added to forms via the **Editor Tools** window and the **New Form Wizard**.
- **Model Data Access** has been renamed to **Data Access**, and is now available also for properties in the application part of the model. Only properties that can be modified when an application is running are available for data access. **Data Access** is now also available on the **Home** toolbar.
- For **Choice List** nodes added under **Declarations**, you can now load and save list values from a file as well as clear the table of list values.

IMPROVEMENTS TO THE FORM EDITOR

- Support for data picking in Graphics windows. It is now possible to select points and evaluate data in geometries and plots via mouse interaction with graphics form objects. You can enable data picking in the **Settings** window for **Graphics** objects. There is also a new **Graphics Data** node under **Declarations**, where you can specify settings for design time values for coordinate values, plot evaluation, and 3D probe options.
- The **Slider** form object now has an orientation property allowing you to make vertical sliders.
- In a form editor you can now select form objects and cut them by pressing Ctrl+X or choosing **Cut** from the context menu.
- Form objects set as invisible or disabled now have a distinct visual style in the form editor. Also, form objects that have their visible setting set to false are not made invisible in the form editor if they are selected or part of a multiselection.

IMPROVEMENTS TO THE METHOD EDITOR AND APPLICATION LANGUAGE

See also the *Application Programming Guide* for a complete overview of the application language for programming methods in the Application Builder.

- You can now create *application methods*, for use in applications like in earlier versions, and *model methods*, which are new in this version. Model methods can be used in the Model Builder as a complement to the model tree for automating special sequences or tasks.
- When hovering the mouse pointer over a property name, a declaration, or a shortcut in a method editor window, a tooltip containing information about the property name, declaration, or shortcut is displayed. When hovering the mouse pointer over a name corresponding to an entity in the application part of the API (such as a form or a form object), information about what type of object the name represents (such as **Input Field**, **Button**, **Form**, or **Menu**) is displayed in a tooltip.
- It is now possible to automatically declare the type of a local variable. For example, you can type `x = model.geom()` and click the **Create Local Variable** button on the **Method** ribbon tab. The code is then changed to `GeomList x = model.geom()`.
- The **Extract Variable** button on the **Method** ribbon in previous versions has been replaced with a new button called **Use Shortcut**. Clicking the **Use Shortcut** button creates a shortcut corresponding to the extracted code instead of a hidden instance variable, which was what clicking the **Extract Variable** button created in previous versions.

- New application language methods have been introduced to create, load, and remove models without having to specify a model tag. The create and load methods generate a unique tag for the new model, and the remove method takes a model object instead of a model tag. The syntax is:

  ```
  Model createModel();
  void removeModel(Model model);
  Model loadModel(String filename);
  Model loadProtectedModel(String filename, String password);
  Model loadRecoveryModel(String folderName);
  ```

- Two new standard commands, Select All and Clear Selection, have been added to the Application Builder. The commands are similar to their Model Builder versions but work on a target **Graphics** form object. Two new methods, `selectAll` and `clearSelection`, have been added to the application API. In the Application Builder, the new commands appear in command sequences (for a button, for example) under **GUI Commands>Graphics Commands**. The syntax for the new methods:

  ```
  void clearSelection(FormObject graphics);
  void clearSelection(String graphics);
  void selectAll(FormObject graphics);
  void selectAll(String graphics);
  ```

- A **Go to Source** button has been added to the **Settings** window for local methods, which, when clicked, causes the form object, which the local method belongs to, to be selected in the form editor.
- Ctrl+/ is now an alternative shortcut for auto completion in the method editor.

New Functionality in COMSOL Server

- COMSOL Server now include support for reverse proxies.
- In the **Application Library**, there is more live data information displayed for running applications. On the **Preferences** page, you can select to display up to five of the following information items: Session time, Connected in, Idle time, Process CPU, Start time, CPU time, Physical memory, and Virtual memory. All those items appear in the details view of a running app.
- The **Monitor** page now have two views: **Servers**, showing monitoring data for all COMSOL Server processes (primary and secondary server processes) as well as the application server processes running on these COMSOL Servers, and **Sessions**, showing monitoring data for all currently running application sessions. From both views, administrators and power users can close application sessions for users.

- You can now upload multiple apps and use drag-and-drop to do so.
- The **Preferences** page has been improved. Changes are indicated using a red asterisk, and new **Revert to Saved** and **Factory Settings** buttons are available.
- **Prelaunching** on the **Preferences** page in previous versions has been renamed **Application Processes**. An administrator can now choose to allow applications to run on both the primary server and on secondary servers, or only run on secondary servers, when running in a primary-secondary server setup. Furthermore, the settings for prelaunching are now per COMSOL Server instance instead of per primary- secondary cluster of servers. The options for prelaunched application server processes have also been extended.
- The **Preferences** page now also includes preferences for cluster computing and remote computing.
- A log file with data for application usage is now available. The data is stored in a text format so that it can be analyzed in Microsoft® Excel®, for example.

New and Improved General Functionality in COMSOL Multiphysics

- There is a new **Developer** ribbon tab in the Windows version of the COMSOL Desktop. It contains tools for creating application methods and model methods, running model methods, and creating and testing applications. Some of those tools were previously available on the **Home** ribbon tab. Also, the **Add Multiphysics** button is no longer available on the **Home** ribbon tab but is available on the **Physics** ribbon tabs as well as on the **Windows** menu. The **Add Component** and **Select Component** buttons on the **Home** tab have been merged into a single **Component** menu button.
- On the **Developer** tab, you can run model methods, which you create in the Application Builder but can use in the Model Builder as a complement to the model tree for automating some tasks when modeling in COMSOL Multiphysics.
- A new *PDE, Boundary Elements* interface provides the possibility to solve unbounded PDEs of Laplace type using the *boundary element method*.
- Under **Classical PDEs,** a *Stabilized Convection-Diffusion Equation* interface is now available. In that version, streamline, crosswind, and isotropic diffusion can be used to stabilize the convection-diffusion equation when needed. Streamline and crosswind diffusion are active by default.
- In the Solid Mechanics interface, a new **Rigid Motion Suppression** boundary condition has been introduced for handling cases where the loads are self-equilibrating as long as rigid body motions are not possible and no reaction

forces are introduced. This new condition removes the need to constrain rigid body motion by inserting point constraints in such cases.

The **Symmetry** boundary conditions are also available for the Solid Mechanics interface in this version without any additional module requirement.

- The **Help** and **Documentation** windows have been updated and now includes a breadcrumb trail at the top, indication the location in the documentation set for the information that is displayed.
- For physics interfaces that use symbols to indicate, for example, boundary conditions, you can now control the general display of physics symbols and also show and hide all physics symbols in the physics interface using the **Physics Symbols** section in the **Settings** window for the main physics interface node. When physics symbols are enabled, you can also control their display individually in the settings for each physics interface node that supports symbols.
- The hide functionality in the graphics windows has been simplified. The **View** ribbon tab now has a single **Hide** button instead of, in previous versions, separate **Hide for Geometry**, **Hide for Physics**, and **Hide for Mesh** buttons. This button adds the type of hide feature suitable for the current context.
- The **Selection List** window now indicated which geometric indices that are selected.
- The default font handling has been improved to better handle non-Latin fonts. You can now specify the font to use directly in the installer when installing using a non-Latin (Asian) language.
- The local table for interpolation functions, parameters, variables, polygons, and interpolation curves now includes a **Clear Table** button.
- For model inputs, a **Go to Source** button is now available for moving to the node in another physics interface that provides the input.
- The **COMSOL Launchers** folders in the version 5.3 installation now includes dedicated launchers for all applicable graphics rendering options (DirectX, OpenGL, and software rendering).
- Triple-clicking in text fields now selects the whole line in all versions of the COMSOL Desktop.
- On macOS, Command-. (that is, the Command button and period) is now available for stopping a method as an alternative to Ctrl+Pause.

New Geometry and Mesh Functionality

GEOMETRY FUNCTIONALITY

General Improvements

In version 5.3, there are significant improvements in execution time, memory usage, and disk usage for parts of the functionality in the geometry sequences.

New Functionality

- Coordinate systems in the geometry: In geometric primitives and transform features in 3D, you can specify the settings in a local coordinate system given by a work plane. This makes it easier to position new objects relative to existing objects.
- Coordinate systems from geometry under **Definitions**. A new coordinate system type, **System from Geometry**, creates a coordinate system from a work plane. This makes it easy to align a coordinate system with geometric entities. Some other coordinate system types can also be defined relative to a work plane.
- Combined coordinate systems under **Definitions**. You can use the new **Combined System** node to create a coordinate system that has different definitions on different domains. This is useful when an anisotropic material has different orientations in different domains.
- A new virtual operation, **Remove Details**, automatically generates a sequence of virtual operations that removes details smaller than a given size. The Remove Details node is available on the **Virtual Operations** menu. The generated sequences of virtual operations are possible to edit if needed.
- Extrude enhancements: You can now extrude until you hit a selected vertex and extrude in both directions at the same time.
- A new **Line Segment** feature that constructs a line segment between selected vertices or given point coordinates.
- Enhancements for the **Cylinder Selection** and **Disk Selection** features: It is now possible to specify inner radius and sector angles.
- In the **Cross Section** feature, you can now generate 2D selections from 3D selections.
- In the **Sweep** feature, you now get better results when sweeping along a chain of edges, thanks to the new **Parameterization** and **Smooth edge connections** settings.
- When visualizing a work plane, its local coordinate system triad `xw`, `yw`, `zw` is also visualized.
- It is now possible to select a geometry object in draw mode and then delete it by pressing the Delete key.

- For geometry part variants, there is a new **Show as variant in part library** check box in the settings of a geometry part. This new check box makes it possible to have an MPH file that contains different variants of a part. The variants typically differ only in the set of input parameters. When loading the part, a dialog box appears where you can select which variant you want to use. In version 5.3, this functionality is used by parts in the Ray Optics part library.
- For import of geometry using MPHBIN or MPHTXT files, there is a new possibility to exclude result from virtual operations in order to make sure that the imported geometry is not virtual.

MESHING FUNCTIONALITY

- Automatic transitions with pyramid elements are now available when interfacing a swept hex or prism mesh with a tetrahedral mesh. There is no longer a need for converting a swept mesh before meshing the surrounding domain with the Free Tetrahedral operation.
- There is a new **Size Expression** attribute node, which you can use to define a space-varying mesh size based on an expression using the spatial coordinates.
- There is also a new **Adapt** node, which you can use to perform mesh adaptation. Such **Adapt** nodes are created automatically in the mesh refinement meshing sequences when using mesh adaptation from a study step.
- There is a new optimization option in the Free Tetrahedral operation that you can use if you want to maximize the size of the smallest element while still trying to respect the desired local element size. This optimization can improve the performance when solving problems using explicit time stepping.

- You can now select between several quality measures when evaluating a mesh using the **Statistics** window or a mesh plot:
  - *Skewness*, which is the new default mesh quality measure in the COMSOL Desktop and for the API.
  - *Maximum angle*
  - *Volume versus circumradius* (the previous default mesh quality measure and the only quality measure available in earlier versions)
  - *Volume versus length*
  - *Condition number*
  - *Growth rate*

  The growth rate mesh quality measure replaces the previously reported values of the maximum and average growth rate in the Statistics window.

  The value for the minimum element quality is now based on the skewness measure. Inverted mesh elements, however, are considered as inverted regardless of the selected mesh quality measure.
- The meshing performance for 2D geometries with many domains has improved significantly, mainly through parallelization. Furthermore, quad meshing of domains and planar faces with four right-angled corners now gives more efficient and higher-quality quad meshes.
- New settings are available in the **Import** operation for controlling the partitioning of an imported 3D mesh on the edge level. This can be important if you want to combine an imported mesh with geometry objects.
- A new button in the **Graphics** window toolbar allows you to switch off the rendering of the mesh. This makes it easier to see the interior of a 3D object regardless of what part of the geometry that has a mesh.
- It is now possible to copy the mesh from the entire geometry and to copy 1D meshes.
- If you get an error referring to a coordinate in 3D, a new **Center at Coordinate** button appears, which zooms to the corresponding location in the **Graphics** window. In addition, a small red sphere (wireframe rendering) appears around the coordinate where the meshing error is located.
- An issue where the mesh generated might add an unexpectedly refined mesh when meshing domains has been fixed. This change may affect the results in old models if you rebuild the mesh (in apps with dynamic remeshing, for example). To avoid

remeshing caused by a physics-controlled mesh, you can switch to a user-controlled mesh.

New and Updated Operators, Functions, and Definitions

- You can now use projection coupling operators with all mesh types.
- A new operator, `atdomat`, is available for evaluation of mesh-independent expressions in unmeshed domains in a given position. It is primarily intended for use in void regions, for models using the boundary element method.
- You can now better control the scaling of coordinates used for unstructured interpolation data in cases when such unstructured interpolation data can be entered. Use the **Internal scaling of data points** setting to control this scaling. The automatic option applies the scaling if the bounding box of the interpolation points has a bad aspect ratio.
- Some of the values for the physical constants have been updated slightly to conform to the CODATA internationally recommended 2014 values. The *COMSOL Multiphysics Reference Manual* contains the updated values for all supported physical constants.

New Functionality in Studies and Solvers

STUDIES AND STUDY STEP FUNCTIONALITY

- The mesh adaptations and error estimation functionality has been combined and extended for more flexible adaptive mesh generation. A new **Adaptation and Error Estimates** section is now available in the **Settings** window for stationary, eigenfrequency, and frequency domain study steps, where you can choose to perform mesh adaptation and compute error estimates or to compute error estimates only. The adaptive mesh refinement steps are available as meshing sequences (with **Adapt** or **Size Expression** nodes, depending on the type of adaptation or error estimation). Also the solutions (and corresponding data sets) for the adaptive mesh refinement steps are available so that you can inspect and postprocess each step in the adaptation process. The previous **Adaptive Mesh Refinement** subnode to a **Stationary Solver** node is now added automatically and does normally not need to be changed.

  The time-dependent mesh adaptation works in the same way as in previous versions.
- A new **Combine Solutions** node makes it possible to combine two solutions (from the same study or from different studies) into a new solution — for example, to

concatenate two time-dependent solutions into one solution that can be postprocessed in its entirety for all time steps in both original solutions.

- A new **Stationary Source Sweep** study step is available for users of the AC/DC Module.
- In the **Physics and Variables Selection** section of the study steps' **Settings** window, when you select the **Modify physics tree and variables for study step** check box, a new **Control Frame Deformation** button makes it possible to control which physics interface that controls the spatial frame, for example, in models that included moving meshes.

SOLVER FUNCTIONALITY

- An improved aggregated AMG solver using a smoothed aggregation multigrid methods is now available and is possible to use as an alternative to GMG for fluid-flow models, for example.
- The solver log now contains more information about the use of available cores and used and available memory during the solution process.
- A new option for the direct solvers makes it possible to automatically switch to out-of-core storage if the estimated memory (for LU) is exhausting the physically available memory.
- For small eigenvalue problems, it is now possible to specify that all eigenvalues are returned from the solver.
- The scaled absolute tolerance is now proportional to the relative tolerance by default.
- The FFT solver can now handle larger data sets. If needed, you can save intermediate FFT data to disk to reduce the memory requirements. You can also choose **Times to store** with the options **Time steps taken by solver** (the default) and **From list**. In the latter case, the output data is interpolated. This way, you can decrease the amount of data stored in MPH files.
- For manual time stepping using the BDF method, you can now control the BDF order and time steps in the **Time Stepping** section.
- In parametric sweeps and batch sweeps, continuing the solver and getting the initial value are now supported. The initial value for a parametric sweep uses the first parameter value.
- For a batch process, you can now stop it after a given time from the command line using the option `-stoptime` *`<time to stop in seconds>`*.
- The Intel MKL has been upgraded to Intel MKL 2017 and IMPI 2017.

NEW AND IMPROVED GRAPHICS AND PLOT FUNCTIONALITY

- For 1D graph plots, you can now use two y-axes for plotting two quantities with different magnitudes, for example. You can freely choose which graph to plot to be represented by the primary (left) y-axis and which plot to be represented using the secondary (right) y-axis.
- Add a new 3D plot, **Streamline Surface**, is available for plotting streamlines on surfaces in 3D.
- Units have been added to the coordinate axes in plots.
- Also, the color legend can now show the unit of the quantity it represents. Select the **Show units** check box in the **Color Legend** section of the settings for plots.
- A **Coordinate system** list have been added to the settings for vector plots of revolutions of 2D axisymmetric geometries and for 2D vector plots on cut planes. It is now easier to plot vector quantities using coordinate systems defined by the data sets.
- It is now easier to step between solutions in a parametric sweep or an eigenfrequency or time-dependent simulation, for example. Use the **Plot Previous** and **Plot Next** buttons (or press F6 and F7) at the top of the **Settings** window for a 2D or 3D plot group to step forward or backward in the set of solutions and update the plot.
- For many plot types, including Line, Surface, Volume, Arrow, Contour, Isosurface, Slice, Max/Min and Streamline plots, you can now add a **Selection** subnode to apply the plot only to some selected geometric entities (for example, to make a surface plot on a subset of the boundaries).
- The **Goto XY View**, **Goto YZ View**, and **Goto ZX View** buttons in **Graphics** and plot windows now toggle through all four possible variants of those views.
- A new **Export Expressions** subnode is available for **Streamline**, **Far Field**, **Point Trajectories**, and **Ray Trajectories** plot nodes. You can use it to evaluate additional expressions to include in a plot data export (for example, both the real and imaginary parts of data used in far-field plots).
- You can now position the legend box in 9 positions. New positions are lower middle, center, and upper middle.
- The x- and y-axes in 1D plots can not be flipped if that switch is desired for better visualization.

- For arrow surface plots, you can now choose to plot all vector components, as in previous versions, or to plot the normal components — only the vector components orthogonal to the surface are plotted — or the tangential components — only the vector components parallel to the surface are plotted.
- You can now restrict streamline start positions to parametric curves and surfaces.
- You can now choose **Copy Plot Data to Clipboard** from the context menu of the individual plot nodes to directly copy the data in plots to the clipboard without adding an Export node.
- You can now easier control the length of streamlines in the **Advanced** section in the settings for streamline plots by entering the maximum length of each streamline in the **Maximum streamline length** field.
- It is now possible to synchronize the radius and color expressions in 3D far-field plots by selecting the **Use as color expression** check box at the bottom of the **Expression** section.
- Scatter plots now include a **Range** section for specifying the color and data range manually.
- For Max/Min plots, in the **Advanced** section, you can now add an optional prefix and suffix to the maximum and minimum values.
- For the view scale in the **Camera** node settings, you can now elect **Anisotropic** to define an anisotropic automatic scaling (a block) using different relative weights in the x-, y-, and z-direction. An anisotropic scaling can, for example, improve waterfall plots.
- For the Particle Tracing Module, particle trajectories can now be visualized using ribbons as the line style.
- For acoustics and RF applications, 1D **Far Field** plots now include settings for computing the beam width.

IMPROVED DATA EXPORT AND ANIMATIONS

- A **File type** list has been added to the setting for the **Data** and **Plot** export features for easier specification of the exported data file format.
- Is is now easier to export data from parametric sweeps. The **Export>Data** node can export data for several outer solutions, not only one, and it is easier to select parameter values for which to export data.
- A new **Summation** section in the settings for **Point Evaluation** nodes makes it possible to compute the sum or average of the value of a variable evaluated in several points.

- In the **Global Matrix Evaluation** node, **From inverse Maxwell to mutual capacitance** is a now transformation option.
- A new sequence type, **Global parameter**, makes it possible to animate over global parameters in the model for more flexible animations (animating over view parameters to simulate camera-based animation, for example).

NEW AND IMPROVED DATA SETS

- The **Parameterized Curve** data set now support shells, so that you can use it in models with physics on shells. In 3D, the **Parameterized Curve** data set also includes a new **Snap to closest boundary** check box.
- It is now possible to use 1D data in 2D and 3D line plots; the 1D data is embedded into 2D or 3D.
- A new **Array 1D** data set is available, which you can use, for example, when the model simulates only one unit cell out of an infinite periodic structure, and you would like to plot the result in more than one unit cell.
- In the Array data sets, you can now clear a **Check for overlap between cells** check box in the **Advanced** section to disable the cell overlap test, which makes it possible to create a hexagonal lattice by composing two linear arrays, for example.
- You can now use Microsoft® Excel® files in cut point data sets.
- For the Structural Mechanics Module, a new **Shell** data set makes in easy to visualize the top and bottom surfaces of a shell.

IMPROVED REPORT GENERATION

- It is now possible to turn off image generation in reports.

New Java® API Methods

- The first COMSOL instance that opens a file locks it and is the only instance that can save the file again. To check if a file is locked so that it is read-only, use the new methods `model.isReadOnly()`, which returns whether the file where the model is saved is read-only and cannot be overwritten or not. The file can be read-only for two reasons:
  - The COMSOL process does not have permission to write to the file.
  - On Windows, the file can be locked by another COMSOL instance.

  If the model has not been saved, this method returns false.

- `ModelUtil` has the following new methods for product license information and checkout:
  - `checkoutLicense`
  - `checkoutLicenseForFile`
  - `checkoutLicenseForFileOnServer`
  - `hasProduct`
  - `hasProductForFile`
  - `hasProductForFileOnServer`

General Backward Compatibility Considerations

COMSOL 5.3 is backward compatible with COMSOL versions 4.0–5.2a. The support for loading models made in version 3.5a has been removed in this version. The command `comsol convertpre35a`, which converted even older models to version 3.5a, has also been removed.

Backward Compatibility with Version 5.2a

FRAME CONTROL

In previous versions of COMSOL Multiphysics, it was possible to apply multiple frame-controlling physics on the same selection. For example, it was possible to add two Solid Mechanics interfaces (with the **Include geometric nonlinearity** option enabled) on the same selection and solve. In this case, an "override rule" was applied between the controlling physics, effectively meaning that the last physics in the tree had control of the frame deformation for the overlapping domains. This situation could cause unexpected results.

In version 5.3, multiple physics interfaces controlling the same frame are not allowed on the same selection and will cause an error when trying to solve. However, it is still possible to use multiple frame-controlling physics with overlapping selection, but you now have to explicitly disable frame control on all but one of the physics. This gives you better control on what is going on. The functionality to disable frame control for physics has been incorporated in the **Settings** window for study steps to allow choosing different physics to control the frame in different study steps.

Frame-Scoped Variables

Certain variables pertaining to frame deformation are now defined with "frame prefix". For example, the relative element volume is now called `spatial.relVol`.

Previously, these variables were added by the physics interface controlling the frame deformation so they used a "physics prefix" (for example, `ale.relVol`). For backward compatibility of old models, the physics interfaces still define the old variables, but they are aliases of the new frame variables.

API SYNTAX FOR ERROR HANDLING

The old syntax for an error

```
feature("foo").feature("prob1").feature("error1")
```

is no longer supported. Instead, use the following documented syntax:

```
feature("foo").problem("error1")
```

FREE QUAD MESHING

Models saved in version 5.2a with the free quad tessellation method set to **Automatic** (in a **Free Quad** feature) or face meshing method set to **Quadrilateral** (in a **Swept** feature), will, when opened in 5.3, have the tessellation method or face meshing method set to **Legacy version 5.2a** or **Quadrilateral (legacy version 5.2a)**, respectively.

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 5.3

ELECTROSTATICS, BOUNDARY ELEMENTS INTERFACE

The new *Electrostatics, Boundary Elements* interface is based on the boundary element method and available in 2D and 3D. It can be used for modeling electrostatics in unbounded domains. Coupling to finite-element-based physics is supported.

STATIONARY SOURCE SWEEP STUDY

The new Stationary Source Sweep study is a custom study for fast calculation of lumped parameters. It is available for the Electrostatics; Electrostatics, Boundary Elements; and Electric Currents interfaces. This study reuses the LU decomposition for direct solvers, making such studies many times faster than before.

CONCATENATED FLUX METHOD FOR INDUCTANCE CALCULATIONS

The new concatenated flux methods is a highly accurate integral method for solid and homogenized multiturn coils.

COIL FEATURE UPDATES

An issue with excessive memory usage for circuit-coupled coils has been fixed.

New and Updated Application in Version 5.3

The following new and updated applications and example models are available in the AC/DC Module Application Library:

- The Touchscreen Simulator app has been updated to use the faster Stationary Source Sweep study.
- The Capacitor Tunable model has been updated to use the new Electrostatics, Boundary Elements interface.
- A tutorial series on three-phase submarine cable modeling.
- Axisymmetric model of a 3D inductor.
- 3D model of a permanent magnet motor
- 3D power switch with motion
- Topology optimization of a loudspeaker magnet

- Operational Amplifier With Capacitive Load model
- Capacitive position sensor model in two versions using finite elements and boundary elements

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 Functionality in Version 5.3

The following new functionality is available:

- Perfectly matched layers are now available for the Pressure Acoustics, Transient Interface.
- A new Thermoviscous Acoustics, Transient interface.
- The Poroelastic Waves Interface has been updates with a new Biot-Allard model to include both thermal and viscous losses.
- The Convected Wave Equation, Time Explicit interface is now supported in 2D axisymmetric geometries.
- Many physics interfaces and multiphysics couplings now provide predefined iterative solver suggestions.
- The transient solver is now controlled from the physics nodes in a new Transient Solver Settings section.
- Improved stabilization methods for the Linearized Navier-Stokes Physics Interfaces.
- Serendipity elements are now available in most acoustics interfaces.
- Preview evaluation plane functionality for far field and directivity.
- Beam width calculation functionality for 1D far-field plots.
- Updated interior perforate plate boundary condition in the Pressure Acoustics, Frequency Domain interface.
- Heat source for the Pressure Acoustics interface.
- Improved ray acoustics in 2D axisymmetric geometries.
- New reflection coefficient models for wall features in ray tracing.
- New ray termination feature.

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 5.3

The following new functionality is available:

- A new Electrophoretic Transport interface.
- A new Transport of Diluted Species in Fractures interface. See the release notes for the Chemical Reaction Engineering Module for more information.
- New and updated default plots in the Electrochemistry interfaces:
  - New Electrode Potential vs. Ground plots
  - New Electrode Potential vs. Adjacent Reference plot
  - Added arrows to Electrolyte Potential plots
  - New Boundary Potential vs. Ground plot (for time-dependent studies)
  - Added Real and Imaginary Impedance vs. Frequency (for frequency-domain studies)
  - New Average SOC plot (in Lithium-Ion Battery and Battery With Binary Electrolyte interfaces for time-dependent studies)
  - Added positive/negative ion flux arrows to Concentration plot (in 2D and 3D for Lithium-Ion Battery and Battery With Binary Electrolyte interfaces)
- New **Ion-Exchange Membrane** node in the Tertiary Current Distribution, Nernst-Planck Interface. The updated *Vanadium Redox Flow Battery* tutorial uses the new node.
- New **Charge Conservation Model** options in the Tertiary Current Distribution, Nernst-Planck Interface.
- New **Thin Electrode Layer** node. This node replaces and extends the Thin Insulating Layer node of previous versions.
- In **Porous Electrode** nodes: Changed the default to **Bruggeman** transport correction in the Tertiary Current Distribution, Nernst-Planck; Lithium-Ion Battery; and Battery With Binary Electrolyte interfaces.
- New **Fracture** node in the Transport of Diluted Species in Porous Media Interface.
- For **Porous Electrode** nodes in the Lithium-Ion Battery and Battery with Binary Electrolyte interfaces: Moved the Particle Intercalation Settings to a new **Particle Intercalation** subnode.

- New **No spatial gradients** option for the transport model in intercalating **Porous Electrode** nodes' **Particle Intercalation** subnodes in the Lithium-Ion Battery and Battery With Binary Electrolyte interfaces.
- New **Lithium Insertion Reaction** node with simplified reaction kinetics in the Single Particle Battery interface.
- Updated the Reacting Flow in Porous Media Multiphysics interfaces.
- The **Reacting Flow** multiphysics node now supports coupling between Brinkman Equations and Transport of Concentrated/Diluted Species. The **Reacting Flow** multiphysics node adds mass sources/sinks from the Porous Electrode Coupling node in the Transport of Concentrated interface to the Brinkman Equations interface.
- New **Charge-Discharge Cycling** boundary node, applicable to external electrode and porous electrode domains. Updated the *Capacity Fade of a Lithium-Ion Battery* tutorial to use the new node.
- New **Charge discharge cycling** boundary condition option in the **Electrode Surface** node.
- New **Charge discharge cycling** operation mode in the Single Particle Battery interface.
- Updates to the Batteries & Fuel Cells material library:
  - Updated value for the reference (maximum) solid lithium concentration in the NMC electrode material.
  - Added KOH Binary Electrolyte. Updated the *1D Isothermal Nickel-Metal Hydride Battery* tutorial to use this new material.

For users with licenses for both the Electrodeposition Module and the Batteries & Fuel Cells Module: The **Nondeforming Boundary** and **Deforming Electrode Surface** multiphysics features now support couplings to the **Electrode Surface** node in the Lithium-Ion Battery and Battery with Binary Electrolyte interfaces.

For users with a license for the AC/DC Module:

- New **Circuit Terminal** boundary node, applicable to external boundaries to electrode and porous electrode domains.
- New **Circuit terminal** boundary condition in the Electrode Surface node.
- New **Circuit terminal** condition operation mode in the Single Particle Battery interface.

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 5.3

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 5.3 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 5.3

The LiveLink™ interface now supports synchronizing wireframe geometry.

New Functionality in LiveLink™ for Inventor® in Version 5.3

For assemblies, the LiveLink™ interface now automatically synchronizes selections that are defined in the component files. Previously, all selections had to be defined in the synchronized assembly file.

New Functionality in LiveLink™ for PTC® Creo® Parametric™ in Version 5.3

The LiveLink™ interface now supports PTC® Creo® Parametric™ 4.0.

New Functionality in LiveLink™ for Revit® in Version 5.3

The LiveLink™ interface now supports synchronizing geometry for Revit® projects without room definitions, and supports synchronizing elements that do not belong to a room. The added functionality also expands the type of elements that can be synchronized to include, for example: wall layers, structural framing and foundation elements, mechanical and electrical equipment. Selections, which can be used to define simulation settings, are automatically generated for the synchronized elements.

LiveLink™ for Solid Edge®

NEW FUNCTIONALITY IN VERSION 5.3

The LiveLink™ interface now supports Solid Edge® ST9.

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 5.3

- The LiveLink™ interface now supports SOLIDWORKS® 2017.
- For assemblies, the LiveLink™ interface now automatically synchronizes selections that are defined in the component files. Previously, all selections had to be defined in the synchronized assembly file.
- The new tutorial *Stress Analysis of a Pipe Fitting* has been added to demonstrate how to set-up a simulation on a 2D cross-section of a 3D geometry that has been synchronized 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 5.3

CONSERVATIVE FORMULATION FOR THE NAVIER-STOKES EQUATION

The new formulation of the momentum equation conserves momentum exactly in all cases. Slight changes in the results may be observed for some applications as previous versions of COMSOL Multiphysics only guaranteed momentum conservation down to the discretization error.

TURBULENT FLOW, V2-F INTERFACE

The $\overline{v^2}$-f turbulence model is an extension of the k-ε model with the turbulent viscosity based on the cross-stream velocity fluctuations. This makes it possible to model turbulence anisotropy and to distinguish between wall blockage and low-Reynolds-number effects.

ENHANCED WALL TREATMENT

The Single-Phase Flow interfaces have a new Wall feature, and for turbulent flows there is a new Wall Treatment property. The new Wall feature has a simplified user interface and automatically accounts for moving meshes.

The Wall Treatment Property introduces an Automatic option for all turbulence models except the k-ε model. Automatic wall treatment gives an accurate low-Reynolds-number formulation in regions where the mesh is fine enough and falls back on a wall-function formulation in regions with a coarser mesh.

Models created in previous versions of COMSOL Multiphysics open without the new Wall Treatment Property and with a legacy Wall Feature. To guarantee backward compatibility for Model Java® files using a Single-Phase Flow, Brinkman, or High Mach Number Flow interface, the following line should be added following the line creating the interface,

```
model.physics("phys").prop("PhysicalModelProperty").
     set("BackCompState", 0);
```

where `"phys"` is the interface tag (fore example, `spf`).

AUTOMATIC TRANSLATION BETWEEN TURBULENCE MODELS

The global feature Generate New Turbulence Model Interface creates a new Turbulent Flow interface (including a new study) from an existing one. Initial values and boundary conditions for turbulence variables not present in the original model are estimated using the turbulent viscosity and the turbulent length scale.

ALGEBRAIC MULTIGRID SOLVER FOR FLUID FLOW

The new default iterative solver for Fluid Flow applies algebraic multigrid (AMG) instead of geometric multigrid (GMG). The COMSOL implementation is based on the smooth aggregated AMG (SAAMG) method. Only one mesh is needed for AMG, whereas GMG requires multiple meshes. The performance for AMG is similar to that of GMG, but GMG is in general a bit faster, provided that enough mesh levels can be constructed. To facilitate a change between the two types of preconditioning, the default solver sequence includes a separate disabled iterative solver applying GMG.

INTERIOR WALL BOUNDARY CONDITION IN POROUS DOMAINS

The Interior Wall boundary condition is now available in the Darcy's law and Two-Phase Darcy's Law interfaces and is applicable in porous domains in the Single-Phase Flow and Brinkman interfaces.

THIN BARRIER BOUNDARY CONDITION IN THE DARCY'S LAW INTERFACE

The Thin Barrier boundary condition defines a permeable wall on an interior boundary.

New Models in Version 5.3

NEW TUTORIAL MODELS IN HIGH MACH NUMBER FLOW

- Euler Bump 3D
- Expansion Fan
- Supersonic Ejector

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`.

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)

Nonisothermal 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 5.3

NEW REACTING FLOW IN POROUS MEDIA MULTIPHYSICS INTERFACES

New Reacting Flow in Porous Media multiphysics interfaces, replacing the previous versions, are available in the Model Wizard. Selecting a Reacting Flow in Porous Media interface, under the Chemical Species Transport branch, the Brinkman Equations interface together with either the Transport of Diluted Species interface, or the Transport of Concentrated Species interface, are added to the Model Builder. In addition, a multiphysics coupling feature is added under the Multiphysics node. The coupling feature predefines and controls the couplings between the separate interfaces. A Flow Coupling feature is used with the Transport of Diluted, while a Reacting Flow feature is used with the Transport of Concentrated Species interface. Opening an application from an earlier version, the previous Reacting Flow in Porous Media interfaces are used.

NEW TRANSPORT IN DILUTED SPECIES IN FRACTURES INTERFACE

The new Transport in Diluted Species in Fractures interface is used to model mass transport along thin fractures occurring in solid or porous media. The interface is defined and solved on boundaries in the model geometry, assuming that the fracture thickness is very small compared to its other dimensions.

ADDITIONAL NEW FUNCTIONALITY

- New **Fracture** boundary feature in the Transport of Diluted Species interface when mass transfer in porous media is active. The **Fracture** feature is used to model mass transport along thin fractures in solid or porous media. It contains the same functionality as the Transport in Diluted Species in Fractures interface. By using this feature, the fast transport along thin fractures can be studied at the same time as the slower transport in adjacent solid or porous domains.
- The Creeping Flow interface is now available with a license for the Chemical Reaction Engineering Module. Correspondingly, the **Neglect inertial term (Stokes flow)** option is available in the Single-Phase Flow interface.

- The Nernst-Planck-Poisson Equations interface is now available with the Chemical Reaction Engineering Module.
- A New Electrophoretic Transport interface, and a new *Zone Electrophoresis* tutorial using that new physics interface.
- Migration in electric field is now supported by the **Porous Media Transport Properties** feature in the Transport of Diluted Species and Transport of Diluted Species in Porous Media interfaces.
- New mass transfer in porous media interface setting. The following interfaces now have a **Mass transfer in porous media** check box available in the **Transport Mechanisms** section (shown when selecting the interface):
  - Transport of Diluted Species
  - Transport of Diluted Species in Porous Media
  - Transport of Concentrated Species

  The **Mass transport in porous media** check box activates functionality specific to species transport in porous media. Selecting it, the **Dispersion in porous media** and **Volatilization in partially saturated porous media** check boxes are enabled. The following features are also enabled when selecting the Mass transport in porous media check box: **Adsorption**, **Fracture**, **Partially Saturated Porous Media**, **Porous Electrode Coupling**, **Porous Media Transport Properties**, and **Reactive Pellet Bed**.
- New **Adsorption** subfeature. The adsorption functionality available in the Transport of Diluted Species and Transport of Diluted Species in Porous Media interfaces has been moved to a new **Adsorption** subnode. This subnode can be added to the Porous **Media Transport Properties**, **Partially Saturated Porous Media**, and **Fracture** features. The functionality was previously available directly in the **Porous Media Transport Properties** and the **Partially Saturated Porous Media** features.

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 5.3

The following new functionality is available:

- A new Electrophoretic Transport interface.
- A new Current Distribution, Boundary Elements Interface. This interface replaces the Current Distribution on Edges, BEM interface of prior versions. The *Corrosion Protection of Multiple Oil Platforms* tutorial has been updated to use the new interface.
- A new Current Distribution, Shell Interface. The *Atmospheric Corrosion* tutorial has been updated to use the new interface.
- New Transport of Diluted Species in Fractures interface. See the release notes for the Chemical Reaction Engineering Module for more information.
- New and updated default plots in the Electrochemistry interfaces:
  - New Electrode Potential vs. Ground plots
  - New Electrode Potential vs. Adjacent Reference plot
  - Added arrows to Electrolyte Potential plots
  - New Boundary Potential vs. Ground plot (for time-dependent studies)
  - Added Real and Imaginary Impedance vs. Frequency (for frequency-domain studies)
- New **Ion-Exchange Membrane** node in the Tertiary Current Distribution, Nernst-Planck Interface.
- New **Charge Conservation Model** options in the Tertiary Current Distribution, Nernst-Planck Interface.
- New **Thin Electrode Layer** node.
- New **Thin Electrolyte Layer** node. This node replaces and extends the Thin Insulating Layer node of previous versions.
- In **Porous Electrode** nodes: Changed the default to **Bruggeman** transport correction in the Tertiary Current Distribution, Nernst-Planck interface, Lithium-Ion Battery, and Battery With Binary Electrolyte interfaces.
- New **Fracture** node in the Transport of Diluted Species in Porous Media Interface.

- New tutorial models:
  - The *Atmospheric Corrosion of a Busbar* model demonstrates a simulation of atmospheric galvanic corrosion of a busbar.
  - The *Estimation of Corrosion Kinetics Parameters* tutorial shows how to use the Optimization interface to perform electrode kinetics parameter estimations based on polarization data. This model requires the Optimization Module.

For users with a license for the AC/DC Module:

- New **Circuit Terminal** boundary node, applicable to external boundaries to electrode and porous electrode domains.
- New **Circuit terminal** boundary condition in the Electrode Surface node.
- New **Circuit terminal** condition operation mode in the Single Particle Battery interface.

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

New Functionality in Version 5.3

The new tutorial *Importing And Meshing of a PCB Geometry from an ODB++ Archive* has been added to demonstrate how to import data from an ODB++ archive to generate a geometry of a printed circuit board (PCB). The instructions detail how to find and remove small details from the geometry, how to create a mesh, and how to use automatically generated selections to define physics and mesh settings.

Backward Compatibility with Version 5.2a

- The default value for the `sellayer` property of the `Import` function is now `on`. Previously, the default was `off`.
- The default value for the `sellayershow` property of the `Import` function is now `all`. Previously, the default was `dom`.

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 5.3

The following new functionality is available:

- A New Electrophoretic Transport interface, and a new *Zone Electrophoresis* tutorial using that new physics interface.
- New and updated default plots in the Electrochemistry interfaces:
  - New Electrode Potential vs. Ground plots
  - New Electrode Potential vs. Adjacent Reference plot
  - Added arrows to Electrolyte Potential plots
  - New Boundary Potential vs. Ground plot (for time-dependent studies)
  - Added Real and Imaginary Impedance vs. Frequency (for frequency-domain studies)
- New **Ion-Exchange Membrane** node in the Tertiary Current Distribution, Nernst-Planck Interface. The updated *Electrodialysis* tutorial uses the new node.
- A new *Isoelectric Separation* tutorial model, using the Electrophoretic Transport and Laminar Flow interfaces to model isoelectric separation in a free flow electrophoresis device.
- New **Charge Conservation Model** options in the Tertiary Current Distribution, Nernst-Planck Interface.
- New **Thin Electrode Layer** node. This node replaces and extends the Thin Insulating Layer node of previous versions.
- In **Porous Electrode** nodes: Changed the default to **Bruggeman** transport correction in the Tertiary Current Distribution, Nernst-Planck interface.

For users with a license for the AC/DC Module:

- New **Circuit Terminal** boundary node, applicable to external boundaries to electrode and porous electrode domains.
- New **Circuit terminal** boundary condition in the Electrode Surface node.
- New **Circuit terminal** condition operation mode in the Single Particle Battery interface.

SURFACE REACTIONS INTERFACE

See Chemical Reaction Engineering Module.

Electrodeposition Module

New Functionality in Version 5.3

The following new functionality is available:

- A new Electrophoretic Transport interface.
- A new Current Distribution, Boundary Elements Interface. This interface replaces the Current Distribution on Edges, BEM interface of prior versions.
- New and updated default plots in the Electrochemistry interfaces:
  - New Electrode Potential vs. Ground plots
  - New Electrode Potential vs. Adjacent Reference plot
  - Added arrows to Electrolyte Potential plots
  - New Boundary Potential vs. Ground plot (for time-dependent studies)
  - Added Real and Imaginary Impedance vs. Frequency (for frequency-domain studies)
- New **Ion-Exchange Membrane** node in the Tertiary Current Distribution, Nernst-Planck Interface.
- New **Charge Conservation Model** options in the Tertiary Current Distribution, Nernst-Planck Interface.
- New **Thin Electrode Layer** node. This node replaces and extends the Thin Insulating Layer node of previous versions.
- In **Porous Electrode** nodes: Changed the default to **Bruggeman** transport correction in the Tertiary Current Distribution, Nernst-Planck interface.

For users with licenses for both the Electrodeposition Module and the Batteries & Fuel Cells Module: The **Nondeforming Boundary** and **Deforming Electrode Surface** multiphysics features now support couplings to the **Electrode Surface** node in the Lithium-Ion Battery and Battery with Binary Electrolyte interfaces.

For users with a license for the AC/DC Module:

- New **Circuit Terminal** boundary node, applicable to external boundaries to electrode and porous electrode domains.
- New **Circuit terminal** boundary condition in the Electrode Surface node.
- New **Circuit terminal** condition operation mode in the Single Particle Battery interface.

SURFACE REACTIONS INTERFACE

See Chemical Reaction Engineering Module.

Fatigue Module

New Functionality in Version 5.3

VIBRATION FATIGUE

Vibration Fatigue is a new type of fatigue damage evaluation in the Fatigue interface based on a frequency domain analysis containing results for a number of frequencies. It can for example be used for predicting the results from the common type of experiment where a structure is subjected to an excitation with increasing frequency, sometimes called a sine sweep. The following types of descriptions are available:

- A certain time spent at each frequency
- A certain number of cycles spent at each frequency
- Linearly increasing frequency
- Logarithmically increasing frequency

VISCOPLASTIC STRAINS

In the fatigue criteria based on inelastic strains, it is now possible to select also viscoplastic strains. This is an effect of the fact that a clear distinction between creep strains and viscoplastic strains has been introduced in the Nonlinear Structural Materials Module.

The same is true for energy based criteria, where there is now a distinction between energy dissipated by creep and by viscoplasticity.

This will affect existing applications with fatigue evaluation based on the Anand viscoplastic material model. Previously the strains from this material model were referred to as creep strains. The conversion is automatic when opening an .mph file from a previous version.

When using the API, however, you will need to replace expressions like

```
model.physics("ftg").feature("elif1").set("strainTypeCM", "Creep")
```

by

```
model.physics("ftg").feature("elif1").set("strainTypeCM",
"Viscoplasticity")
```

This also applies to energy based criteria, where for example

```
model.physics("ftg").feature("ener1").set("ftgEnerType", "Creep")
```

should be replaced by

```
model.physics("ftg").feature("ener1").set("ftgEnerType",
"Viscoplasticity")
```

Geomechanics Module

New Functionality in Version 5.3

HARDENING OF ELLIPTIC CAP

The Mohr-Coulomb and Drucker-Prager models for soil plasticity have been extended by a nonlinear hardening model for the optional elliptic cap of the yield surface.

EXTERNAL STRAIN FOR NONLINEAR ELASTIC MATERIAL

You can now add an **External Strain** node to the **Nonlinear Elastic Material**. This makes it possible to add inelastic strain contributions to the material in several ways:

- Explicit entry of strain tensor, deformation gradient or its inverse, or stretches.
- Incorporation of strain tensor, deformation gradient, or its inverse, announced from other physics interfaces.
- Compute inelastic strain contribution using a user-coded external function.

ASSOCIATED FLOW RULE FOR TRESCA YIELD FUNCTION

An associated flow rule has been added to the Tresca yield function in **Plasticity**. As before, the default flow rule is based on using the von Mises yield surface as plastic potential.

SCOPE CHANGE OF PLASTICITY VARIABLES

As an effect of the addition of new plastic and viscoplastic material models, several variables in the existing plasticity models have been moved from having “physics scope” to having “feature scope” in order to resolve possible conflicts. As an example the variable previously called `solid.Fyield` will now have a scope similar to `solid.lemm1.soil1.Fyield`.

In most cases, the old physics scope variables can still be accessed, so you should not experience any backward compatibility problems, even if you have used such variables in your own expressions.

Heat Transfer Module

New and Improved Functionality in Version 5.3

HEAT AND MOISTURE TRANSPORT

A set of new interfaces and features is available for modeling coupled heat and moisture transport in air and building materials. They complete the features already available in Version 5.2a for the modeling of heat and moisture transport in building materials, and allow you to create models where the air and building material domains are automatically coupled.

Moisture transport by convection and diffusion in air is handled by the new Moisture Transport in Air interface, which adds the **Moist Air** node as default model. You can account for the turbulent mixing caused by eddy diffusivity in turbulent moisture convection by adding the **Turbulent Mixing** subfeature under the **Moist Air** node. The **Moist Surface** and **Wet Surface** boundary conditions provide modeling of evaporation and condensation on boundaries. Finally, this new version of the Moisture Transport interface benefits from all the functionalities already available for moisture transport in building materials in Version 5.2a (**Building Material**, **Moisture Source**, **Moisture Content**, and so on).

The new **Moist Air** version of the Heat and Moisture Transport interface is found (along with the **Building Materials** version) under the new **Heat and Moisture Transport** group in the **Heat Transfer** branch in the model wizard. It adds the Heat Transfer in Moist Air interface, the Moisture Transport in Air interface, and the **Heat and Moisture** multiphysics node.

The Heat Transfer in Moist Air interface is a new version of the Heat Transfer interface, only available with the **Moist Air** version of the Heat and Moisture Transport interface, and having the new **Moist Air** domain feature as default model.

The new **Moist Air** domain feature of Heat Transfer interface corresponds to a preset version of the **Fluid** feature, with **Fluid type** set to **Moist air**. With the predefined multiphysics interface, the default **Input quantity** is set to **Relative humidity**, and the required inputs are taken from the multiphysics coupling node.

Finally, the following improvements have been added to the Moisture Transport interfaces:

- New default solver settings have been defined to improve convergence in study steps solving for the Moisture Transport interface: the default BDF order is now 2 in **Time Stepping** section; the **Number of iterations** in the **Method and Termination** section has been increased to 2 for the segregated step containing the relative humidity; and the default tolerance has been set to 1e-4. These last two new settings improve the quality of the solution.
- The **Thin Moisture Barrier** feature has been modified to be in better accordance with the material properties available for thin building materials. The default option models a vapor diffusion barrier, for which three alternative sets of material properties can be specified, depending on availability. The second option still allows to set a user-defined value for the **Moisture Transfer Coefficient**.

HEAT TRANSFER IN THE FREQUENCY DOMAIN

The Heat Transfer interfaces now support the **Frequency-Domain Perturbation** solver, which computes the harmonic temperature variations around an equilibrium state. The predefined **Thermal perturbation, Frequency domain** solver sequence first solves a **Stationary** study step to find the equilibrium state and then reuses that solution as a starting point for the second **Frequency-Domain Perturbation** step.

A new **Harmonic perturbation** subfeature can be added under the temperature boundary conditions (**Temperature**, **External Temperature**). Harmonic perturbation can also be applied on heat source features (**Heat Source**, **Boundary Heat Source**, **Layer Heat Source**, **Line Heat Source**, **Line Heat Source on Axis**, **Point Heat Source**).

The **Frequency to Time FFT** and **Time to Frequency FFT** study steps have been made available to perform inverse and forward FFT between time and frequency domains.

NEW GEOMETRY PARTS FOR HEAT SINKS

New geometry parts have been added for the definition of heat sinks. They contain different parameterized geometries for heat sinks with pin fins, straight fins, or pin fins with dissimilar dimensions on the borders, that make it possible to easily include a heat sink in any 3D model.

NEW BUILDING AND REFRIGERANT MATERIALS

A new Building material database is available. It provides hygroscopic and thermal properties of 12 materials commonly used in buildings: cellular concrete, wood (pine), gypsum board, glass wool board, cellulose board, glass wool batt, cellulose batt, cellulose (loose), lime silica brick, plastic coated paper, expanded polystyrene board (EPS), and extruded polystyrene board (XPS).

In particular, the hygroscopic properties (moisture diffusivity, water content, vapor resistance factor, and vapor permeability) needed by the domain and boundary features of the Moisture Interface are given.

Two new refrigerant materials have been added in the Liquid and Gases material database: R-134A (C2H2F4) and R-22 (CHClF2). Their thermal properties are given.

IRREVERSIBLE TRANSFORMATION IN SOLIDS

The **Irreversible Transformation** attribute node is now available under the **Solid** domain node. It should be used to model thermally-induced irreversible transformations in solids. The **Temperature threshold** model evaluates the transformation from the time spent over (for **Overheating analysis**) or below (for **Overcooling analysis**) a given temperature. The **Energy absorption** model is based on Arrhenius equation.

The heat source or sink due to enthalpy change during the transformation is accounted for in the energy balance. Finally, you can define different thermal properties for the transformed state; in this case, the material thermal properties are dependent of the material damage indicator.

When this attribute is active, the segregated solver attribute is set by default, and the group containing the irreversible transformation indicator is solved by a **Jacobi** iterative method.

SERENDIPITY SHAPE FUNCTIONS FOR TEMPERATURE

The temperature dependent variable in Heat Transfer and Heat Transfer in Thin Structures interfaces, which was always of Lagrange type, can now be of serendipity type. This is reflected in the following new choices in the **Discretization** section of the physics interface node: **Quadratic serendipity**, **Cubic serendipity**, and **Quartic serendipity**.

For highly distorted elements, Lagrange shape functions provide better accuracy than serendipity shape functions of the same order. The serendipity shape functions will, however, give significant reductions of the model size for a given mesh containing hexahedral, prism, or quadrilateral elements. At first order, Lagrange and serendipity shape functions are the same.

The shape functions for `TextFace` and `Txdim` in the **Thin Layer** feature, and `TWall_u` and `TWall_d` in the **Nonisothermal Flow** multiphysics coupling feature also follow the temperature shape function type, in particular if a serendipity option is chosen.

SYMMETRY FOR SURFACE-TO-SURFACE RADIATION EXTENDED CAPABILITIES

The **Symmetry for Surface-to-Surface Radiation** global feature has new options available to take advantage of multiple plane symmetries for 2D and 3D models. You should use these options to reduce the mesh size, for a more efficient computation of the view factor. In 2D models, the **Two perpendicular planes of symmetry** option is available. In 3D, either the **Two perpendicular planes of symmetry** option (planes with intersection parallel to one of the axis), or the **Two perpendicular planes of symmetry** option (planes parallel to the axis), can be defined.

In addition, you can visualize the plane(s) of symmetry in the **Graphics** window along with the geometry. This is meant to ease the user experience when defining the plane(s), and it applies to the options **Plane of symmetry** (2D, 2D-axi, and 3D models), **Two perpendicular planes of symmetry** (2D and 3D models), and **Three perpendicular planes of symmetry** (3D models). A check box is available to show/hide the planes in the **Graphics** window.

IMPROVED FUNCTIONALITY FOR CONVECTIVE HEAT FLUX MODELING

In the **Heat Flux** boundary condition, the **Convective heat flux** option has now two new heat transfer coefficient correlations for **External forced convection**: **Cylinder in cross flow** and **Sphere**. Such a heat transfer coefficient can be used to reduce the simulation cost when the model configuration corresponds to one of the predefined correlations. In that case, the flow computation and the heat convection in the fluid are replaced by a heat flux boundary condition on the solid boundaries using the corresponding predefined correlation. For the **Cylinder in cross flow** correlation, the cylinder axis is supposed to be perpendicular to the flow direction.

In addition, sketches and more detailed equations have been added in the user interface for all the correlations used for the **Convective heat flux** option. The images appear in a new **Sketch** section, collapsed by default, showing the geometry along with the input parameters used in each correlation.

DIFFUSE AND DIRECT SOLAR RADIATION IN SURFACE-TO-SURFACE RADIATION MODELS

A new option is available in the **Diffuse Surface** and **Diffuse Mirror** features to account for diffuse solar irradiation in surface-to-surface radiation modeling. This contribution corresponds to solar radiation scattered by the atmosphere. It is added to the direct solar irradiation (sun rays directly striking the surfaces) defined by the **External Radiation Source** feature. The **Clear sky noon diffuse horizontal irradiance**, defined in the **Ambient Settings** section of the physics interface, can be used as input.

AUTOMATIC TEMPERATURE DIFFERENCE WHEN EVALUATING THE EQUIVALENT CONDUCTIVITY FOR CONVECTION

In the **Fluid** feature of the Heat Transfer interface, the **Equivalent Conductivity for Convection** section provides a new **Automatic** option for evaluating the **Temperature difference** in the Rayleigh number, when choosing correlations for **Horizontal cavity heated from below** and **Vertical rectangular cavity**. The temperature difference is then determined by finding the difference between the maximum and minimum temperature of the domain boundaries.

TIME-DEPENDENT ENERGY STORAGE TERM IN THERMALLY THICK LAYER

The **Thermally thick approximation** option in the **Thin Layer** feature now accounts for a time-dependent term modeling the energy storage over time. This term requires the material density and the heat capacity at constant pressure as inputs in the new **Thermodynamics** section in the feature user interface. For time-dependent models modeling a thermally thick layer by a boundary, such an energy storage term improves the accuracy of the temperature profile compared to modeling the thin layer by a full geometrical domain.

THIN FILM AND THIN LAYERED SHELL EXTENDED CAPABILITIES

In the Heat Transfer and Heat Transfer in Thin Shells interfaces, a new subfeature **External Temperature** is available under **Thin Film** node, when **Thin film model** is set to **General**. When used in Heat Transfer in Thin Shells interface, you can specify the **Downside** and **Upside** temperatures.

In Heat Transfer in Thin Shells interfaces, the **Thin Layered Shell** node has two new subfeatures: **External Temperature**, to define the temperature on the downside and/or upside of a thin layered shell, and **Heat Source**, to add an internal heat source within each layer of a thin layered shell.

IMPROVED AMBIENT DATA FUNCTIONALITY

In the **External Radiation Source** global feature, the user interface (**Location**, **Date**, and **Local time**) has been synchronized with the **Ambient Settings** functionality, when **Ambient data** is set to **Meteorological data (ASHRAE 2013)**.

Additionally, the ambient temperature has been added as a feature input in the **Isothermal Domain** feature, when **Temperature definition** is set to **From prescribed temperature**.

Finally, all the ambient data variables (`ht.Is_amb`, `ht.Ish_amb`, `ht.Isn_amb`, `ht.T_amb`, `ht.p_amb`, `ht.phi_amb`, `ht.v_amb`, and `ht.xvap_amb`) are now available as global variables and can be used in a **Global Evaluation** for postprocessing.

IMPROVED BIOLOGICAL TISSUE FUNCTIONALITY WITH DAMAGE INTEGRAL ANALYSIS

In the **Damaged Tissue** section of the **Biological Tissue** node, it is now possible to account for the enthalpy change due to the damage. Depending on damage integral model, new fields for enthalpy change are added in the user interface, and the new variable `ht.Qdmg` contains the damage heat source.

When the **Biological Tissue** node is active with the **Include damage integral analysis** check box selected, the damage indicator is solved by an iterative **Jacobi** method by default.

NEW ONE-WAY COUPLED STUDIES FOR NONISOTHERMAL FLOW MODELING

Two new predefined studies are available with all the versions of the **Nonisothermal Flow** and **Conjugate Heat Transfer** interfaces: **Stationary, One-Way Coupled, NITF** and **Time Dependent, One-Way Coupled, NITF**.

These studies contain two steps (**Stationary** or **Time Dependent**), one solving for the fluid flow interface, and one solving for the heat transfer interface.

While multiphysics stabilizations (streamline and crosswind diffusion) are available for the fully-coupled studies, the stabilization of each solved physics is used instead when one of the physics interfaces or the multiphysics coupling is not solved in a study step, and the solver suggestions are uncoupled.

FEATURES AND OPTIONS NAME CHANGE

- Moisture Transport interface is renamed Moisture Transport in Building Materials.
- **Porous Medium** node in Moisture Transport interface is renamed **Building material**.
- **Non-Isothermal Flow** interface and feature are renamed **Nonisothermal Flow**.
- **Inward heat flux** option is renamed **Convective heat flux** in **Line Heat Flux** feature.

MISCELLANEOUS

- New predefined variables are available with the Heat Transfer in Thin Shells interface for the control of fluxes on boundaries where this physics interface is active. You can plot the (vectorial) variables `htsh.tflux`, `htsh.dflux`, `htsh.cflux`, and `htsh.teflux` to control the total heat flux, conductive heat flux, convective heat flux, and total energy flux on the boundaries. In addition, the (scalar) variables htsh.ttflux, htsh.tdflux, htsh.tcflux, and htsh.tteflux contain the corresponding in-plane fluxes.

- A new predefined variable ht.c_ref has been added when the **Fluid** feature is active with **Fluid type** set to **Moist air** and **Input quantity** set to **Relative humidity**, to control the concentration corresponding to the other reference variables set in the user interface (temperature, pressure, and relative humidity).
- For turbulent flows, a clarification has been done for boundary temperature definitions. Side temperatures (Tu and Td), wall temperatures (TWall_u and TWall_d), and temperatures of the fluid in the turbulent boundary layer (TuWF and TdWF) are defined.
- When the **Local Thermal Non-Equilibrium** coupling node is active, a new plot showing the average temperature ltne1.T is available in the default plots.
- The **Mass flow** option is now available in the **Inlet** feature with the Heat Transfer Module.
- A new option has been added in **Isothermal Domain Interface** feature to control the **Interface Opacity** when **Surface-to-surface radiation** is enabled. In addition, the variables Tu, Td, and TextFace have been added on boundaries, along with variables for heat balance control.
- When the **Equivalent Conductivity for Convection** functionality is used in the **Fluid** feature, the correlation used for the Nusselt number definition has been added to the **Equation** section for clarity.
- The handling of different discretization orders for the temperature and the damaged tissue indicator variables when **Biological Tissue** feature is active has been improved when a multigrid solver is used.
- The default **Discretization** for temperature has been set to **Linear** in the Heat Transfer in Fractures interface.
- The stabilization has been improved for the features **Thin Film** (with **Thin film model** set as **General**) and **Thin Layered Shell**. In addition, heat balance has been improved in **Thin Film** feature.
- In the **External Radiation Source** feature, when the **Wavelength dependence of emissivity** is **Solar and ambient** or **Multiple spectral bands**, and **Source position** is **Infinite distance**, the **Source heat flux** feature input has been added in the user interface.
- The definition of the layer thickness variable has been fixed in **Thin Layer** node when **Layer type** is set to **Thermally thick approximation** or **General**.
- The assembly block size has been changed to 100 when the **Diffuse Surface** feature is active, with the **Include surface-to-surface radiation** check box selected.

New Application in Version 5.3

ELECTRONIC CHIP COOLING

A new tutorial, Electronic Chip Cooling, uses a heat sink geometry that is defined from the new Heat Transfer Module section part library. This tutorial model shows different heat transfer modeling approaches to study the cooling of an electronic chip. In the first part, only the solid parts are modeled, and the air flow cooling is modeled using convective heat flux boundary conditions. Then, the model is extended to include a fluid domain for the flow channel to compute the temperature and the velocity of the fluid. In the last part, surface-to-surface radiation is considered to compare the results with and without surface-to-surface radiation.

Updated Applications in Version 5.3

EVAPORATIVE COOLING OF WATER

The Evaporative Cooling of Water application has been updated to take advantage of the new **Moist Air** version of the Heat and Moisture Transport interface. In this tutorial, a beaker filled with water is surrounded by an air domain. The airflow transports the water vapor which causes the liquid to cool down. The computation of the amount of water that is evaporated from the beaker into the air uses in particular the **Moist Air** domain node of the Moisture Transport interface with a **Turbulent Mixing** subnode and the **Wet Surface** boundary node.

In addition, the size of the air domain has been increased such that further increasing the domain would have no noticeable effect on the flow field around the beaker.

GEOMETRY UPDATE IN APPLICATIONS

The following applications have been updated to use geometry sequences: Free Convection in a Light Bulb, Condensation Detection in an Electronic Device, Shell-and -Tube Heat Exchanger, Power Transistor, Heat Sink, Microwave Heating of a Cancer Tumor, Heat Transfer in a Surface-Mount Package for a Silicon Chip, and Glazing Influence on Thermal Performances of a Window.

In all these applications, the **Insert Sequence** functionality under **Geometry** node has been used to import the geometry defined in a separated MPH-file. With this process, the geometry stay editable when imported in the application.

MISCELLANEOUS

- The **Thin Layer** feature has been removed from the Heat Sink tutorial as it did not have any thermally relevant effect.
- The Marangoni Effect tutorial now uses an auxiliary sweep for better performance.
- The Thermal Annealing tutorial now uses a symmetry plane for surface-to-surface radiation.
- Several applications have been updated to use the new algebraic multigrid solver set as the default solver for fluid flow.

Backward Compatibility with Version 5.2a

The obsolete features **Convective Heat Flux** and **Out-of-plane Convective Heat Flux** are removed in Version 5.3. These were marked as obsolete with a warning message since Version 5.0. These features will be removed from your model when opening in Version 5.3. Use instead the corresponding **Heat flux** and **Out-of-Plane Heat Flux** features, with the option **Convective heat flux**. The variables `ht.chflux`, `ht.chflux_u`, `ht.chflux_d`, and `ht.chflux_z` defined by these features have been also removed from equation view and post-processing variables menu.

The new default option in **Thin Moisture Barrier** feature is **Vapor barrier material properties**, and the **Moisture barrier material properties** option has been removed. When loading a MPH file that was created before version 5.3 this latter option will be cleared, and the corresponding user inputs used to define the **Moisture transfer coefficient** directly. However, for the API, it will be necessary to update the code with an existing option name.

In the **Equivalent Conductivity for Convection** section of the **Fluid** node, the new **Automatic** option for **Temperature difference** has become the default option. When loading a MPH file that was created before version 5.3, the **User defined** option will be set automatically with the previous value. However, for the API, it will be necessary to update the code to set the **User Defined** option.

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 **Nonisothermal 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
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/m² (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
- Nonisothermal 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:

- **Convective Heat Flux** (replaced by **Heat Flux** with **Convective heat flux option**)
- **Out-of-plane Convective Heat Flux** (replaced by **Out-of-plane Heat Flux** with **Convective heat flux option**)
- **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 5.3, 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 5.3

The following new functionality is available:

- Spring boundary conditions for the Rigid Domain and Rigid Connector features.
- Energy quantity variables have been added for contact modeling.
- Frequency domain analysis with contact modeling is now supported
- Harmonic Perturbation is now available for the Prescribed Velocity and Acceleration features.
- Enhancements have been added for including external material functionalities.

See also the Structural Mechanics Module for improvements for the structural mechanics interfaces 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

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.

Molecular Flow Module

New Functionality in Version 5.3

A Plane Symmetry boundary condition is now available for models with geometric symmetry.

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 5.3

HIGHLIGHTING OF SELECTIONS

When working with complex assemblies in the Multibody Dynamics interface, there may be many joints, each containing the selection of two rigid domains or attachments. When you select a **Joint** node in the model tree, the selected objects are now automatically highlighted in the graphics window as a quick feedback of the correctness of the source and destination selections. Similarly, the two gears in a **Gear Pair**, and the two parts joined by a **Spring-Damper** will be highlighted when the corresponding nodes are selected.

PENALTY METHOD FOR COMPUTING JOINT FORCES

In pure rigid body systems, it is not unusual that model of the system is over-constrained. To handle this situation, a penalty method for enforcing the joint constraints has been added. This makes it possible for you to determine the joint forces in such a case, but the method is generally applicable. You select the method to be used in the **Joint Forces and Moments** section in the settings for a certain joint. For this situation, a penalty method for enforcing the joint constraints has been added. This makes it possible for you to determine the joint forces in such a case, but the method is generally applicable. Related to this, the default is now not to compute joint forces between rigid bodies in order to reduce the risk of numerical problems caused by overconstraints.

MODAL REDUCED ORDER MODEL STUDY TYPE

The Modal Reduced Order Model study type is now supported also by the Multibody Dynamics interface. This study type is useful for extracting system matrices represented in a modal base for large MBD models.

ATTACHMENTS ON RIGID BODIES

It is now possible to use an **Attachment** also on a rigid domain. The purpose is to make it easier to switch between a rigid representation and an elastic representation of a certain domain, since its selection in a joint node does not have to be changed.

SERENDIPITY SHAPE FUNCTIONS

When modeling flexible parts in the Multibody Dynamics interface, you can choose to discretize by serendipity shape functions. This will reduce the number of degrees of freedom when using structured meshes with higher order elements.

OFFSET OPTION FOR POSITIONS IN SEVERAL FEATURES

When specifying positions, such as center of rotation, or location of forces, in the **Rigid Domain, Rigid Connector, Initially Rigid**, **Spring-Damper**, gear, and joint features, it is now possible to add an offset vector to the specified position.

New Applications in Version 5.3

VIBRATIONS AND NOISE IN A GEARBOX

This tutorial example illustrates the modeling of vibration and noise in a 5-speed synchromesh gearbox inside a manual transmission vehicle. A transient multibody dynamics analysis computes the gearbox vibrations for the specified engine speed and external load. The normal acceleration of the gearbox housing is converted to the frequency domain, to be included as a source of noise. An acoustics analysis is then performed in order to compute the sound pressure levels in the near, far, and exterior fields.

VIBRATION IN AN INDUCTION MOTOR

This tutorial example models an induction motor in which eddy currents are induced in the rotor by the time harmonic currents on the stator windings and the rotation of the rotor. The air gap between the rotor and stator is assumed asymmetric when analyzing the resulting vibrations in the motor.

The electromagnetic simulation is modeled in 2D, whereas the multibody dynamics simulation is modeled in 3D. The rotational torque, when an alternating current is passed through the stator windings, is calculated as a function of time. The torque is then used in the multibody dynamics model in order to compute the rotor speed, accounting for inertial effects.

Nonlinear Structural Materials Module

New Functionality in Version 5.3

POROUS PLASTICITY

A new family of material models for modeling porous plasticity has been added. The **Porous Plasticity** subnode is available for **Linear Elastic Material** and **Nonlinear Elastic Material**.

There are five new porous plasticity models:

- Shima-Oyane
- Gurson
- Gurson-Tvergaard-Needleman
- Fleck-Kuhn-McMeeking
- FKM-GTN

NEW VISCOPLASTIC MATERIAL MODELS

Two new material models have been added to **Viscoplasticity**:

- Chaboche
- Perzyna

EXTERNAL STRAIN FOR NONLINEAR ELASTIC MATERIAL

You can now add an **External Strain** node to the **Nonlinear Elastic Material**. This allows you to add inelastic strain contributions to the material in several ways:

- Explicit entry of strain tensor, deformation gradient or its inverse, or stretches.
- Incorporation of strain tensor, deformation gradient, or its inverse, announced from other physics interfaces.
- Compute inelastic strain contribution using a user-coded external function.

INELASTIC STRAINS FOR HYPERELASTIC MATERIAL

The possibility to describe inelastic strains when using **Hyperelastic Material** has been significantly extended.

- Thermal expansion can now be non-isotropic.
- Hygroscopic swelling can now be non-isotropic.
- A new **External Strain** subnode has been added, in which you can prescribed any set of inelastic strains, including strains computed from a user-coded external function.

MATERIAL PROPERTIES FOR ANAND MODEL

The material properties for the Anand viscoplastic model can now be specified under the **Material** node.

ASSOCIATED FLOW RULE FOR TRESCA YIELD FUNCTION

An associated flow rule has been added to the Tresca yield function in **Plasticity**. As before, the default flow rule is based on using the von Mises yield surface as plastic potential.

SCOPE CHANGE OF PLASTICITY VARIABLES

As an effect of the addition of new plastic and viscoplastic material models, several variables in the existing plasticity models have been moved from having “physics scope” to having “feature scope” in order to resolve possible conflicts. As an example the variable previously called `solid.sY` will now have a scope similar to `solid.lemm1.plsty1.sY`.

In most cases, the old physics scope variables can still be accessed, so you should not experience any backward compatibility problems, even if you have used such variables in your own expressions.

New Applications in Version 5.3

LEMAITRE-CHABOCHE VISCOPLASTIC MODEL

This tutorial model demonstrates the Lemaitre-Chaboche viscoplastic constitutive law, having a combination of isotropic hardening and a nonlinear kinematic hardening. This viscoplastic model is commonly used in areas such as additive manufacturing, laser welding, laser cutting, and thermal processing metals and alloys at high temperatures.

Optimization Module

New Functionality in Version 5.3

The Optimization Module has been improved with better stability, consistency, and error handling.

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 5.3

PERIODIC BOUNDARY CONDITION

The new **Periodic Condition** feature is used to model particle tracing in periodic structures or in geometries with sector symmetry. When a particle reaches a surface with the **Periodic Condition**, it immediately gets mapped to a destination point on another surface. As the particle exits the destination surface, its velocity can either be kept the same, rotated (for sector symmetry), or reset to a user-defined expression.

ROTATING FRAMES

Use the **Rotating Frame** feature to automatically apply the centrifugal, Coriolis, and Euler forces for particle tracing in a rotating frame of reference. When releasing particles, it is possible to specify the initial velocity in an inertial frame or with respect to the rotating frame.

RANDOM INITIAL POSITIONS

It is now possible to release particles at random initial positions on selected domains, boundaries and edges. Unique positions are chosen for each release time. This is available in the **Release**, **Inlet** and **Release from Edge** features.

COORDINATE SYSTEM SELECTION FOR INLETS

When releasing particles at a boundary using the **Inlet** feature, it is now possible to initialize the particle velocity or momentum using any coordinate system that has been defined for the model component.

LAMBERTIAN VELOCITY DISTRIBUTION

The particle release features can now release particles with a velocity distribution in 3D following Lambert's cosine law.

NONUNIFORM MAGNITUDES IN VELOCITY DISTRIBUTIONS

For the spherical, hemispherical, conical, and Lambertian velocity distributions, it is now possible to release particles with a distribution of speeds as well as directions.

By default, every particle that is released from the same point in a velocity distribution will have the same magnitude. However, by expressing the initial speed in terms of the unique particle index, it is now possible to apply a different initial speed to each particle without changing the distribution of particle directions. This makes it easier to include distributions of particle speed or energy as well as direction.

LIFT FORCE FOR THE PARTICLE TRACING FOR FLUID FLOW INTERFACE

A dedicated **Lift Force** feature is now available for the Particle Tracing for Fluid Flow interface.

Two different formulations of the lift force are available. The Saffman lift force is applicable to inertial particles in a shear flow that are an appreciable distance away from boundaries. A specialized wall induced correlation is also available for neutrally buoyant particles in channels.

ANISOTROPIC TURBULENT DISPERSION

When applying a random turbulent dispersion term to the drag force on particles in a fluid using the continuous random walk model, the turbulent dispersion can now be either isotropic (the default) or anisotropic. If anisotropic turbulence is used, the turbulent dispersion terms are computed using specific expressions for the streamwise, spanwise, and wall normal directions. Anisotropic turbulence can provide a more realistic depiction of particle motion in a turbulent flow when the particles are close to walls.

THERMIONIC EMISSION OF ELECTRONS

The dedicated **Thermionic Emission** feature is available to model the release of electrons from a hot cathode. This feature is available in the Charged Particle Tracing interface. The total current density released from the boundary is computed using Richardson's law, where the effective Richardson constant, work function of the metal, and temperature can be specified.

DRAG CORRECTION FACTOR FOR PARTICLES CLOSE TO WALLS

The **Drag Force** feature for the Particle Tracing for Fluid Flow interface now supports corrections to account for the presence of nearby walls.

SYMMETRY CONDITION FOR PARTICLE TRACING

The **Symmetry** condition is a specialized boundary condition for the Charged Particle Tracing and Particle Tracing for Fluid Flow interfaces. It is a special case of the **Wall** condition that always causes model particles to be specularly reflected at the boundary. The physical interpretation of this reflection is that, for every particle that would leave the modeling domain through a symmetry plane, an identical particle would simultaneously enter the modeling domain at the same location and the same time.

NEW OPTIONS FOR INLET PAIRS

When releasing particles from an **Inlet Pair** defined on an assembly, it is now possible to release the particles from only the source boundary, destination boundary, or both. This is most noticeable when using a mesh-based release of particles, since the mesh on either side of the identity pair can be different.

Backward Compatibility with Version 5.2a and Earlier

REFERENCE EDGE FEATURES HAVE BEEN REMOVED

The **Reference Edge** subnode, which was previously used to specify the orientation of the transverse beam directions for the **Particle Beam** feature in 3D, has been removed. Instead, the orientation of the transverse beam directions is specified using an extra selection field in the settings window for the **Particle Beam** feature itself. Similarly, the **Reference Point** subnode for specifying the beam center has been replaced by an extra selection in the settings window for the **Particle Beam** feature.

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.

Backward Compatibility with Version 5.0 and Earlier

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

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

ELASTIC COLLISION FORCE NODE HAS BEEN REPLACED

If a model created in version 5.0 or earlier containing the **Elastic Collision Force** node is opened in version 5.1 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 5.3

NEW OPTION FOR T-JUNCTION FEATURE FOR THE PIPE FLOW INTERFACE

An additional option for the T-Junction feature called *Loss coefficients, extended model* is available for the Pipe Flow interface. This option allows you to specify six dimensionless loss coefficients to account for joining and separating flows in junctions.

INTERNAL PRESSURE LOCK FEATURE

For the recirculating flows of incompressible fluids, the pressure value should be specified in at least one point somewhere in the pipe network. The new **Internal Pressure Lock** feature allows you to specify such pressure, and it can be used together with valves, bends, contraction/expansion, and lossless fittings.

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 5.3

The following new functionality is available:

- Global modeling from within the Plasma interface is now possible.
- The local field approximation can be used to compute transport and source coefficients instead of solving the electron energy equation.
- The electron mobility can now be automatically computed from the list of electron impact reactions.
- Units can be assigned to all of the tables in the Plasma interface.

New and Updated Applications and Models in Version 5.3

NEW APPLICATION

Boltzmann DC Glow Discharge: This application models a DC glow discharge that periodically solves the two-term Boltzmann equation to update the transport and source coefficients.

NEW MODELS

- Atmospheric Pressure Corona Discharge in Air
- Negative Streamer in Nitrogen in 1D
- Chlorine Discharge Global Model
- Microwave Microplasma

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 5.3

RAY TERMINATION FEATURE

The new **Ray Termination** feature can be used to annihilate rays without requiring them to stop at a boundary. The rays can be terminated as they exit a bounding box, which can be based on the geometry or user-defined spatial extents. Use the **Ray Termination** feature to easily discard unnecessary information about the ray paths and remove clutter from the trajectory plots.

IMPORT PHOTOMETRIC DATA FILES

It is possible to specify nonuniform distributions of ray intensity and power by importing photometric data files into a ray optics model. The **Photometric Data Import** feature supports the `*.ies` file extension, the standard photometric data file format of the Illuminating Engineering Society of North America (IESNA).

When the photometric data file is imported into a model, it generates a set of functions that are used to initialize the ray intensity and power as a function of the initial ray direction. You can specify directions for the photometric horizontal and photometric zero, which indicate the orientation of the lamp according to IES standards.

GEOMETRY PART VARIANTS

There are now several different ways to specify the dimensions of geometry parts in the dedicated Part Library for the Ray Optics Module. It is possible to choose from different combinations of input parameters when loading the part into a model.

NEW GEOMETRY PARTS: COMPOUND PARABOLIC CONCENTRATORS

The Part Library for the Ray Optics Module now includes the Compound Parabolic Concentrator (CPC). The CPC has parabolic surfaces that are positioned close enough together so that the end of each side is located at the focal point of the opposite side. Light that is incident at anything less than a specified angle, called the acceptance half-angle, will always be transmitted through the concentrator, making it a useful tool for focusing incoming radiation from several different directions.

LAMBERTIAN EMISSION

The ray release features can now release rays with a Lambertian distribution of initial directions. Rays can be released with initial directions based on Lambert's cosine law.

IMPROVED HANDLING OF RAY TRACING IN 2D AXISYMMETRIC GEOMETRIES

When computing ray intensity in 2D axisymmetric models, the wavefront associated with the propagating ray is now treated as a spherical or ellipsoidal wave instead of a cylindrical wave (which is only an appropriate simplification in true 2D models). In other words, the principal radius of curvature associated with the azimuthal direction is computed for all rays. This leads to more realistic calculations of ray intensity in 2D axisymmetric models.

In addition, dedicated release features are now available to release rays from edges, points, or at specified coordinates along the axis of symmetry. When using one of these dedicated release features, a built-in option is available to release rays in an anisotropic hemisphere such that each ray subtends approximately the same solid angle in 3D.

RAY DETECTOR FEATURE

A **Ray Detector** feature is a domain or boundary feature that provides information about rays arriving on a set of selected domains or surfaces from a release feature. Such quantities include the number of rays transmitted and the transmission probability, or the ratio of the number of transmitted rays to the number of released rays. It is possible to count all rays or only those rays released by a specific physics feature. The feature provides convenient expressions that can be used in the **Filter** node of the **Ray Trajectories** plot, which allows only the rays that reach a specified set of domains or boundaries to be visualized.

The following variables are defined by an instance of the **Ray Detector** feature with the feature tag `<tag>`:

- `<tag>.Nsel` is number of transmitted rays from the release feature to the detector.
- `<tag>.alpha` is the transmission probability from the release feature to the detector.
- `<tag>.rL` is a logical expression for ray inclusion. This can be set in the **Filter** node of the **Ray Trajectories** plot in order to visualize the rays that connect the radiation source to the detector.

COORDINATE SYSTEM SELECTION FOR INLETS

When releasing rays at a boundary using the **Inlet** feature, it is now possible to initialize the particle velocity or momentum using any coordinate system that has been defined for the model component.

ADVANCED OPTIONS TO SPECIFY RAY RELEASE TIMES

It is now possible to enter a range of different ray release times. In previous versions, all rays had to be released at the same time. To enable specification of different release times, select the **Allow multiple release times** check box in the physics interface **Advanced Settings** section.

Backward Compatibility with Version 5.2a and Earlier

REFERENCE EDGE FEATURES HAVE BEEN REMOVED

The **Reference Edge** subnode, which was previously used to specify the orientation of some optical components in 3D, has been removed. Instead, the orientation of these components is specified directly in the parent feature settings using an extra selection input. The following features are affected:

- **Linear Polarizer**
- **Linear Wave Retarder**
- **Mueller Matrix**
- **Grating**

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 5.3

RF MODULE PART LIBRARY

The RF Module part library includes the following parts:

- 36 rectangular waveguides (straight and 90-degree bend, H-bend)
- 22 surface mount device footprint
- 3 SMA connectors (4 holes, 2 holes, and vertical mount)

NEW AND ENHANCED FEATURES

A surface magnetic current density feature is now available.

Enhanced features:

- Lumped element feature with extended options: series LC, parallel LC, series RLC, and parallel RLC.
- Two-port network for advanced two-port device modeling with Touchstone file import.
- Transient lumped port supporting S-parameter calculation.
- Scattering boundary condition for Electromagnetic Waves, Transient can be used now also under high loss conditions.

NEW FAR-FIELD POSTPROCESSING VARIABLES

- `EIRP` and `EIRPdB`: effective isotoropic radiated power and its dB-scaled value.
- `gainEfar` and `gaindBEfar`: gain excluding input mismatch and its dB-scaled value.
- `rGainEfar` and `rGaindBEfar`: realized gain including input mismatch and its dB-scaled value

NEW SETTINGS FOR ENHANCED USABILITY

In the Electromagnetic Wave, Frequency Domain physics interface:

- Default enabled physics-controlled mesh
- The mesh picks up the frequency from study steps automatically
- Default fast analysis methodology for solver settings (FGMRES)

The default 3D far-field plot now uses a finer angular resolution (theta 45, phi 45).

Automatic excitation on for the first port.

The GHz frequency unit is now available for Frequency Domain, Frequency-Domain Modal, and Eigenfrequency studies. The GHz units is also now used unit in the default S-parameter plot with simpler S-parameter descriptions.

For frequency-domain modal analysis:

- Eigenfrequency search method around shift as Larger real part.
- The `linper` operator applied internally in an excited port or lumped port feature.

New and Updated Models and Applications in Version 5.3

NEW AND UPDATED MODELS

- Modeling of a Low-Pass and Band-Pass Filter Using Lumped Elements
- Anechoic Chamber Absorbing Electromagnetic Waves
- Double-Ridged Horn Antenna
- Fast Modeling of a Transmission Line Low-Pass Filter
- Fast Modeling of a Transmission Line Wilkinson Power Divider
- Fast Prototyping of a Butler Matrix Beamforming Network
- Time-to-Frequency Fast Fourier Transform of a Coaxial Low-Pass Filter
- Filter Characterized by Imported S-parameters via a Touchstone File
- High-Speed Interconnect Tuning by Time-Domain Reflectometry
- Defining a Mapped Dielectric Distribution of a Metamaterial Lens has been renamed as Defining a Mapped Dielectric Distribution of a Material

UPDATED TUTORIALS USING REDUCED ORDER MODELING TECHNIQUES

Using asymptotic waveform evaluation:

- Evanescent Mode Cylindrical Cavity Filter
- RF Coil

Using frequency-domain modal analysis:

- Cascaded Rectangular Cavity Filter
- Coupled Line Filter

- CPW Bandpass Filter
- Waveguide Iris Filter

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

Rotordynamics Module

New Functionality in Version 5.3

INLET AND OUTLET ON THE BEARING SURFACE

In the Hydrodynamic Bearing interface, **Inlet** and **Outlet** boundary conditions for surfaces have been added. This makes it possible to model situations where a lubricant is supplied inside the bearing.

MODIFIED REYNOLDS EQUATION

The modified Reynolds equation for modeling gas lubricated bearings is now available in the Hydrodynamic Bearing interface in the Rotordynamics Module. In previous versions, the MEMS Module license was required for this feature.

VISCOUS DISSIPATION

The viscous dissipation is now computed in the Hydrodynamic Bearing interface. This makes it possible to compute the temperature distribution in the lubricant. The variable containing the dissipated energy is named `hdb.Qvd`.

New Applications in Version 5.3

ROTOR BEARING SYSTEM SIMULATOR

The Rotor Bearing System Simulator application can be used to design and analyze rotor systems, consisting of a rotor, and varying disks and bearings. The app user can modify the rotor settings, enter geometrical and inertial properties of the disks, and specify the bearing stiffness. The visual results include whirl and Campbell plots, and the critical speeds are calculated.

Semiconductor Module

New Functionality in Version 5.3

The following new functionality is available:

- A new physics interface: Schrödinger Equation. This physics interface solves the Schrödinger equation for general quantum mechanical problems as well as for the electron and hole wave functions in quantum-confined semiconductors under the assumption of the envelope function approximation.
- Better convergence for current-driven Schottky contacts.
- More options for the trap density specification.
- A user-defined impact ionization model.

New Application and Models in Version 5.3

NEW APPLICATION: SUPERLATTICE BAND GAP TOOL

The superlattice band gap tool helps the design of periodic structures made of two alternating semiconductor materials (superlattices). The tool uses the effective mass Schrödinger equation to estimate the electron and hole ground state energy levels in a given superlattice structure.

NEW AND UPDATED MODELS

New models:

- Double Barrier 1D
- ISFET
- MOSCAP 1D
- Si Solar Cell 1D

The Bipolar Transistor Thermal has been improved and updated.

Structural Mechanics Module

New Functionality in Version 5.3

SPECIAL STUDY STEP FOR BOLT PRE-TENSION

When using **Bolt Pre-Tension** and **Bolt Selection**, one extra degree of freedom (DOF) is added for each bolt. Typically, these DOFs should only be solved for once, in an initial study step. In version 5.3, the bolt DOFs are only active when you use the new **Bolt Pre-Tension** study type, while they are inactive in all other types of study steps. This means that you do not any longer actively have to make manual changes of the status for these DOFs in the solver sequence.

Study sequences in models created in a version prior to 5.3 are not affected. In such a model you will also, in the settings for the **Bolt Pre-Tension** node, find a **Solve in bolt pre-tension study only** check box through which you can control the solver behavior for any new study sequence you may generate.

AUTOMATIC DETECTION OF BOLTS IN A SYMMETRY PLANE

Sometimes a symmetry plane cuts through a bolt. This can now be automatically detected. Modeling is then simplified, since both the given pre-tension force and computed bolt forces are interpreted with respect to a full bolt. This functionality is controlled by the **Automatic symmetry detection** check box in the settings for the **Bolt Selection** node.

RIGID MOTION SUPPRESSION

For cases where the loads are self-equilibrating, the actual locations of the constraints are not important, as long as they fulfill two conditions: rigid body motions are not possible, and no reaction forces are introduced. A new boundary condition, **Rigid Motion Suppression** has been introduced for handling such cases. If you add this boundary condition to a number of domains, a suitable set of boundary conditions is automatically added, based on the type of physics interface and geometrical dimension. The **Rigid Motion Suppression** boundary condition is available for the following physics interfaces:

- Solid Mechanics (3, 2D, 2D axisymmetric)
- Shell (3D)
- Plate (2D)

- Membrane (3D, 2D)
- Beam (3D, 2D)
- Multibody Dynamics (3D, 2D)

BUCKLING STUDY TYPE FOR THE BEAM INTERFACE

It is now possible to perform linearized bucking analysis also using the Beam interface. This makes if easier to analyze critical loads for various frame structures under compression. Also, models in which different structural mechanics interfaces are mixed can now use this study type, since it was available with, for example, the Solid Mechanics and Shell interfaces already in previous versions.

MODE ANALYSIS IN SOLID MECHANICS

Mode analysis has been added in the Solid Mechanics interface. This makes is possible to study waves traveling in the out-of-plane direction for 2D structures and also circumferential modes in axially symmetric models. Mode analysis is particularly important in coupled acoustic-structure problems. You enable mode extension in the **2D Approximation** and **Axial Symmetry Approximation** sections, respectively, in the settings for the Solid Mechanics interface.

NEW FRAMEWORK FOR INELASTIC STRAINS IN GEOMETRICALLY NONLINEAR ANALYSIS

For the case of geometric nonlinearity, a more rigorous handling of the decomposition into elastic and inelastic deformations has been implemented. In previous versions, an additive decomposition approach was used with a few exceptions (most notably, large strain plasticity), which uses a multiplicative decomposition approach.

In COMSOL Multiphysics version 5.3, the default option uses a multiplicative decomposition of deformation gradients for all inelastic contributions in studies where geometric nonlinearity is active. The main advantage with this approach is that it is possible to handle several large inelastic strain contributions in a material. Also, linearization will be more consistent; as an example, it is now possible to accurately predict the shift in eigenfrequencies caused by pure changes in the size due to thermal expansion.

A multiplicative decomposition is, for finite deformations, order dependent. The contributions are treated in the order that thee appear in the model tree.

The default is to use the new framework. If you want to switch to the behavior in previous versions, there is a new **Additive strain decomposition** check box in the settings for the material models.

NEW MULTIPHYSICS COUPLINGS BETWEEN STRUCTURAL MECHANICS INTERFACES

Connecting different types of structural mechanics interfaces is significantly easier through the introduction of three new multiphysics couplings: **Solid-Shell Connection**, **Solid-Beam Connection**, and **Shell-Beam Connection**. The old **Beam Connection**, **Shell Connection**, and **Solid Connection** nodes inside the Solid Mechanics, Shell, and Beam interfaces are now obsolete, since the same functionality is available in a more convenient manner using these new multiphysics couplings.

With the **Solid-Shell Connection** and **Solid-Beam Connection** couplings you can connect domains from either the Solid Mechanics or the Multibody Dynamics interfaces.

ELASTIC LAYERS DESCRIBED BY PHYSICAL DATA

You can now prescribe the elastic properties of a **Spring Foundation** or a **Thin Elastic Layer** also by material data such as Young's modulus and shear modulus, together with a given thickness of a layer. This simplifies the modeling of, for example, adhesive layers.

Since a layer thickness is known, it is also possible to compute strains in the elastic layer when this type of input is used, and such variables have been made available.

STRESS LINEARIZATION

Stress linearization is a technique where the stresses through a thin section in a solid model are represented by a constant membrane stress and a linearly varying bending stress. Using this type of evaluation is common when analyzing pressure vessels. It is described in *ASME Boiler & Pressure Vessel Code, Section III, Division 1, Subsection NB*. Other fields of application are for computing reinforcement in concrete structures, and for some types of weld analysis.

The new **Stress Linearization** node make it possible to select edges along which a stress linearization is to be performed. Membrane, bending and peak stresses are reported, and stress intensities are computed for each such stress classification line.

SAFETY FACTOR COMPUTATION

The new safety factor evaluation allows you to study the utilization of the materials in the structure. You can add it to a **Linear Elastic Material** and a **Nonlinear Elastic Material**. The safety factor can be computed with respect to a large number of different failure criteria. When a **Safety** node is present, you will get access to postprocessing variables for safety factor, margin of safety, damage index, and failure index.

NEW DATA SET FOR SHELL RESULTS

A new type of data set, **Shell**, has been added under **Results**. When this data set is used, results will be plotted on two parallel surfaces, which as a default are separated by the shell thickness. You can also modify the separation to improve visualization of very thin shells. Result which depend on the through-thickness position, like stresses and strains, will be evaluated at each side, so you will get a correct plot with a 3D look and feel. When using the Shell interface, a Shell data set is automatically added when default plots are generated.

SPRING BOUNDARY CONDITIONS FOR RIGID DOMAIN AND RIGID CONNECTOR

The **Rigid Connector** and **Rigid Domain** features in all physics interfaces have been augmented with a spring boundary condition, called **Spring Foundation**, with the following properties:

- The spring can be attached at an arbitrary position
- Both translational and rotational springs are available
- The spring can have loss factor damping
- The spring can act in parallel with viscous damping (for both translation and rotation)

RIGID DOMAIN IN SHELL AND BEAM INTERFACES

The **Rigid Domain** material model has been added also to the Shell and Beam interfaces. This is an efficient modeling technique for parts that are significantly stiffer than the surroundings, since it only requires the degrees of freedom of a rigid body for an entire set of boundaries (Shell) or edges (Beam). Just as for the corresponding material model in the Solid Mechanics and Multibody Dynamics interfaces, it is possible to apply loads, springs, and inertia at arbitrary locations on the rigid body.

RIGID CONNECTOR IN BEAM INTERFACE

A **Rigid Connector** condition has been added to the Beam interface. A set of nodes can be selected to form a rigid region. This can, for example be used for not overestimating the flexibility at beam connections. It can also be a means of applying off-center loads, springs, or extra inertia contributions.

OFFSET OPTION FOR POSITIONS IN RIGID DOMAIN AND RIGID CONNECTOR

In all types of selection of positions in the **Rigid Domain** and **Rigid Connector** features, such as center of rotation, or location of forces, it is now possible to add an offset vector to the position specification.

EXTERNAL MATERIAL ENHANCEMENTS

Several enhancements have been made to the usability of material models created with user-defined C-code:

- Inelastic strain contributions for nonlinear elastic and hyperelastic materials can now be implemented
- Two new interfaces to the C code that takes several more quantities as input, including deformation gradients
- Serendipity shape functions can now be used
- Small strain formulation has been added
- You can start by making a special initialization call to the user function
- You can make a cleanup call to the user function (for example closing files)
- The state variables can be named individually
- The time is available as an input argument

EXTERNAL STRAIN ENHANCEMENTS

In the **External Strain** node, you can now include inelastic strain contributions to the material in several ways:

- Explicit entry of strain tensor, deformation gradient or its inverse, or stretches.
- Incorporation of strain tensor, deformation gradient, or its inverse, announced from other physics interfaces.
- As before, the inelastic strain can be computed from a user coded external function.

ENERGY VARIABLES FOR CONTACT

New variables containing energy quantities for contact have been added. You can now obtain the energy dissipated by friction, as well as the elastic energy stored in the contact penalty factors. This is useful for checking energy balances, as well as a posteriori checks on selected penalty factors. The variable are:

- Energy dissipated by friction: `solid.Wfric`
- Total energy dissipated by friction: `solid.Wfric_tot`

- Energy stored in penalty stiffness: `solid.Wcnt`
- Total energy stored in penalty stiffness: `solid.Wcnt_tot`

FREQUENCY DOMAIN ANALYSIS WITH CONTACT

The frequency response of a structure where a contact state has been computed in a previous study can now be analyzed. As an example, you can perform frequency domain analyses of bolted structures and study the influence of contact states on the dynamic properties.

HARMONIC PERTURBATION FOR PRESCRIBED VELOCITY AND ACCELERATION

In the Shell, Plate, and Beam interfaces, you can provide values for harmonic perturbation for the **Prescribed Velocity** and **Prescribed Acceleration** nodes.

IMPROVED PERIODICITY CONSTRAINTS IN PLATE AND SHELL INTERFACES

In the settings for **Periodic Condition** in the Plate interface, the sector angle when using the **Cyclic Symmetry** option is now as a default automatically determined. It can, as before, also be set manually.

When using the same functionality in the Shell interface, it is now possible to automatically remove superfluous constraints when parts of the shell is located on the axis of rotation.

New Model in Version 5.3

TRANSIENT ROLLING CONTACT

This conceptual example shows how to handle a transient contact problem with stick-slip friction transition. A soft hollow pipe subjected to gravity load is released at the top of a halfpipe. The motion varies between sliding and rolling, depending on its position in the halfpipe and its velocity. The cross section of the pipe is also ovalized due to the contact and inertial forces. An energy balance validates the accuracy of the solution.

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 5.3

NEW TRANSPORT OF DILUTED SPECIES IN FRACTURES INTERFACE

The new Transport in Diluted Species in Fractures interface is used to model mass transport along thin fractures occurring in solid or porous media. The interface is defined and solved on boundaries in the model geometry, assuming that the fracture thickness is very small compared to its other dimensions.

NEW FRACTURE NODE IN THE TRANSPORT OF DILUTED SPECIES IN POROUS MEDIA INTERFACE

NEW INTERNAL WALL NODE IN THE DARCY'S LAW, RICHARDS EQUATIONS AND TWO-PHASE DARCY'S LAW INTERFACES

The Darcy's law, Richards Equations, and Two-Phase Darcy's Law interfaces can now define thin interior walls. This is useful to avoid meshing thin impermeable structures embedded in porous media, such as retaining walls, plates, and slabs.

NEW THIN BARRIER NODE IN THE DARCY'S LAW AND RICHARDS EQUATIONS INTERFACES

The Darcy's Law and Richards Equations interfaces can now define permeable walls on interior boundaries. These internal boundaries are typically used to represent thin low permeability structures. The new Thin Barrier feature allows to avoid meshing thin structures like geotextiles or perforated plates.

NEW WELL NODE IN THE DARCY'S LAW, RICHARDS EQUATIONS AND TWO-PHASE DARCY'S LAW INTERFACES

The Darcy's law, Richards Equations, and Two-Phase Darcy's Law interfaces include the possibility to model wells. The new Well node makes it possible to select edges in 3D or points in 2D where either Injection wells or Production wells are active.

New Model in Version 5.3

AQUIFER WATER TABLE MODEL

This tutorial model demonstrates the application of COMSOL Multiphysics to a benchmark case of steady-state subsurface flow and transient solute transport along a vertical cross section in an unconfined aquifer. The solute transport is subjected to highly irregular flow conditions with strong anisotropic dispersion.

Wave Optics Module

New Functionality in Version 5.3

NEW FEATURE

A **Surface Magnetic Current Density** boundary condition node is now available.

UPDATED FEATURE

The **Scattering Boundary Condition** node for Electromagnetic Waves, Transient can now be used also under high loss conditions.

NEW FAR-FIELD POSTPROCESSING VARIABLES

- `EIRP` and `EIRPdB`: effective isotoropic radiated power and its dB-scaled value.
- `gainEfar` and `gaindBEfar`: gain excluding input mismatch and its dB-scaled value.
- `rGainEfar` and `rGaindBEfar`: realized gain including input mismatch and its dB-scaled value

NEW SETTINGS FOR ENHANCED USABILITY

In the Electromagnetic Waves, Frequency Domain physics interface:
physics-controlled meshes reads the wavelength from study steps automatically.

For the periodic port:

- The maximum frequency is defined automatically from study steps.
- The angles of incidence are defined only with the excitation port.

For frequency-domain modal analysis:

- The eigenfrequency search method around shift is now set as Larger real part.
- The `linper` operator is applied internally in an excited port or lumped port feature.

RENAMED TUTORIAL MODEL

The Defining a Mapped Dielectric Distribution of a Metamaterial Lens model has been renamed as Defining a Mapped Dielectric Distribution of a Material.

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 5.3

Data has been added for the following materials: C36000, P92, VM12-SHC (ASME Code Case 2781), IN-939 SSC, C26000, 17-4 PH, Copper, A182 Grade F1 (Ravne No. 760), A537 (S185), Silicon, BS 460B steel reinforcing bars, BS B500B steel reinforcing bars, 1040 steel, SMn443, JIS G 4801 SUP12, 52100, 6150, G4802 SK5-CSP, ASTM A723, PMMA, BK7 glass, GaS, 28CrMoV5-8, 5132, 9310, Ferrium C61 (AMS 6517), Ferrium S53, Inconel 803, Inconel 864, 201, 9310, Ferrium C64, Ferrium M54, AerMet 100, 304LN, 718, Al2TiO5, NaBr, Sulfur, Alloy D9, A319, MAR-M247, GRCop-84, Dupont Delrin resins, Dupont Crastin, Hytrel, Minlon, Rynite, Zenite, Zytel resins, Elf Atochem KYNAR resins, Petra 140 BK-112, Lexan 101, Celacex 1600A, Pro-fax 6823, ULTEM PEI resin, 1070, TRIP 800, P91, ASME P23, T23 steels, ductile iron 80-55-06 (NE-GJS-600), 2101 duplex stainless, 2024 Al, and for the following Cr steels: 1.4037 (X65Cr13), 1.4122 (X39CrMo17-1), 1.4742 (X10CrAlSi18), 1.4910 (X3CrNiMoBN17-13-3), 1.3964 (X2CrNiMnMoNNb21-16-5-3), and 430F.

Also, data for several electronic potting materials was added.

For steels 2304, 2507, S32304, and S32750, an additional reference was added for the elastic modulus resulting in the values changing by 0 to 5%. For M2 steel an additional reference was added for the expansion resulting in the values changing by 0 to 2%.

LiveLink™ for MATLAB®

New Functionality in Version 5.3

LiveLink™ *for* MATLAB® now supports MATLAB® R2016b and R2017a.

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

ACCESS TO FUNCTIONS FROM THE APPS TAB OF THE TOOLSTRIP IN MATLAB®

By installing the `LiveLink for MATLAB.mltbx` file you gain access to four functions from the Apps tab of the toolstrip in MATLAB®. These functions are: `mphmodellibrary`, `mphnavigator`, `mphopen`, and `mphsearch`. This makes it easy to perform common tasks such as loading and searching for models, navigating the model object structure, and viewing the model settings.

MPHRAY FUNCTION

The new `mphray` function has been added to support ray optics and ray acoustics data sets.

MPHPARTICLE FUNCTION

The `mphparticle` function now supports the properties `'solnum'`, `'outersolnum'`, `'times'` and `'velocities'`. These new options give improved access to particle data from parametric sweep studies.

MPHPLOT FUNCTION

Support is added for the following plot types: Height Aberration Plots, Arrow Surface Plots (normal and tangent angles), Streamline Surface Plots, and Line Plots with two y-axes.

MPHPLOT, MPHGEOM AND MPHMESH FUNCTIONS

Support for views has been introduced for these functions. This adds support for some of the view settings associated with the plot when creating the plot in MATLAB®. This is controlled by the property `'view'` that can have the value `'auto'` for automatically selecting a view based on the plot type, or a specific tag for the view to be used.

The new syntax is

```
mphplot(model, 'pg1', 'view', 'view1')
mphgeom(model, 'geom1', 'view', 'view1')
mphmesh(model, 'mesh1', 'view', 'view1')
```

MPHTHUMBNAIL FUNCTION

The new mphthumbnail function enables the setting and getting of the thumbnail images of models. Note that model thumbnails are now preserved by default when loading and saving models. In previous versions, these were silently removed.

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 is no longer supported starting with version 5.3 of COMSOL Multiphysics.

New Functionality in Version 5.3

The following new functionality is available:

- Buttons for loading and saving COMSOL Multiphysics models and handling server connections have been moved from the **COMSOL** ribbon tab to the **File** tab in Excel®. This leaves more space on the COMSOL tab for functionality for working with the COMSOL Multiphysics models in the Excel workbook.
- To enable easier access to the product documentation, all windows and dialog boxes now have a **Help** button to open context-sensitive help.
- Two new buttons have been added to the **Numerical Results** section of the **COMSOL** ribbon tab. The **Result Parameters** button provides access to parameters defined in the **Parameters** node in the **Results** section of the Model Tree. When clicked, the **Clear and Evaluate All** button first clears all table entries and then evaluates all derived values.

The COMSOL API for Use with Java®

COMSOL 5.3 API Changes

NEW COMPONENT SYNTAX

The geometry component is now included in the code for features that appear in a component. For example,

```
model.geom("geom1")
```

in earlier versions is now instead

```
model.component("comp1").geom("geom1")
```

to indicate which component the geometry belongs to. To keep the old syntax in generated Model Java® files, clear the **Use component syntax** check box under **Code generation** on the **Methods** page in the **Preferences** window. All old scripts and Java® source code files will continue to work using the old syntax.

ECAD IMPORT MODULE CHANGES

- The default value for the `sellayer` property of the `Import` function is now `on`. Previously, the default was `off`.
- The default value for the `sellayershow` property of the `Import` function is now `all`. Previously, the default was `dom`.

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 1-4: 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()`
`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()`

TABLE 1-4: DEPRECATED METHODS AND THEIR REPLACEMENTS

DEPRECATED METHOD	NEW METHOD
`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.
`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 5.3.

Index