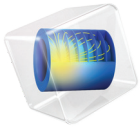

COMSOL Multiphysics

Release Notes

6.4

COMSOL

COMSOL Multiphysics Release Notes

Version: COMSOL 6.4

Contact Information

Visit the Contact COMSOL page at www.comsol.com/contact to submit general inquiries 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 on the COMSOL Access page at www.comsol.com/support/case. Useful links:

- Support Center: www.comsol.com/support
- Product Download: www.comsol.com/product-download
- Product Updates: www.comsol.com/product-update
- COMSOL Blog: www.comsol.com/blogs
- Discussion Forum: www.comsol.com/forum
- Events: www.comsol.com/events
- COMSOL Video Gallery: www.comsol.com/videos
- Support Knowledge Base: www.comsol.com/support/knowledgebase
- Learning Center: www.comsol.com/support/learning-center

Part number: CM010001

Contents

Chapter 1: Release Notes

1

Release Notes

COMSOL Multiphysics® version 6.4 includes a new product: the Granular Flow Module. It also provides new functionality and improvements across the COMSOL products. These *Release Notes* provide information regarding new functionality in version 6.4 for the entire COMSOL product suite.

COMSOL Multiphysics

New Product

GRANULAR FLOW MODULE

The Granular Flow Module, new in version 6.4, offers a Granular Flow interface based on the discrete element method (DEM). DEM is a numerical technique that simulates granular flow by tracking individual particles (or grains) over time, resolving them as discrete entities with both translational and rotational degrees of freedom. Their dynamics are governed by Newton's laws of motion, with forces arising from gravity, collisions with other grains, and interactions with enclosing boundaries.

In this framework, the grains may represent powders, pellets, rocks, seeds, tablets, or other bulk solids. The module supports a wide spectrum of physical interactions: elastic and viscoelastic contact forces, adhesion, rolling resistance, and even heat transfer between grains or between grains and walls. During a simulation, grains are treated as soft particles that deform elastically when in contact; at each time step, grain–grain and grain–wall collisions are detected, forces are computed via contact models, and the trajectories of the grains are updated accordingly to predict the collective motion of the system.

You have precise control over how grains are introduced: initial release conditions, velocity distributions, and spatial arrangements can be specified. You can also define how the grains interact with walls or leave the simulation domain. Additionally, the Granular Flow interface provides a built-in feature to model the motion of walls, permitting moving boundaries in the simulation.

Applications can be found within various industries, including food processing, pharmaceuticals, chemical industries, agriculture, cosmetics, and plastics. Several tutorial examples are provided to illustrate key uses such as grain packing, hopper flow, powder spreading, mixing of grains in a ribbon mixer and in a rotating drum, separation of grains in vibrating sieves, and transport of grains using a screw conveyor.

New General Functionality

UPDATED LICENSE SERVER

The FlexNet® license manager software has been upgraded to version 11.19.8. Before installing and running version 6.4 of the COMSOL software, you will need to update

the license manager on your license server. This can be done by running the COMSOL Installer on the license server and, in the list of products, only selecting License Manager. For more information, see the *COMSOL Multiphysics® Installation Guide*.

LMADMIN LICENSE SERVER MANAGER

This functionality was removed in version 6.3 update 1 due to security vulnerabilities.

New Functionality in the Model Manager

Also see the *Model Manager Reference Manual* for more information.

STUDY SUPPORT FOR BATCH AND CLUSTER

The **Batch**, **Batch Sweep**, **Cluster Computing**, and **Cluster Sweep** study nodes can now be configured to use models and data stored in a Model Manager database. You enable database storage for a study node by selecting **Database** in the **Model storage** and **Data storage** lists in the **Settings** window.

FIND MODELS SAVED IN A SPECIFIC COMSOL VERSION

Support for filtering and sorting models by the COMSOL Multiphysics version that a model was saved in has been added to the Model Manager search tool. The new functionality is available via a **Saved in** menu option in the **Open**, **Model Manager**, **Select File**, and **Select Model** windows, as well as via the Model Manager search syntax using `@savedIn:6.4`, for example.

MODEL MANAGER API IMPROVEMENTS

The following new methods and types have been added to the API:

Access Database Objects Using a Location URI

New methods have been added to `DatabaseApi` for accessing the corresponding database, repository, branch, and latest version given a model or file location URI for a specific model or file version. For example,

```
DatabaseApi api = DatabaseApiUtil.api();
String modelLocationUri = "dbmodel:///?...";
Branch branch = api.branchByLocationUri(modelLocationUri);
```

Test for Existence

You can test for the existence of items and item versions in a database via new `exists` methods added to, for example, `ModelItem`, `ModelItemVersion`, and `BranchModelItem`. This is useful when, for example, you want to test if the version

corresponding to a model location URI has been permanently deleted in the database before proceeding with other logic:

```
ModelItemVersion modelVersion = api
  .modelVersionByLocationUri(modelLocationUri);

if (modelVersion.exists()) {
  // ...
}
```

Manipulating Location URI Strings

Some new convenient methods for manipulating model and file location URI strings have been added, including methods for:

- Reading the file resource path of a file location URI obtained, for example, from a fileset:

  ```
  String fileLocationUri = "dbfile:///cad_component.prt?...";
  FileResourcePath path = FileResourcePath
    .ofFileLocationUri(fileLocationUri);
  ```

- Writing a new file location URI for a file version using a specific file resource path:

  ```
  FileResourcePath path = ...
  FileItemVersion fileVersion = ...
  String fileLocationUri = fileVersion.fileLocationUri(path);
  ```

- Writing a new model location URI for a new, but not yet created, model item to be used, for example, for the `-outputfile` command line argument when running COMSOL batch:

  ```
  Branch branch = ...
  String title = "Batch Output";
  String modelLocationUri = branch.newModelLocationUri(title);
  ```

Partial Updates to Filesets

A new input parameter type, `TargetFileResourceParam`, has been added, enabling you to perform a partial update of file resources belonging to a fileset version of a file item. Two examples of updates you might make and the methods you would use include:

- Saving a new version that updates an existing file resource with new file contents while keeping any other file resource unmodified:

  ```
  BranchFileItem branchFileItem = ...

  TargetFileResourceParam targetFileParam = DatabaseApiUtil.param()
    .forTargetFileResource()
    .withSourceFileResourcePath("cad_component.prt")
    .withReplaceExisting();
  ```

```
UpdateFileItemParam updateFileParam = DatabaseApiUtil.param()
  .forUpdateFile()
  .withTargetFileResources(targetFileParam);

String commitComment =
  "Updated the ‘cad_component.prt’ file in the CAD assembly.";
branchFileItem.update(updateFileParam, commitComment);
```

- Deleting a specific file resource while keeping other file resources unmodified:

```
TargetFileResourceParam targetFileParam = DatabaseApiUtil.param()
  .forTargetFileResource()
  .withFileResourcePath("cad_component.prt")
  .withDeleteExisting();

UpdateFileItemParam updateFileParam = DatabaseApiUtil.param()
  .forUpdateFile()
  .withTargetFileResources(targetFileParam);

String commitComment =
  "Removed the ‘cad_component.prt’ file in the CAD assembly.";
branchFileItem.update(updateFileParam, commitComment);
```

Optionally Ignore Missing Files When Exporting

A new option has been introduced to silently ignore when a file resource is missing from a fileset while exporting.

Use this method to ignore if a CAD assembly has a specific component file or not:

```
ExportFileItemVersionParam exportParam = DatabaseApiUtil.param()
  .forExportFileVersion()
  .withSourceFileResourcePathSegments("cad_component.prt")
  .withSourceFileResourceOptions(ExportFileItemVersionSourceFileR
esourceOption.IGNORE_MISSING);

fileVersion.export(exportParam);
```

Without this option, the export would fail with an exception if the component file was not found in the database.

Enum Constant for Saved In

A new `SAVED_IN` enumeration value for sorting on the COMSOL Multiphysics version that a model was saved in has been added to `ItemSortField`.

More Lenient Behavior When Saving a Target Item

The target item may now be absent from the database when providing an item key via `SaveItemParam.targetItemKey()`. Instead, a new item is created as long as necessary source data for field values and item contents is available via other input

parameters. The same holds true when invoking `BranchModelItem.update(...)` when the model item is absent (as well as for file items and tag items).

Deprecated Enum Constants

The `COMPUTEDDATA` constant in `ItemSortField` and the `INPUTFILE`, `OUTPUTFILE`, and `GEOMETRYPART` constants in `ItemVersionReferenceType` have been deprecated. Their replacement constants are found in the same enum types.

COMPATIBILITY BETWEEN VERSIONS

Local Databases

Local databases can be shared between COMSOL Multiphysics versions 6.0 through 6.4, as the database format is designed for both backward and forward compatibility. Models saved to a database from a specific COMSOL Multiphysics version can, however, only be opened in that and later versions.

Older Versions of Model Manager Server

COMSOL Multiphysics version 6.4 supports connecting to all previous versions of a Model Manager server, although some new Model Manager functionality is disabled in the COMSOL Desktop when connected to an older server version.

Version Conflicts

The version conflict analysis involving branches has been updated. It is no longer considered a conflict to save to a branch that a model did not have any prior versions in. It is also possible to update a model on a parent branch when opened from a child branch without conflicts as long as the model's latest version on the parent branch is found in the version history (with respect to the child branch) of the model being saved from.

Rank Tie-Breaker

The tie-breaking sort field when sorting by relevance ranking is now always the saved timestamp (in descending order). Previously, the title field was used as the tie-breaker for the special case of an empty search term.

New Functionality in the Model Manager Server

Also see the *Model Manager Server Manual* for more information.

WORKING WITH MODELS IN THE WEB INTERFACE

The support for working with models via the Model Manager server web interface has been improved. You can now save new versions of models to your repositories by

uploading MPH files as well as download existing model versions as MPH files. Any file saved in COMSOL Multiphysics version 5.3 or later may be uploaded and downloaded in this manner. Uploaded and downloaded models keep their model format as originally saved in COMSOL Multiphysics, which means that they can also be opened in that specific COMSOL Multiphysics version in the future.

The model tree of a model can now be viewed on its **Model** page. This enables you, for example, to quickly get an overview of a model's capabilities by browsing its components, geometries, physics, studies, results, and other model nodes. You can see the required add-on products for building the model in a new **Product Requirements** field on the page.

If you have an installation of COMSOL Multiphysics 6.4, you can open the model in the COMSOL Desktop by clicking a new **Open in COMSOL Desktop** button on the **Model** page. The model can be opened in an already running COMSOL Multiphysics program session if available; otherwise, a new program session is launched. For applications, you can also click a new **Run in COMSOL Desktop** option.

BETTER SUPPORT FOR ADDING MODELS AND FILES TO ASSETS

Adding links to models in your repositories on the **Asset** page has been streamlined. A new in-page search field enables you to quickly find the model version you want to add. You can also press Ctrl+V to paste the model location URI of a model version into the field directly from your clipboard. The location URI can be copied, for example, from the **Copy Location to Clipboard** button on the **Model** page or via the **Copy Location** context menu option in the Model Manager workspace. The same functionality is also available when adding links to data files.

Assets linking to a model or data file are now displayed in a new **Assets** table on the **Model** and **File** pages, thus helping you to better navigate the dependencies and relationships in your database.

USING WORKFLOWS TO SET UP ORGANIZATIONAL PROCESSES

A new workflow management system has been added to the Model Manager server web interface. The system enables you to set up processes and other business rules relevant to your organization by adding *workflows* to your assets. This includes, for example, defining the various stages of a project, the group of people involved in each project stage, and how a project may evolve between different stages.

By adding workflows to the asset management system, you can:

- Designate a primitive attribute as representing the various stages of a process, with the current value of the primitive attribute corresponding to a specific stage. A process could be defined for the asset itself or for individual table rows of a composite attribute. Any number of primitive attributes may be designated in this way.
- Let a specific value for a primitive attribute be its default stage. When adding a new asset, or when adding a new table row to a composite attribute that has the primitive attribute as a column, the field is automatically populated with this value.
- Only allow the values for a primitive attribute to change according to predefined transition rules. The transitions are shown on the **Asset** page as buttons that, when clicked, updates the asset with the new value.
- Grant permissions that controls which users can perform specific transitions.
- Grant permissions to primitive and composite attributes. You can restrict all but a select group of users from updating the value of a specific attribute.
- Grant permissions to assets based on their primitive and composite attribute values. You can, for example, let the value of an attribute determine which users can open the **Asset** page for an asset version or save new versions of an asset.
- Conditionally set a new value for a primitive attribute when some transition occurs.
- Conditionally cancel the save of an asset when some transition occurs.

MODEL MANAGER SEARCH TOOL IMPROVEMENTS

Some general improvements to the Model Manager search tool in the web interface include:

- You can now display the values of attributes as search result columns. Any primitive attribute or composite attribute in **Simple** mode may be added as a column.
- A new search filter for the **Identifier** field of an asset is available in the **Filters** menu. The field can also be shown as a search result column.
- The description of an attribute is shown as a tooltip in the **Filters** menu.
- The web interface now automatically detects if there is a syntax error when typing filters using the Model Manager search syntax. A warning message is displayed below the problematic input field.

ATTRIBUTES IN THE ASSET MANAGEMENT SYSTEM

New capabilities have been added for attributes in the asset management system:

- A new **Users picker** attribute enables you to define a field that can store multiple users as its value.
- A new **Datetime picker** attribute is available for selecting a date and a time.
- You can easily sort the values for an **Input fields** widget on the **Asset** page using a new **Sort alphabetically** button.
- The allowed values of a primitive attribute can now have a custom sort order. This is useful for options that have a natural sort order other than alphabetical. You can also use a separate label for an allowed value different from the value itself.
- Members of a composite attribute can be set as *primary*. The column values of primary members are guaranteed to be unique and nonempty for composite attributes displayed as tables. This enables you, for example, to set up workflow rules for the individual rows of a table on the **Asset** page, with the primary value identifying the row itself.
- Primitive and composite attributes created on pages for adding and editing asset types can now be edited or permanently deleted without leaving those pages.

GENERAL IMPROVEMENTS

- A new **Time zone** field is shown on the **My Account** page. The value is the time zone currently used when displaying date and time fields in the web interface, as well as the time zone used when evaluating search filters involving timestamps.
- The current uptime of managed server components is displayed on their respective pages in the **System** administration area.
- The memory usage of a managed search index server can now be monitored in the **System** administration area. The maximum allocated memory can be increased if the usage remains close to the maximum for extended periods of time.

COMPATIBILITY BETWEEN VERSIONS

Older Versions of COMSOL Multiphysics

COMSOL Model Manager server version 6.4 supports client connections from COMSOL Multiphysics version 6.0 and newer.

Renamed Search Fields

The field identifiers `@library` and `@sizeInBytes` have been renamed to `@assetLibrary` and `@size`, respectively.

Fixed Value Types

The **Value** field for an **Input field** and **Text area** are now always **Keyword** and **Text**, respectively. The previously allowed **Date** and **Integer** value types should instead be combined with a **Date picker** and **Number** widget, respectively.

New Functionality in the Application Builder

Also see the *Introduction to the Application Builder*, *Application Builder Reference Manual*, and *Application Programming Guide* for more information.

GENERAL IMPROVEMENTS AND CHANGES

- You can now record code directly to the **Java Shell** window. A **Record Code** button has been added to the **Java Shell** window. When clicked, changes to the model or application will be recorded directly to the input area of the **Java Shell** window. This is useful, for example, if you want to find out how to change something programmatically in the model in order to then run it from the **Java Shell** window directly. Otherwise, **Record Code** works the same as when recording to methods. You can stop the recording from the **Java Shell** input toolbar or from the ribbon where you could previously stop method recording (in the Model Builder on the **Developer** tab; in the Application Builder on the **Home** tab or the **Method** tab).
- The **Data Viewer** window now has a toolbar. It includes the following buttons:
  - **Edit Node**: If a variable in the **Data Viewer** window is selected and it is part of the model, you can use this button to go to the definition of this variable in the model.
  - **Collapse All**: Collapses the tree in the **Data Viewer** window. You can also open its menu to choose **Collapse Selected** and **Collapse Other**.
  - **Expand Main Levels**: Expands all "header nodes" so that all variables are visible in the **Data Viewer** window. You can also open its menu to choose **Expand Selected**.
  - **Full Precision**: Toggles full precision on and off for the variables in the **Data Viewer** window.

    Note that the **Data Viewer** toolbar with only the **Full Precision** button is available in version 6.3 update 1.
- When opening the **Java Shell** context menu on a results node that corresponds to a variable in the **Data Viewer** window, there is a new **Show in Data Viewer** option. This option opens the **Data Viewer** window and highlights the corresponding variable there. The **Show in Data Viewer** option in the toolbar works in the same way, taking the currently selected node into account and highlighting that variable if possible.

If the current selection does not correspond to a variable in the **Data Viewer** window, the window is opened as before without highlighting anything, and the **Java Shell** branch in the tree table is expanded (if present).

- If you right-click on a shell variable in the **Data Viewer** window, a **Delete From Session** option now appears. This option removes the variable from the current Java Shell session.
- It is now possible to create multiple primitive declaration nodes of the same type in the same list for all types of declarations. You can group the declaration nodes if desired for additional structure of data declarations in an app.
- More properties are now available for **Data Access**; that is, they get a green checkbox when this option is enabled. The improved data access support includes properties on the **Model (root)** node as well as for properties under **Component**, **Geometry**, **Mesh**, **Study**, and **Results**.
- When method code prints to System.out or System.err, each completed line is sent to the **Debug Log** window, in addition to where it was sent in 6.3 (for example, a console window if started from a Linux terminal). This output does not happen if the method is called from the **Java Shell** window because in that case, the output already goes to the **Java Shell** window like in version 6.3.

IMPROVEMENTS TO THE FORM EDITOR

- The **Button** and **Toggle Button** form objects have a new option for **Size** in their settings called **Use icon size**. When a user has added images to the image library, this option can be used to show the images in their original size.
- **Combo Box** form objects with the **Allow other values** checkbox selected now have an **Enable text search** option to control if the combo box should try to find an item matching what you type in its text field. Text search is only available in the Windows client.
- It is now possible to change the color, top color, bottom color, and icon of the **Graphics** form object while an app is running. See Changing Color of Graphics Form Object below.

IMPROVEMENTS AND CHANGES TO THE METHOD EDITOR AND APPLICATION LANGUAGE

Copying from Method Errors and Warnings Window

The table in the **Method Errors and Warnings** window now supports selecting multiple cells and copying the selected cells to the clipboard. It also includes an option to select

all cells. **Copy** and **Select All** can be selected from the table context menu or using the standard keyboard shortcuts (Ctrl+C and Ctrl+A, respectively).

Improved Copy as Code to Clipboard Functionality

The action on the **Copy as Code to Clipboard** menu previously referred to as **Set All** is now called **Set All Modified Settings**. A new action, based on report-generator code, that records code also for properties with their default values, called **Set All Displayed Settings**, has been added.

Toggle Folding Expansion

Right-click in the Method Editor and choose **Toggle Folding Expansion** (or use the keyboard shortcut Ctrl+Shift+-) in the method to toggle all the expansion states of the code folds that are represented by minus signs or plus signs in the left margin of the Method Editor.

Changing Color of Graphics Form Object

It is now possible to change the color, top color, bottom color, and icon of the **Graphics** form object while an app is running. The syntax for changing this is the same as for other color and icon options. The color properties are called `"background2d"`, `"topBackground"`, and `"bottomBackground"`. They can be set either from a string (such as `"blue"`, `"black"`, and so on) or from RGB values 0–255. The icon property is called `"icon"`. Valid values are images defined in the **Images** > **Libraries** node in the application tree, or use an empty string for no icon. The default icon is called `"logo_graphics.png"`.

Examples:

```
app.form("form1").formObject("graphics1").set("background2d", "blue");
app.form("form1").formObject("graphics1").set("topBackground", "black");
app.form("form1").formObject("graphics1").set("bottomBackground", "red");
app.form("form1").formObject("graphics1").setColor("background2d", 230,
236, 232);

app.form("form1").formObject("graphics1").set("icon", "compute_32.png");
```

Multiple and Grouped Declaration Nodes

Additions to the `DeclarationGroup` objects.

To retrieve a `DeclarationGroup` object containing entries of the `"string1"` string declaration node:

```
DeclarationGroup group = app.declaration().group("string1");
```

To retrieve a `DeclarationGroup` object containing entries of the `"string1"` string declaration node under the given form:

```
group = app.form("form1").declaration().group("string1");
```

To get the value of the `"svar"` declaration among the entries:

```
String value = group.get("svar").getString();
```

Iterate over the entries of the `"integer1"` Integer declarations node:

```
int i = 0;
for (Primitive entry: app.declaration().group("integer1")) {
  entry.set(i++);
}
```

Additions to the `DeclarationGroupList` objects.

To get the names of all global primitive declaration nodes:

```
String[] names = app.declaration().group().names();
```

To get the names of the local primitive declaration nodes of a given form:

```
String[] names = app.form("form1").declaration().group().names();
```

New WriteFile Overloads for User-Specified Delimiters

Two new `writeFile()` overloads have been added to the application language to make it possible to save a string matrix with a user-specified delimiter character:

```
void writeFile(String file, String[][] value, char delimiter,
  boolean append);
void writeFile(String file, String[][] value, char delimiter,
  boolean append, boolean bom);
```

Exporting Materials

Materials and their data can be exported to the model XML file format using the following methods for global materials and materials under a component, respectively:

```
String[][] com.comsol.model.MaterialList.export(String filename, String[]
materials)

String[][] com.comsol.model.ComponentMaterialList.export(String filename,
String[] materials)
```

New and Improved General Functionality in COMSOL Multiphysics

PHYSICAL CONSTANTS

The built-in physical constants have been updated to use full numerical precision and match the CODATA internationally recommended 2022 values available at physics.nist.gov/cuu/Constants/index.html.

MATERIAL LIBRARY LINKS

You can now add **Material Library Link** nodes from the **Add Materials** and **Material Browser** windows or by choosing **Add Material from Library** in a **Material Link** node. A **Material Library Link** provides a link to a material from a material library, which can easily be updated when material properties change in the material library file. The **Material Library Link** nodes works just like any other global or local material when added to the model. You can break the link to the material in the material library to add or change its material properties.

CHATBOT

The chatbot window functionality is now installed by default on the Windows® version of COMSOL Multiphysics. Clear the **Chatbot Components** checkbox in the installer if you prefer not to install the feature.

The chatbot functionality has been extended with the following features:

- You can now use the **Chatbot** window with any Open AI API-compatible provider.
- The **Chatbot** window now supports the new GPT-5 language model when using OpenAI or Azure OpenAI as the provider. Choose between **GPT-5 Low**, **GPT-5 High**, **GPT-5 mini Low**, and **GPT 5 mini High**, where the **Low** and **High** suffixes denote the reasoning effort of the model.
- More attachment types can be added to the **Chatbot** window conversations by using the **Send to Chatbot** option from the context menus within COMSOL Multiphysics. The following attachment types have been added:
  - Snapshot images from the **Graphics** window
  - The model history as Java code
  - Nodes in the Model Builder as Java code
- You can also attach image and text files from your file system by clicking the **Add File Attachment** button in the **Chatbot** window.
- The **Chatbot** window now has built-in tools for searching the COMSOL documentation, improving the accuracy of the advice in its responses.

The **Maximum generated tokens** preference has been removed.

GENERAL CONTACT PAIR

A new **General Contact Pair** is available to specify a contact interaction between multiple objects. A general contact pair defines boundaries of parts in an assembly that can come

into contact under deformation. The pair establishes a mapping between these boundaries by performing a contact search.

SELECTIONS

- The **Logical Expression** selection features under **Definitions** support usage of the index attributes as variables. Such index attributes can now be created by the geometry operations **Array, Copy**, **Move**, and **Rotate**.
- The **Adjacent** and **Explicit** selection features under **Definitions** have new options: **Adjacent to inside** and **Adjacent to outside**.
- When adding selections to other features within COMSOL Multiphysics, a new search bar in the **Add** dialog makes it possible to filter the list of selections.

HIGHLIGHT SEARCH MATCHES

Matches from a search done in the **Find and Replace** window are now scrolled into view and highlighted (Windows® only) in the **Settings** window, expanding collapsed sections if necessary. When there are matches inside text areas, the relevant row is selected. On Linux® and macOS, matches are also scrolled into view but might not be highlighted.

THE GRAPHICS CONTEXT MENU

The content of the context menu in the **Graphics** window differs depending on the selected node in the Model Builder. The following three new options have been added to the context menu and are available regardless of which node is selected:

- **Copy Image** — Copy the current state of the **Graphics** window as an image to the clipboard. If a 2D geometry or work plane is in sketch mode, **Copy Image** is replaced by **Copy**, which copies selected geometric objects.
- **Send to Chatbot** — Attach a snapshot image of the **Graphics** window to the current **Chatbot** window conversation.
- **Help** — Receive context-based help about the available tools and buttons in the **Graphics** window.

SEARCH SUPPORT IN THE PREFERENCE DIALOG

A search field has been added to the **Preferences** dialog to make it easier to find particular preference settings.

SET PARAMETER CASES IN BOTH DIRECTIONS

Parameter cases can now be set in both directions; that is, you can now also reset the parameter values in a **Parameter Case** node to the values in the main **Parameters** node.

IMPROVED NAVIGATION ON THE COMSOL DESKTOP

- A window switcher, showing all open windows on the desktop, now appears when using Ctrl+Tab and Ctrl+Shift+Tab to navigate between open windows. In the Application Builder, the same keyboard shortcuts let you navigate between the editors.
- Press Enter in input fields to store the value and update any corresponding features accordingly.
- Close open windows on the desktop using the middle mouse button (usually the mouse wheel).

New Geometry and Mesh Functionality

GEOMETRY NEWS

- The primitives **Block**, **Sphere**, **Cylinder**, **Rectangle**, and **Circle** now have an option to define the primitive as a bounding box, ball, cylinder, or disk of a selection of input objects or entities. You can also make the primitive object larger by adding margins, or make it smaller.
- In the primitives, the **Position** section now has an option to specify the position by selecting a vertex of some other object. This option has also been added in the **Scale** feature (**Center of Scaling**) and in the **Rotate** feature (**Point on Axis of Rotation**).
- The primitives **Block**, **Sphere**, **Cylinder**, **Rectangle**, and **Circle** now have a checkbox labeled **Create parameters**, which creates parameters for the primitive's size, center, and axis directions. These parameters can be used in subsequent geometry features, physics, result features, and so on. These parameters will appear in the Ctrl-Space variable tree.
- The **Geometry** node now has options to reduce geometries by using symmetry planes in 3D and symmetry lines in 2D. Selecting these options means that the corresponding part of the geometry will be removed by the **Form Union/Assembly** feature. Also, a corresponding selection of the symmetry boundaries will be created (in 3D: **yz-plane**, **zx-plane**, or **xy-plane**).
- The **Cleanup Wizard** has been improved when it comes to detection and visualization of 3D problems as well as accuracy of details size measurements.

- The **Form Assembly** feature for a 3D component geometry now has a checkbox labeled **Fast pair detection for stacked objects**, selected by default. A set of objects is stacked if any two objects can be separated with a plane that is parallel to the *xy*-plane, which is often the case for ECAD geometries. If the checkbox is selected, and the objects are stacked, then the pairs are generated in a fast way so that you get one pair for each *z*-coordinate where two objects touch. This means that you can get a pair also in the case when the objects do not touch.
- The features **Array**, **Copy**, **Move**, and **Rotate** now have a possibility to assign index attributes to the resulting objects and their geometric entities. The **Logical Expression** selection features in the geometry sequence and under **Definitions** support usage of the index attributes as variables.
- The **Adjacent Selection** feature in the geometry sequence and the **Adjacent** and **Explicit** selection features under **Definitions** have new options **Adjacent to inside** and **Adjacent to outside**.
- For geometry features that can select objects, there are new alternatives **All objects** and **All nonconstruction objects** to increase usability. The latter selects all available geometry objects that are not construction geometry.
- The colors of projected lines in a plane geometry have been updated.
- When measuring two edges that are either straight or adjacent, the angle between them is now displayed.
- There is now a toggle action **Sketch** on the graphics context menu in 2D that enters/exits the sketch mode.
- Edges that are a result of an intersection of two faces are now represented accurately in the COMSOL kernel.
- The **Layers** section in primitives now supports creating much thinner layers.
- The 2D and 3D Boolean operations in the COMSOL kernel are faster than in 6.3.
- **Part Instance** features use less memory than in 6.3.
- If a model loads two parts, A and B, that both use another part C, the part C will now be stored twice in the model.
- Adding many global parameters to a model with many geometry parts is now much faster.

MESH NEWS

- The **Swept** operation now supports disconnected source faces. This means that some domains that previously required manual partitioning to be swept meshed can now be meshed directly.

- For **Adapt**, the **General modification** adaptation method has been improved and is now faster and more robust.
- The operations **Free Quad** and **Swept** have new options, **Quad dominant** and **Hex dominant**, respectively. In the mesh generated by these operations, triangles and prisms are inserted to aid element size transition. The same method is used when the option **Prefer Quads** is selected in **Remesh Faces** and **Remesh Domains**.
- More build data is now displayed in tables in the **Information** section of the **Settings** window. The tables can report the number of generated elements of each element type along with the minimum and average quality, the net change in element counts, added and removed entities, and imported elements and entity information.
- The option **Smooth across removed control entities** now has an **Automatic** option that turns off smoothing in adjacent domains if they contain hexahedral elements.
- There is now a new option to create a **Mesh-Based Geometry** from a **Filter** or **Partition** dataset. The option **Create Mesh in New Component** is now only available in 1D.
- It is now possible to manually add a **Mesh-Based Geometry** sequence from the **Component** node, from the **Mesh** toolbar, and from the **Filter** and **Partition** datasets in **Results**. This is the recommended sequence to use when importing meshes that need repair or editing, as it will be remeshed using the regular physics-controlled mesh. The physics will be defined on the mesh-based geometry.
- For selections that are imported to a meshing sequence via an **Import** operation, new checkboxes have been added: **Keep** and **Physics** (**Instance** in parts). When **Keep** is selected, the selection is imported and can be used locally in the meshing sequence. When **Physics** or **Instance** is selected, the selection is also visible and can be used outside the meshing sequence, such as in physics or materials. The previous option **Show outside part** has changed to **Show in instances**.
- **Ball Selection**, **Box Selection**, **Cylinder Selection**, **Union Selection**, **Difference Selection**, **Intersection Selection**, and **Complement Selection** are now also available for mesh-based geometries and imported meshes.
- Version 6.4 comes with a new mesh file format that includes support for unmeshed domains.

New Functionality in the Physics Interfaces

- The Laminar Flow interface has a new **Average Pressure** constraint under **Global** features.

- For the PDE interfaces, in the **Auxiliary Dependent Variable** and **Dependent Variable Field** features, it is now possible to set a space-dependent shape order for the shape functions that support it: **Hierarchical** and **Hierarchical serendipity**. The **Element order** list then has a new **Expression** option. When selected, an **Element order expression** field appears. The expression can be a parameter or a space-dependent expression, which must evaluate to a supported element order. You can, for example, use an if statement with a condition on the local mesh size `h` to use a lower element order where the mesh density is high.
- Luminous flux and illuminance are now available as built-in physical quantities. They also represent two new units:
  - Luminous flux (light flux): unit lumen (symbol: `lm`), defined by candela per steradian (`cd/sr`)
  - Illuminance: lux (symbol: `lx`), defined by lumen per square meter (`lm/m^2`)
- The **Yeoh** mesh smoothing method used by Moving Mesh and Deformed Geometry functionality has additional settings. You can now use a logarithmic volumetric strain and enable **HuHu** or **HuHu-LuLu** stabilization.

New and Updated Operators, Functions, and Definitions

- A **CFD Data (CGNS)** function is available as a new global function. You can use it to import CFD analysis data as CGNS files (`*`.cgns). This function requires the Acoustics Module or the CFD Module. Evaluation is only supported on Windows®.
- For **Interpolation** functions with a single argument, a new option for specifying the sampling points called **Adaptive** is available. With this option selected, the interval between two adjacent sampling points depends on the slope of the interpolation curve.
- A **Plot Parameters** section has been added to the settings for the **Gaussian Pulse**, **Ramp**, **Rectangle**, **Step**, **Triangle**, and **Waveform** functions.
- A new built-in function, `expinti(x)`, evaluates the exponential integral function $Ei(x)$.
- A new built-in global boolean variable, `issolving`, indicates whether a problem is being solved.
- The built-in operators `elemint` and `elemavg` now accept a suffix `_`*`frame`*, where *`frame`* is `spatial` (gives the default operator), `mesh`, `material`, or `geometry` to explicitly specify the frame in which to perform the integration. For example, the

operator `elemavg_mesh(gporder,expr)` evaluates the average of the expression *expr* in the mesh frame.

- For DNN functions, the following functionality has been added:
  - There is now an **Export** button that lets you export the deep neural network to an ONNX (Open Neural Network Exchange) file.
  - More scaling, loss function, and activation function options have been added.
  - The **Settings** window now includes a **Post-training test data** section.
- **Weak Contribution** and **Constraint** equation contributions now allow using a new built-in **Gauss–Lobatto** quadrature method or a user-defined integration point and weight pattern instead of the default Gaussian quadrature. Also, the integration order can now be controlled by a user-defined parameter expression, for example in a Parametric Sweep.
- The new **Integration Rule** feature allows defining custom quadrature rules for use with Weak Contribution and Constraint equation contribution features.
- The new **P-Norm** feature computes the p-norm of an expression over a selection for a chosen value of p. The p-norm provides a differentiable approximation of the maximum value.
- The new **Standard Deviation** feature computes the standard deviation of an expression over a selection.

New Functionality in Studies and Solvers

- The NVIDIA CUDA® direct sparse solver (cuDSS) is an optimized first-generation GPU-accelerated direct sparse solver. This new direct solver can solve some types of problems that require direct solvers significantly faster than traditional CPU-based solvers. It requires installed CUDA driver and runtime versions. The NVIDIA cuDSS direct sparse solver is available for Windows® and Linux®.
- For simulations based on the discontinuous Galerkin method, multiple GPUs are now supported.
- A new **Verlet** explicit time-stepping method is now available as an efficient time-stepping method for the new Solid Mechanics, Explicit Dynamics and Truss, Explicit Dynamics interfaces.
- The **Combine Solution** feature can now concatenate parametric sweep solutions and convert outer solutions from a parametric sweep to inner solutions.
- You can perform eigenmode following with respect to a specific parameter in **Parametric Sweep** and **Auxiliary Sweep** with an **Eigenvalue Parametric** solver. A new

Mode following checkbox has been added to the **Eigenvalue** study step and **Eigenvalue Solver**, and a **Mode Following** attribute feature has been added.

- A new option to add an expression-based scaling has been added to the settings for the **Eigenvalue Solver**. In the **Output** section, you can now choose **Manual** from the **Scaling of eigenvectors** list. An **Expression** field then becomes available.
- A new **Linearize small mesh elements** list in the settings for the **Compile Equations** node can be used to control the geometric shape-function order of small mesh elements used to compile the equations of the current study. This option only affects the current study and modifies the setting for the **Geometry shape function** in the **Curved Mesh Elements** section of the **Settings** window of the **Component** node when the **Avoid inverted elements by curving interior domain elements** option is enabled. The default option is **Physics controlled**. By specifying the **User controlled** option, you can enter a length expression for the maximum element size threshold per component in the table below to prevent mesh elements smaller than this threshold from being curved.
- It is now possible to specify the phase in the **Usage** section with a stateless interface in reduced-order model features. If not specified, it is by default equal to the phase global variable. In the **Parameters** table in results features, the name of the phase variable from a reduced-order model like `rom1` is `rom1.phase`; its value is by default given by another variable, `rom1.input.phase`, which, in turn, is defined by the entry in the new **Phase** field in the **Usage** section.
- A new **Reuse null-space data** checkbox has been added to the **Advanced** attribute node. If selected, when solving a model that requires many iterations, the block structure of the matrix N (the constraint matrix) is reused if possible. This speeds up the solver for certain models.
- All changes to solver settings are now added to the **Changes from Default Settings** tables, including properties that have become inactive.
- Usability improvements have been made for enabling and disabling solver and job configurations.
- The study context menu has been simplified by adding submenus for different study types.
- The MUMPS and PARDISO direct solvers now support reusing the sparsity pattern.
- The MUMPS direct solver has improved support for shared memory parallelism. New **Multithreaded matrix factorization** and **Multithreaded forward and backward solve** options have been added to the settings for MUMPS.

- A new **Stop when no eigenvalues are found** checkbox has been added to the settings for the **Eigenvalue** solver to determine whether to return an empty solution or stop the process with an error.
- Batch parametric sweeps can now use asynchronous queues based on the TaskQueue class.
- MPI is now supported for Mac ARM computers.
- You can now specify multiple plots groups to plot while solving in the **Results While Solving** settings in studies.
- Synchronized rank reduction has been added for the hybrid null function.
- Improved unit handling for the remeshing time-step setting in **Automatic Remeshing** nodes. You can now select a **Remeshing time step type**: **Constant** or **Expression**.
- A **Show Location** option has been added to the **Batch Data** node, **External Processes** node, and **External Processes** window. It shows the location of the model on the file system or in the version history view in the Model Manager database, depending on where the model is stored.
- Cluster and batch sweep support has been added for the **Surrogate Model Training** study node.
- For explicit time stepping, cell time scale expressions can now be evaluated for every mesh element in a given geometric entity level. It is also possible to control the Gauss-point order for the evaluation.
- The **Shift used in the modes computation** setting has been removed from the **LAPACK (filled matrix)** eigenvalue solver because it is no longer useful.

New and Improved Results and Visualization Functionality

GENERAL NEW RESULTS AND VISUALIZATION FUNCTIONALITY

User-Defined Environment Maps

The new **Add Environment Map** option lets you add user-defined environment maps to the list of available environment maps. Define the environment map as a set of colors or use images as input. Two different image types are supported: **Equirectangular**, a panorama image of the environment, and **Cube map**, six images for each face of a cubed mapped environment. There is also support for importing environment-map definitions into the model object to facilitate sharing MPH files with users who do not have the same local environment-map definitions.

More Built-In Environment Maps

In addition to the possibility to add user-defined environment maps, five new built-in environment maps have been added: **Auto Shop**, **Cloudy**, **Dusk**, **Golden Gate Hills**, and **Sky**.

Visualization of the World Space Origin and Axes

You can now visualize the world space origin and the axes of the world coordinate system in the **Graphics** window. Enable **Show origin** and **Show coordinate axes** from the **Graphics** window toolbar.

API Methods for Running All Active Results Features

You can now use the following API methods to run all active plots, evaluation groups, or derived values in a model:

```
model.result().run();
model.result().evaluationGroup().run();
model.result().numerical().run();
```

Color Tables for Safety Plots

The new built-in color tables **Safety** and **SafetyBlue** can be used to visualize safety states in, for example, fasteners in structural mechanics. **Safety** is a discrete color table containing the colors red-orange-green, and **SafetyBlue** is a discrete color table containing the colors red-orange-blue.

More LaTeX Arrow Types

Added support for 25 new arrow types in LaTeX. For example, \circlearrowleft and \circlearrowright can be used to show rotation direction.

Improvements to Preferred Units

You can now define preferred units in the **Preferences** window to apply them universally across models. Support for defining a custom unit for a physical quantity has also been added. Additionally, in the **Preferred Units** node under **Configurations**, you can now save and load preferred units to or from a text file.

More Input Formats in the Color Table Settings

It is now possible to use both decimal and hexadecimal values to specify RGB colors in color tables. For a decimal input, you can define colors as RBG values with a range from 0 to 1 or 0 to 255. For a hexadecimal input, you can define colors using hexadecimal color codes.

Information Section

The **Information** section is shown in more results features. You can now get information about, for example, calculation time and memory usage in 2D and 3D plot groups, derived values nodes, and evaluation group nodes.

Paste Clipboard Data into Result Features

It is now possible to paste any numerical data you have copied to the clipboard — from, for example, a plot, spreadsheet, or document — into 1D and polar plot groups to create a **Table Graph** plot, or directly into **Table** nodes.

Lagrange Order and Gauss Point Order

More results features now support Lagrange order 0 and Gauss point order 0, including **Maximum** and **Minimum** datasets and **Maximum** and **Minimum** numerical evaluation features, which previously only had support for values 1 and above.

Improved Headers in Plot Export Features

When exporting a plot as a text file, the data headers for each column now also display the parameter values. To disable this feature, clear the **Include parameter values in data header** checkbox in the **Advanced** section of a **Plot** export node.

Support for Exporting 16-Bit per Channel PNG Files

In the **Image** and **Animation** export features, you can now export PNG images with higher color precision by selecting the new **16-bit color channels** checkbox. You can also generate 16-bit PNG files using the **Image Snapshot** functionality in the **Graphics** window toolbar.

NEW AND IMPROVED DATASET AND EVALUATION FUNCTIONALITY

- Two new datasets have been added for use in results evaluation of granular flow: **Grain** and **Grain Bin**. These datasets require the Granular Flow Module.
- New **Standard Deviation** nodes are available under **Derived Values** and **Evaluation Group** nodes. You can use them to calculate standard deviation quantities for each solution in a dataset.
- For **System Matrix** evaluations, it is now possible to set a **Cutoff Threshold** at which matrix entries with roundoff errors are considered 0.
- The advanced setting **Define variables** in datasets has been removed. The variables are now always defined.
- In **Global Matrix Evaluation** and **Point Matrix Evaluation** features, it is now possible to make the result symmetric by selecting the new **Make symmetric** checkbox.
- You can now enter a variable in the **Number of sectors** field in **Sector** datasets.
- The **Use derivatives** setting is now also available for **Isosurface** and **Contour** datasets.

- The **Cut Line** and **Cut Plane** datasets now define an index variable that evaluates to the line or plane index. This lets you access each individual plane when additional parallel lines or planes are defined.
- You can now evaluate multiple curves and multiple surfaces separately in integration, average, maximum, minimum, and measurement numerical evaluation features. This is applicable when the underlying dataset is **Cut Line** or **Cut Plane** and additional parallel lines or parallel planes are defined. It is also applicable when the underlying dataset is **Shell** or **Layered Material** and several through-thickness locations are defined.

NEW AND IMPROVED GRAPHICS AND PLOT FUNCTIONALITY

- Two new plot types have been added for plotting granular flow: **Grain** (1D) and **Grain Positions** (2D and 3D). These plots require the Granular Flow Module.
- A new **Table Arrow** plot is available for plotting arrows based on data taken from a table or evaluation group.
- The plot **Quality** settings can now be specified centrally in the plot group. A new **Evaluation settings** option in plots defines whether the **Quality** settings are taken from the parent plot group or are specified in the selected plot.
- It is now possible to vary the transparency in plots by using an expression as input in a **Transparency** node.
- A new **Solution Array** attribute node makes it possible to efficiently create plot arrays from a selection of multiple times, eigenvalues, or parameter values.
- You can now use color tables and gradients to assign colors to the curves in graph plots.
- Multiple improvements have been made to the **Streamline**, **Streamline Multislice**, and **Streamline Surface** plots; see details below:
  - To enhance the look of streamlines that originate from a boundary or an edge in **Streamline** (2D and 3D) and **Streamline Surface** (3D) plots, it is now possible to make the density of the streamlines magnitude-dependent.
  - When the positioning is **Uniform density** or **Magnitude controlled**, the **Separating distance**, **Minimum distance**, and **Maximum distance** settings have been changed to **Density level**, **Minimum density level**, and **Maximum density level**, respectively.
  - General improvements of the integration order in streamline plots. For example, all types of streamlines can now be generated with a higher-order integrator.
  - The default values for the **Maximum number of integration steps** settings have been increased. The property is renamed from `maxsteps` to `maxintegrationsteps`.

- The **Use parameter indicator for solution and phase** checkbox is now available when the **Title type** in plots is **Label**. Previously, it was only available for **Custom** titles.
- Legends in graph plots, *x*- and *y*-axis labels in plot groups, and parameter indicators in plot group titles now support the use of mixed-mode LaTeX commands.
- Color legend titles in plots now support the use of HTML tags; enclose text in the title with, for example, `<b> <b>` to render it in a bold font.
- You can now control the color of node points in 2D and 3D **Mesh** plots by selecting a color from the **Point color** list in the **Coloring and Style** section.
- Previously, annotations were automatically taken from the table headers in **Table Annotation** plots; it is now possible to choose where to take the labels from: **Row headers**, **Row numbers**, **Label** (the plot's label), or **Manual** (to specify your own annotation).
- You can now plot the normalized arc length on the *x*-axis in **Line Graph** plots by selecting the **Normalize** checkbox in the **x-Axis Data** section.
- A **Normalization** setting has been added to the **Polarization** plot. The polarization ellipse size can now be normalized either to each diffraction order or to the largest diffraction efficiency. There is also a **Manual** option that allows you to enter a value in the **Scale** field.
- An option for 1/12 octave bands is now available in **Octave Band** plots.
- In the **Plot Array** settings for plot groups, the **Enable** checkbox and **Array shape** list have been replaced with an **Array type** list, from which you can choose **None**, **Line**, or **Square**.
- The frame and background are now optional in graph legends in 1D plot groups. Clear the **Background** checkbox to remove the frame and background, if desired.
- A **From theme** option has been added to the **Point color** setting in **Function** plots in 1D plot groups.
- Smoothing settings can now be applied to **Arrow Line** and **Arrow Surface** plots when the **Placement** of the arrows is set to **Mesh vertices**.
- In **Graph Marker** subnodes, when displaying **Max and Min** markers, there is now a **Use cubic interpolation** checkbox, which, if selected, can provide a more accurate estimation of the extreme values.
- A new functionality in **Selection** subnodes lets you map evaluations on boundaries to their adjacent domains. This functionality can help produce more accurate plots when the adjacent domains have strongly contrasting material properties, for example. When the **Geometric entity level** is **Domain**, choose **Selected domains** from

the **Evaluate in** list to enable the functionality. The **Adjacent boundary** option (selected by default) corresponds to the behavior in version 6.3 and earlier. It is also possible to control whether boundaries external or internal to the selection are included in the plot by selecting the **Exterior boundaries** and **Interior boundaries** checkboxes, respectively.

NEW AND IMPROVED REPORT AND PRESENTATION FUNCTIONALITY

- The **Model Contents** ribbon and context menus have been reorganized to make it easier to set up reports and presentations.
- Reporting on mesh sequences has been improved.

General Backward Compatibility Considerations

COMSOL Multiphysics version 6.4 can open MPH files saved from COMSOL Multiphysics versions 4.0–6.3.

COMSOL Multiphysics version 6.4 can run Java files saved from COMSOL Multiphysics versions 4.0–6.3. However, the Java files may need to be modified in accordance with information in this document and may need to be recompiled with the `comsol compile` command in version 6.4.

Backward Compatibility with Version 6.3

AWE SOLVER

The `minintactive` property has been removed in version 6.4 It has been replaced by the `minintmethod` by following this mapping:

```
minintactive=off => minintmethod="auto"
minintactive=on => minintmethod="manual"
```

FINALIZE

For `Finalize`, the new property `fastpairdetection` is `on` by default, which can lead to other pairs being generated. Turn off the property to get the same behavior as in version 6.3.

REMOVED -IPV6 FLAG

The `-ipv6` flag has been removed. COMSOL® software now supports IPv6 by default.

Backward Compatibility with Version 6.2

REMOVED DEVICE FEATURE FROM THE PHYSICS BUILDER

In version 6.2 and earlier versions, under the **Device Model Feature** node, a node called a **Device Feature** that created an instance of the **Device Model Feature** directly was available. This functionality was inconsistent with the basic framework, both how **Device Model Feature** nodes should be used and how **Device Model** features generally behave. There is a migration that converts the configuration to an equivalent configuration with a **Device Model** and a **Device** node under a **Global Feature.** So the migration changes the type of the **Device Model Feature** to a global feature, move all device-related nodes to the new **Device Model** node, and finally change the obsolete **Device Feature** node to a normal **Device** node. There are some complicated cases where the migration fails to do a completely valid migration. It then adds a warning giving more information about the problem.

CHANGED NAMES FOR PARAMETERIZED CURVE AND PARAMETERIZED SURFACE DATASETS

The datasets called **Parameterized Curve 2D**, **Parameterized Curve 3D**, and **Parameterized Surface** in earlier versions of COMSOL Multiphysics are now called **Parametric Curve 2D**, **Parametric Curve 3D**, and **Parametric Surface**, respectively.

GEOMETRY IMPORT IN WORK PLANES

For **Import**, there is a new setting, **Include result form virtual operations,** when importing MPHBIN and MPHTXT in a work plane. In 6.2, this setting did not exist in this case, which meant that only the real part of the geometry was imported. In 6.3, the setting is on by default, which means that the whole virtual geometry is imported. This concerns the property `includevirtual` in the COMSOL API *for* use with Java.

GEOMETRIC MODEL UPDATE FOR MESH

The method `geometricModel(boolean)` has been replaced with `geometricModel(String)` to specify if a meshing sequence operates on a geometry or on a mesh.

Backward Compatibility with Version 6.1

NEW WEBVIEW2 BROWSER

The CefSharp web browser control used by the .NET client in version 6.1 has been replaced with WebView2.

NEW CEFSHARP BROWSER

The Chromium™-based CefSharp browser component replaces the Internet Explorer-based .NET WebBrowser component in version 6.1.

In the **Video** and **Web Page** form objects, there is now a **Native browser on Windows** option in the settings to determine if the Internet Explorer®-based .NET WebBrowser component from earlier versions should be used instead. The default value for the native browser is **Chromium™ (CEF)** but all existing models have the setting set to **Internet Explorer®** by default to maintain the functionality from earlier versions. Use the launcher flag `cs.legacywebbrowser=true` as a way to keep the previous browser component instead of CefSharp.

The CefSharp browser component stores user data such as cookies, browser history, and local storage in a folder. This `cscefcache` folder is placed among the temporary files of the client and persist between COMSOL and browser sessions. You can delete the `cscefcache` folder to clear this data. The CEF log output is written to `Users\[user]\.comsol\6.2\logs\cef.log`. The log level can be controlled using the `cs.cefloglevel` launcher argument, which accepts the values `info`, `warning`, `error`, and `fatal`.

EQUATION FORM FOR PDE AND ODE INTERFACES

When adding a PDE or ODE interface in the user interface, the equation form will by default be set to **Study controlled**. This will lead to a different interpretation of time derivatives in frequency-domain studies compared to previous versions. When creating a new PDE or ODE interface from the API or opening a model saved in a previous version, the equation form will be set to **Time domain**, which corresponds to the behavior in previous versions.

THE FLASH MOVIE FORMAT

For movie export, the Flash format is no longer available as of version 6.1.

TIME-EXPLICIT SOLVER

The **Time-Dependent Solver** node settings now includes the time-explicit methods from the **Time-Explicit Solver** node that was available in the **Model Builder** in earlier versions. The **Time-Explicit Solver** node is still available in models created in earlier version of COMSOL Multiphysics. It is also available in the COMSOL API *for* use with Java.

MODEL REDUCTION STUDY STEP

The **Study step for eigenmodes** and **Study step for constraint modes** settings in version 6.1 correspond to **Defined by study step** in previous versions because it is not possible to use one study step for two different solution types.

PERIODIC CONDITIONS — DESTINATION SELECTION

The **Destination Selection** subnode under **Periodic Condition** nodes has been removed in version 6.1 and has been replaced by an optional **Destination Selection** section in the **Settings** windows for **Periodic Condition** nodes. It can still be added from the API for backward compatibility. The **Destination Selection** subnode was used for two purposes:

- To provide a manual destination selection in cases where the automatic detection failed for some reason (for example, in models with curved boundaries). The selection of the feature was used as destination domains. When the **All boundaries** option was selected for the selection list, the automatic destination was used from the parent periodic condition.
- To add a manual orientation for vector transforms of the destination boundaries. The **Orientation of Destination** section was used for this.

For more information about the API compatibility, see COMSOL 6.1 API Changes.

FLASH MOVIE FORMAT

The Flash movie format is no longer available for export of movie files using the **Animation** feature. Animation features in models created using version 6.0 or earlier where the format is set to Flash now use GIF as the format from version 6.1.

STREAMLINE PLOT CHANGES

For 2D **Streamline** and 3D **Streamline Surface** and **Streamline Multislice** plots, the algorithm for finding and plotting streamlines has been improved. The **Density** setting in version 6.0 and earlier versions has been replaced with the **Maximum distance** setting, which is the inverse of the **Density**. A corresponding **Minimum distance** setting has also been added. When opening models from version 6.0 and earlier versions, the **Minimum distance** is assigned a value that is 0.05 times the **Maximum distance**. This is a reasonable heuristic, but it is not guaranteed to give a good result in all cases.

PHYSICS SYMBOLS IN 1D AND 0D GEOMETRIES

The **Physics Symbols** section in the settings for physics interfaces, features, and boundary conditions has been removed for all components with dimension less than 2D. This change might cause issues with old COMSOL API *for* use with Java code

that accesses any physics symbol properties in 1D or 0D, even though they did not have any effect in earlier versions.

New and Updated Models, Apps, and Add-Ins in Version 6.4

EIGENVALUE ANALYSIS OF A TURBOCHARGER ROTOR

In this tutorial model, an eigenvalue analysis of a turbocharger rotor is performed. The rotor is analyzed in an unconstrained configuration, also commonly known as *free–free conditions*. A rotor like this is usually intended for high-speed applications, but here the analysis is performed at standstill. The mesh is imported using a file on Nastran® format.

VELOCITY FORM OF THE PID CONTROLLER

You can now use the velocity form of the PID controller algorithm in addition to the previous position form when creating a PID controller using the **PID Controller Setup** add-in. The velocity form works best with a fixed time step.

Material Library

New and Updated Material Data in Version 6.4

In version 6.4, the Material Library contains 18,806 materials and 181,117 material property datasets. The following improvements and additions have been made:

- New materials include acids, fluorides, and rare earth alloys.
- New references for the thermal diffusivity of Inconel 718 and 718SF were added. The values near room temperature have been increased by 50% (from 0.012 to 0.027 cm^2/s). At higher temperatures, the changes were a few percent.
- The heat capacity heat for liquid He was too low by a factor of 16. That value has been corrected.
- Mg – 4 Li (wt %) has been renamed to Mg – 12.7 Li (at %), and the thermal expansion has been revised.
- The specific heat data for G-10CR and G-11CR were too high by a factor of 10. These datasets and the thermal diffusivity datasets have been corrected.
- The specific heat units for Tefzel 200, 210, and 280 were wrong in the reference. These errors have been fixed.
- The density data for the $CaCl_2$ – $MgCl_2$ system were mislabeled in the reference and have been fixed.
- The thermal conductivity for $C_{12}H_{26}$ (n-dodecane) was too high by a factor of 100. This error has been fixed.
- The reference for the elastic properties of PbTe has been changed; the new values are significantly different.
- The reference for the thermal conductivity and diffusivity for liquid 316 stainless has been changed. The new values are 50% larger.
- The thermal expansion data for FLiNaK, (LiNaK)2CO3, and LiF(NaK)2CO3 used an incorrect temperature scale and have been fixed.
- The composition labels for the n-octane/cyclohexane mixes have been reversed.
- The heat capacity and specific heat data for LiBr should only go down to 293 K, not 100 K. This error has been fixed.
- The reference for the electrical resistivity and conductivity for LaB6 has been changed. Above 200 K, the differences are within 10%; below 200 K they are larger.

- The specific heat data for ErPdIn and YbPdIn were incorrect. The wrong molecule weight was used in the conversion but has now been corrected.
- Several minerals have been combined into one material.
- Thermal diffusivity data have been added for some aluminum alloys.
- The specific heat of 310 stainless has been revised. The reference had a discontinuity near 47 K. This error has been corrected.
- The specific heat of R-449A (a hydrocarbon refrigerant) was incorrect and has been revised.
- New references have been added for the thermal conductivity of Inconel 625; values were changed by approximately 5%.
- The various different phases for TiO2 (rutile, anatase, and brookite) have been combined under the material name TiO2.

AC/DC Module

New and Improved Functionality in Version 6.4

NEW MULTIPHYSICS COUPLING AND NEW MULTIPHYSICS INTERFACES FOR MAGNETOMECHANICS

The new multiphysics coupling **Magnetomechanics, Boundary** connects a boundary feature in the Magnetic Fields interface or the Magnetic Fields, No Currents interface (such as the **Impedance Boundary Condition**, the **Transition Boundary Condition**, or the **Magnetic Shielding** boundary condition) to the mechanical Shell or Membrane interfaces. For each of the four physics combinations, a predefined multiphysics interface is available (as listed below). In addition to this, manually constructed combinations of the multiphysics coupling and the four physics interfaces are supported.

- The **Magnetomechanics, Shell** multiphysics interface couples a boundary feature in the Magnetic Fields interface (such as the **Impedance Boundary Condition**, the **Transition Boundary Condition**, or the **Magnetic Shielding** boundary condition) to a mechanical Shell interface, using the **Magnetomechanics, Boundary** multiphysics coupling.
- The **Magnetomechanics, Membrane** multiphysics interface couples a boundary feature in the Magnetic Fields interface (such as the **Impedance Boundary Condition**, the **Transition Boundary Condition**, or the **Magnetic Shielding** boundary condition) to a mechanical Membrane interface, using the **Magnetomechanics, Boundary** multiphysics coupling.
- The **Magnetomechanics, No Currents, Shell** multiphysics interface couples a boundary feature in the Magnetic Fields, No Currents interface (such as the **Magnetic Shielding** boundary condition) to a mechanical Shell interface, using the **Magnetomechanics, Boundary** multiphysics coupling.
- The **Magnetomechanics, No Currents, Membrane** multiphysics interface couples a boundary feature in the Magnetic Fields, No Currents interface (such as the **Magnetic Shielding** boundary condition) to a mechanical Membrane interface, using the **Magnetomechanics, Boundary** multiphysics coupling.

These multiphysics interfaces require the Structural Mechanics Module, together with the AC/DC Module.

TRANSIENT ANALYSIS SUPPORT FOR THE IMPEDANCE BOUNDARY CONDITION AND THE TRANSITION BOUNDARY CONDITION

The **Impedance Boundary Condition** and the **Transition Boundary Condition** in the Magnetic Fields interface now support the **Time Dependent** study type. The **Transition Boundary Condition** supports **Electrically thick layers**, **Electrically thin layers**, and **Electrically very thin layers**, where "thick" or "thin" refers to the thickness of the layer with respect to the skin depth δ at the typical operating frequency — note that the **Impedance Boundary Condition** is a "very thick layer" by definition. For the very thick layer, the boundary conditions use auxiliary degrees of freedom to provide an approximation of the surface admittance. This approximation is optimized before solving using a *partial fraction fitting* approach. For thin and very thin layers, a precomputed approximation of sufficient order is used, which does not require fitting.

As a result of this addition, thin, highly conductive plates, layers, and coatings can now be modeled in the time domain using the **Transition Boundary Condition**, without having to resolve the skin effect in the material using a boundary layer mesh. This feature can also be used for bulk materials with such a small skin depth that the electromagnetic field hardly penetrates at all, meaning that the interior of the material (typically a metal) does not need to be included in the model: The modeling domain is simply truncated at the material boundary, and the **Impedance Boundary Condition** is used to model the skin effect in the material.

IMPROVED FORCE CALCULATION SUPPORT FOR THE MAGNETIC FIELDS, NO CURRENTS INTERFACE

The **Force Calculation** domain feature in the Magnetic Fields, No Currents interface now supports different methods for force calculation (with **Automatic** being the new default). The **Automatic** setting uses the **Virtual gap method**, combined with boundary layer mesh suggestions. This method makes it possible to compute the clamping forces on a magnet when it is attached to an identical magnet, for example — a use case where the old method would return zero, since it does not take into account the fact that air would fill the gap when the two magnets are pulled apart. The new method typically enables a more accurate local field evaluation, improving most force computations. *Note that the approach used in earlier versions is still available.*

Furthermore, a new **Boundary Force Calculation** feature is available in the Magnetic Fields, No Currents interface. It provides functionality that is similar to the regular **Force Calculation** feature but is more suitable for thin structures.

ROTATING MACHINERY, MAGNETIC IMPROVEMENTS

For the **Multiphase Winding** feature in the Rotating Machinery, Magnetic interface (in 2D), the **Automatic three phase** winding layout configuration has been improved to find the "most practical" layout for virtually all slot-pole combinations with dual-layered slots. *Note that you can still modify the layout after it has been generated.*

Furthermore, a new feature, **Exterior Electric Insulation**, has been added to model exterior boundaries that should not allow for an outflow of induced currents (this is typically used for modeling symmetry). The solver defaults have been updated as well to improve both solving speed and robustness. The new solver defaults differentiate between 2D and 3D models.

PERIODIC CONDITION FOR THE MAGNETIC FIELD FORMULATION INTERFACE

A new **Periodic Condition** boundary feature has been added to the Magnetic Field Formulation interface. The feature applies periodic conditions to the *Magnetic Field* (the degree of freedom). Support for twisted periodicity and sector periodicity is included. This addition greatly simplifies the modeling of periodic structures and sector models such as twisted superconducting cables or toroidal helical coils.

BETTER ELEMENTWISE MAPPING FOR THE PERIODIC CONDITION

The **Periodic Condition** in the Magnetic Fields interface, the Magnetic and Electric Fields interface, and the Magnetic Field Formulation interface now uses **Require compatible meshes** as the default elementwise mapping. Since these three physics interfaces use *curl elements* to discretize the degree of freedom, it is important to have matching meshes on the source and destination boundaries of the periodic condition.

The setting **Require compatible meshes** performs an analysis of the source and destination mesh and checks that the mesh nodes match within a certain tolerance. Without it, interpolation errors may occur, reducing both performance and accuracy. The new default setting only applies to new models. Existing models will continue to function as before.

SYMMETRY FACTORS FOR THE TERMINAL FEATURE

In the **Advanced Settings** section of the **Terminal** feature (which is activated through the **Advanced Physics** option in the **Show More Options** dialog), there is now support for a **Terminal area multiplication factor**. This multiplication factor can be used to model the effects of symmetry. When a symmetry condition has been used such that the surface area of the terminal feature selection represents half of the surface area of the full device, use a multiplication factor of two. A connected load or circuit will then perceive

the full device even though the model includes only part of it (higher values can be used for sector symmetry, for instance).

IMPROVED FRAME HANDLING FOR THE ELECTRIC CURRENTS INTERFACE

The **Current Conservation** feature in the Electric Currents interface has been deprecated and replaced with a **Current Conservation in Solids** feature and a **Current Conservation in Fluids** feature, with **Current Conservation in Solids** being the new default feature. The chosen reference frame — Lagrangian or Eulerian — affects how Maxwell's equations should be interpreted (in particular, this applies to material properties and induced electric fields). The two new features provide a more user-friendly implementation that helps to ensure that the appropriate reference frames are used for liquids, gases, and vacuum on one hand, and solid objects on the other.

EXTENDED ELECTROMAGNETIC HEATING SUPPORT FOR THE ELECTRIC CURRENTS IN (LAYERED) SHELLS INTERFACE

The Electric Currents in Shells interface now supports both the **Electromagnetic Heating** multiphysics coupling, as well as the **Electromagnetic Heating, Layered Shell** multiphysics coupling (before, it only supported the layered shell version). This allows for easier modeling of nonlayered shells in the context of *Joule heating*.

ELECTRICAL CIRCUIT IMPROVEMENTS

The dependent sources in the Electrical Circuits interface — the **Voltage-Controlled Voltage Source**, the **Voltage-Controlled Current Source**, the **Current-Controlled Voltage Source**, and the **Current-Controlled Current Source** — have been extended to support multiple current or voltage dependencies. This increases the feature's flexibility and improves compatibility with the SPICE import format. In particular, support has been added for the SPICE "poly" statement in which the current or voltage depends on a polynomial which is based on several currents or voltages elsewhere in the circuit.

USABILITY IMPROVEMENT: NEW CONTEXT MENUS

For all physics interfaces in the AC/DC Module, the feature context menu and the toolbar (or ribbon) have been completely renewed. Features have been organized by type and popularity. Common features like the **Current Conservation in Solids** feature or the **Boundary Terminal** feature are exposed and at the top of the menu, while more exotic features like the **Surface Magnetic Current Density** are at the bottom (and typically hidden in a subcategory).

Features from different physics interfaces with similar properties are organized similarly, and some features are renamed to reduce ambiguity and to better reflect what they do. This allows for a quicker workflow for both simple and advanced models.

MAGNETIC MATERIAL LIBRARY UPDATE

The AC/DC material library has been updated with new magnetic materials from Bomatec®. The materials include NdFeB grades (regular sintered, corrosion stable, corrosion and temperature stable, hot pressed, bonded compressed, injection molded, and extruded), Ferrite (dry pressed, wet pressed, and injection molded), SmCo and SmFeN (injection molded), SmCo5 and Sm2Co17 (sintered), and AlNiCo (cast and sintered).

In addition to this, the entire **Nonlinear Magnetic** branch in the material library has been reprocessed using the updated B-H Curve Checker app to remove nonmonotonic behavior of the differential permeability around the zero point, and to provide better extrapolation into the fully saturated region.

When performing a measurement on a soft magnetic material (extracting a B-H curve) it is not uncommon for the curve to start at a moderate slope around the zero point, before proceeding to a steep slope and then reaching saturation. This initial slope typically does not contribute to the numerical analysis since: 1) completely unsaturated regions contribute little to the overall reluctance of a magnetic circuit such as a laminated core, and 2) most electrical machines are designed to operate reasonably close to saturation (since not doing so would imply having an excess of material, increasing cost). At the same time, the moderate initial slope can have a severe impact on nonlinear (and transient) solver performance. Overall, the newly processed curves give better solver performance while having a negligible impact on the solution.

MINOR USABILITY IMPROVEMENTS

- The **Free Space** feature now issues a warning when its selection overlaps with a material having "nonfree space"-like material properties. This reduces the risk of ignoring material properties due to **Free Space** assuming vacuum conditions.
- The default plots have been updated to allow for easier modification. In previous versions, streamlines used to have a color expression that needed to be kept in sync with the main surface plot feature. The color expression has been removed.

THERMAL ANALYSIS OF A HIGH-POWER IGBT MODULE

This industrial-scale proof-of-concept model demonstrates how to perform an electric–thermal analysis of a high-power insulated-gate bipolar transistor (IGBT) module. The module has a rated voltage of 1200 V and a nominal current of 1800 A. The generated heat is dissipated through a heat sink, and the currents and the temperature profile are evaluated in the semiconductors, the metallization layers, and the bond wires. The model uses the Joule Heating multiphysics interface that combines Electric Currents and Heat Transfer in Solids, together with the Electromagnetic Heating multiphysics coupling.

CELL MEMBRANE ELECTROPORATION

Electroporation is a technique where a localized electric field is used to generate nanometric pores in cell membranes, improving the cell permeability for ions and pharmaceuticals. This model shows the electroporation of a spherical cell by means of a nanosecond electric pulse. The model uses the Electric Current interface, Boundary ODEs and DAEs interface, and multiple Debye dispersive material data in the time domain and in the frequency domain.

INDUCTION HEATING OF A MOVING FERROMAGNETIC MECHANICAL PART

Induction heating is used for various metallurgical processes such as hardening. In this model, the 3D induction heating of a mechanical joint passing through an induction heating coil is simulated. Curie point effects and temperature-dependent resistivity in the iron are taken into account.

THIN CONDUCTIVE LAYER USING THE TRANSITION BOUNDARY CONDITION

In electromagnetic simulations of, for example, transformers and converter stations, one often needs to include geometrically thin conductive layers. Explicitly meshing these thin layers can be computationally expensive and numerically challenging. In COMSOL®, you can approximate conductive layers by using the **Transition Boundary Condition** (TBC) or **Impedance Boundary Condition** (IBC) boundary feature. This model demonstrates that the TBC can produce accurate results in both time-domain and frequency-domain simulations.

NONLINEAR FERROMAGNETIC DIAPHRAGM

A magnetic diaphragm is a flexible, thin structure that interacts with magnetic fields to perform mechanical or sensing functions. When subjected to an external magnetic field, the diaphragm deforms due to magnetomechanical interactions, converting magnetic energy into mechanical displacement or, conversely, mechanical deformation into changes in the magnetic field distribution. This model studies the magnetomechanics of a magnetic diaphragm using the Magnetomechanics, Shell multiphysics interface.

STATIC FIELD MODELING OF A HALBACH ROTOR (MODEL UPDATE)

This example presents the static-field modeling of a flux-focusing magnetic rotor using permanent magnets, a magnetic rotor also known as a *Halbach rotor*. The use of permanent magnets in rotatory devices such as motors, generators, and magnetic gears is increasing due to their no-contact, frictionless operation. This model illustrates how to calculate and change the magnetic field of a Halbach rotor in 3D by changing the number of permanent magnet segments and the number of pole pairs.

B–H CURVE CHECKER (APP UPDATE)

The B-H Curve Checker app has been updated to improve output curve quality both around the zero point and for the saturation region. The app now allows for removing nonmonotonic behavior of the differential permeability around the zero point. In addition to this, it provides better extrapolation options for the fully saturated region.

Backward Compatibility with Version 6.3 and Earlier

- The default **Force name** in the **Force Calculation** feature has been changed from "0" to "1", since this behavior better synchronizes with labels, tags, and other default feature properties. In addition to this — specifically for the **Force Calculation** feature in the Magnetic Fields, No Currents interface — the default force calculation method is now the **Virtual gap method**. This means that this version of the **Force Calculation** feature is, strictly speaking, not compatible with previous versions of COMSOL®, considering the support of model methods, Java code, and stored MATLAB® scripts. This is because these scripts typically do not mention default settings explicitly. To patch up previously recorded code, in most cases it suffices to add instructions to set these values back to their old default:

```
model.physics().create("mfnc", "MagnetostaticsNoCurrents",
"geom1");
model.physics("mfnc").create("fcal1", "ForceCalculation", 3);
```

```
model.physics("mfnc").feature("fcal1").set("ForceCalculationMetho
d", "NoForceOnContact");
model.physics("mfnc").feature("fcal1").set("ForceName", 0);
```

Note that the new default settings are often better than the old ones. If the script works with the new default, it is better not to revert back to the old value.

- The switch to the new **Require compatible meshes** default setting for the **Elementwise mapping** in the **Periodic Condition** for the Magnetic Fields interface, the Magnetic and Electric Fields interface and the Magnetic Field Formulation interface means that this version of the **Periodic Condition** is, strictly speaking, not compatible with previous versions of COMSOL®, considering the support of model methods, Java code, and stored MATLAB® scripts. This is because these scripts typically do not mention the default setting explicitly. To patch up old recordings, in most cases it suffices to add an instruction setting the mapping back to the old default:

  ```
  model.physics().create("mf", "InductionCurrents", "geom1");
  model.physics("mf").create("pc1", "PeriodicCondition", 3);
  model.physics("mf").feature("pc1").set("meshSimilarity", "def");
  ```

  Note that the new default is actually safer than the old one, since it stops solving for unreliable meshes. If the script works with the new default, it is better not to revert back to the old value.

- The switch from the old **Current Conservation** feature to the new default **Current Conservation in Solids** feature in the Electric Currents interface means that this version of the Electric Currents interface is, strictly speaking, not compatible with previous versions of COMSOL®, considering the support of model methods, Java code, and stored MATLAB® scripts. This is because these scripts typically do not mention the creation of the default feature explicitly. To patch up previously recorded code, in most cases it suffices to add a feature-create instruction right after introducing the Electric Currents interface (although migrating to `"CurrentConservationSolid"` or `"CurrentConservationFluid"` is recommended):

  ```
  model.physics().create("ec", "ConductiveMedia", "geom1");
  model.physics("ec").create("cucn1", "CurrentConservation", 3);
  model.physics("ec").feature("cucn1").selection().all();
  ```

  Note that although manual migration is recommended, most scripts will continue to work without it: Unless the model includes large structural displacements, the new default feature will function the same as the old one.

Acoustics Module

New Functionality in Version 6.4

MULTI-GPU SUPPORT FOR PRESSURE ACOUSTICS, TIME EXPLICIT

Distributed computing using multiple GPUs is now supported when solving models using the **Accelerated solver formulation** available with the **Pressure Acoustics, Time Explicit** interface. This is the case both for multiple GPUs on one machine or for GPUs set up in a cluster architecture. The accelerated formulation, when solving on a CPU, is now also available for cluster architectures. This gives important performance gains when solving large transient acoustic models, for example, in room acoustic or car cabin acoustic problems.

IMPORT OF CGNS FLOW DATA WITH IMPORTED FLUID FLOW PHYSICS

On Windows, it is now possible to import and use CFD analysis data in the CGNS (CFD General Notation System) format in a COMSOL Multiphysics model. Importing this data is done by combining the new **Imported Fluid Flow** interface with the new **CFD Data (CGNS)** function. The **Imported Fluid Flow** interface handles the correct and consistent mapping of the data onto the computational mesh. If used in combination with the built-in multiphysics couplings (**Background Fluid Flow Coupling** or **Aeroacoustic Flow Source Coupling**), the imported flow data will also be used in subsequent aeroacoustics models just as if the flow was computed by one of the Fluid Flow interfaces. Data can also simply be imported onto the computational mesh and used, for example, as a pressure load in a subsequent vibroacoustic analysis.

PERIODIC PORT

Periodic Port is a new dedicated feature used to model transmission, reflection, and scattering problems for periodic structures such as absorbers and diffusers. The feature is, in particular, interesting for diffusers, as it can split the reflected energy into specular and nonspecular directions. The periodic port handles plane wave incidence on the structures as well as all reflected and transmitted diffraction orders. The diffraction orders are captured using the **Diffraction Order Port** subfeature to the **Periodic Port** condition. The **Periodic Port** condition is set up together with a **Periodic Condition** with the **Floquet periodicity** (Bloch periodicity) option.

POROACOUSTICS FOR PRESSURE ACOUSTICS, TRANSIENT

The **Poroacoustics** feature is now also available in the **Pressure Acoustics, Transient** interface. It was previously only available for the **Pressure Acoustics, Time Explicit** interface. The feature has the same functionality and options in both interfaces.

JCA AND JCAL POROACOUSTICS MODELS FOR THE POROACOUSTICS FEATURE IN TIME DOMAIN

In the **Poroacoustics** feature, in both the **Pressure Acoustics, Transient** and the **Pressure Acoustics, Time Explicit** interfaces, two new options are available when setting up a porous material. The **Johnson–Champoux–Allard (JCA)** and the **Johnson–Champoux–Allard–Lafarge (JCAL)** models extend the previous **User defined** option. When the new models are selected, the underlying partial fraction approximation is now automatically computed based on the usual poroacoustic material parameters for the two models.

LINEARIZED PERTURBED COMPRESSIBLE EQUATIONS (LPCE) FORMULATION FOR LINEARIZED EULER

In all the linearized Euler physics, two formulations can now be selected. Select the **Linearized Euler formulation** as either the **Full linearized Euler** (the default) or the **Linearized perturbed compressible equations (LPCE)** (the new formulation). The first formulation is valid for all Mach numbers (as long as there is not shock formation), while the second is only valid for low Mach numbers ($\mathrm{Ma} < 0.3$). The second formulation has some additional stability properties, as vorticity waves cannot propagate (they are not supported by the equations in this limit).

IMPROVED STABILIZATION FOR LINEARIZED EULER

The stabilization in the **Linearized Euler, Frequency Domain** and the **Linearized Euler, Transient** interfaces has been improved to a more consistent formulation. In the time domain, a new **Limit small time steps effect on stabilization time scale** option can be selected to remedy issues in models with a very small time scale.

EXTENDED IMPEDANCE CONDITION IN LINEARIZED POTENTIAL FLOW

The so-called Brambley impedance model is now available in **Linearized Potential Frequency Domain**; it offers an extension of the classical Ingard–Myers impedance condition. Where the Ingard–Myers condition assumes an infinitely thin boundary layer of the background flow, the new **Brambley (finite boundary layer)** impedance model option approximates the boundary layer with a finite linear profile.

NEW FLOW MODEL AND WINDOW FUNCTION AVAILABLE IN THE AEROACOUSTICS FLOW SOURCE FEATURE

The **Aeroacoustic Flow Source** now supports the RANS-EVM SST turbulence model with the **Scale-adaptive simulation** option. The model can be selected along with LES and DES as input.

It is also possible to add a **Window Function** to the **Aeroacoustic Flow Source**. When the aeroacoustic flow source region is compact — that is, it is not selected in the entire acoustic domain — it can be necessary to use a window function to truncate the aeroacoustic source terms in a consistent manner. This can often be the case, as it is computationally expensive to resolve the flow source in its full spatial extent.

ACOUSTIC–ACOUSTIC BOUNDARY MULTIPHYSICS COUPLING

The new **Acoustic–Acoustic Boundary** multiphysics coupling can be used to couple any acoustic interfaces where the coupling condition is continuity in pressure: That is, **Pressure Acoustics, Frequency Domain**; **Pressure Acoustics, Transient**; **Pressure Acoustics, Boundary Elements**; and **Thermoviscous Acoustics, SLNS Approximation**. Model configurations coupling two interfaces of the same kind are also possible with this feature.

PROJECTION-BASED QUADRATURE FOR DG TIME-EXPLICIT PHYSICS PAIR CONDITIONS

The time-explicit dG-FEM method uses an efficient quadrature free computation of the boundary flux. In certain cases, this can result in unstable behavior when pair conditions are used with nonconforming meshes. To remedy this, a more accurate projection-based quadrature method is used (per default) on all pair conditions.

UPDATED FEATURES IN RAY ACOUSTICS

The default **Wall** feature in the **Ray Acoustics** interface has been updated to a more user-friendly version when working with acoustic ray-tracing problems. Input for scattering and absorption coefficients are now the default. The old wall feature, with all its advanced modeling options, still exists and has been renamed to **General Wall**.

The **Material Discontinuity** feature is no longer the default in the **Ray Acoustics** interface. A new option for modeling a thin absorbing structure has been added. It is possible to directly enter the transmission coefficient T and the reflection coefficient R. This is useful for modeling, for example, screens or curtains. If a model contains several materials, then manually add the **Material Discontinuity** feature on the interior boundary for a correct physical description.

THE STABILIZED FORMULATION IS NOW DEFAULT FOR PRESSURE ACOUSTICS, BOUNDARY ELEMENT

The **Stabilized formulation** is now the default when modeling with the **Pressure Acoustics, Boundary Element** interface. The stabilized formulation has been improved, including the default solver configuration. This gives important performance improvements. Note that interior conditions are not available for the stabilized formulation; if needed, disable the formulation.

MODE FOLLOWING

When solving mode analysis problems to, for example, compute a dispersion diagram, a new **Mode following** option is available in the **Filtering and Sorting** section of the study step. This option is useful for a large range of acoustic problems.

EVANESCENT MODES ARE NOW AUTOMATICALLY INCLUDED IN ALL PORT FEATURES

All the port conditions in acoustics (available in **Solid Mechanics**; **Pressure Acoustics, Frequency Domain**; **Thermoviscous Acoustics, Frequency Domain**; and **Linearized Potential Flow, Frequency Domain**) now automatically include and support evanescent as well as propagating modes. This in general gives a better representation of the acoustic field at the port near cuton/cutoff frequencies and when the port is located close to sources or boundaries.

New and Updated Models, Apps, and Add-ins in Version 6.4

POINT SOURCE IN A 2D JET: RADIATION AND REFRACTION OF SOUND WAVES THROUGH A 2D SHEAR LAYER

This is a new benchmark model for the linearized Euler physics interfaces. The model results are compared to data from a NASA benchmark.

HELMHOLTZ RESONATOR WITH FLOW: IMPORTED FLUID FLOW FROM CGNS DATA

This tutorial shows how to import CGNS flow data into a model using the **Imported Fluid Flow** interface and the **CFD Data (CGNS)** function. The model couples the flow to a **Linearized Navier–Stokes, Frequency Domain** interface with the built-in multiphysics couplings.

GENERIC NACELLE WITH AN ACOUSTIC LINER

This new tutorial model demonstrates the use of the new **Brambley (finite boundary layer)** impedance option available in the **Linearized Potential Flow, Frequency Domain** interface. The model compares the results with the detailed **Linearized Navier–Stokes, Frequency Domain** interface.

POROUS MATERIAL CHARACTERIZATION FROM IMPEDANCE TUBE MEASUREMENT

This new application is used for retrieving the poroelastic material parameters from impedance tube measurements performed on a sample. The process is based on a parameter estimation procedure.

Backward Compatibility with Version 6.3 and Earlier

- The **Velocity** boundary condition in Thermoviscous Acoustics is no longer available on interior boundaries. A feature added in Java code will therefore not be active on such a boundary. Consider replacing it with the **Interior Velocity** feature or replace it with a pointwise constraint to retrieve the old behavior.
- The Java code to add the **Acoustic BEM–FEM Boundary** will now add the **Acoustic–Acoustic Boundary** instead. These two features perform the same coupling, but the **Acoustic–Acoustic Boundary** supports more interfaces, meaning that the default selection of coupled interfaces can change in models with many interfaces. To retrieve the old behavior, replace `"AcousticAcousticBoundary"` with `"AcousticBEMFEMBoundary"` when creating the coupling.
- The stabilized formulation is now the default in the **Pressure Acoustics, Boundary Elements** interface. To retrieve the old behavior when running a Java file add the command:

  ```
  model.component("comp1").physics("pabe").prop("Stabilization")
  .set("StabilizedFormulation", false);
  ```

- The consistent stabilization in Linearized Euler has been improved. The stabilization constant therefore has a new default value of 1. Consider using the new default value when running models where another value is used.
- The default features in **Ray Acoustics** have been updated. To retrieve the old behavior for exterior boundaries, change all references from `"wall1"` to `"gwall1"` and add the following lines after creating the interface:

  In 3D:

  ```
  model.component("comp1").selection().create("extsel","Explicit");
  ```

```
model.component("comp1").selection("extsel")
.geom("geom1", 3, 2, new String[]{"exterior"});
model.component("comp1").selection("extsel").all();
model.component("comp1").physics("rac")
.create("gwall1","GeneralWall", 2);
model.component("comp1").physics("rac").feature("gwall1")
.selection().named("extsel");
```

In 2D:

```
model.component("comp1").selection()
.create("extsel", "Explicit");
model.component("comp1").selection("extsel")
.geom("geom1", 2, 1, new String[]{"exterior"});
model.component("comp1").selection("extsel").all();
model.component("comp1").physics("rac")
.create("gwall1", "GeneralWall", 1);
model.component("comp1").physics("rac").feature("gwall1")
.selection().named("extsel");
```

To retrieve the old behavior for interior boundaries, add the following after creating the interface:

In 3D:

```
model.component("comp1").selection().create("intsel","Explicit");
model.component("comp1").selection("intsel")
.geom("geom1", 3, 2, new String[]{"interior"});
model.component("comp1").selection("intsel").all();
model.component("comp1").physics("rac")
.create("matd1","MaterialDiscontinuity", 2);
model.component("comp1").physics("rac").feature("matd1")
.selection().named("intsel");
```

In 2D:

```
model.component("comp1").selection().create("intsel","Explicit");
model.component("comp1").selection("intsel")
.geom("geom1", 2, 1, new String[]{"interior"});
model.component("comp1").selection("intsel").all();
model.component("comp1").physics("rac")
.create("matd1","MaterialDiscontinuity", 1);
model.component("comp1").physics("rac").feature("matd1")
.selection().named("intsel");
```

- In the **General Wall** feature of **Ray Acoustics**, when the wall condition is set to **Mixed diffuse and Specular** reflection, and the **Primary ray condition** is either **Probability** or **Expression**, the **Otherwise** list has been updated to support nonreflected conditions. Therefore, if you have the following lines in the API code:

```
model.component("comp1").physics("gop").feature("wall1")
.set("WallCondition","MixedDiffuseSpecular")
```

```
model.component("comp1").physics("gop").feature("wall1")
.set("Otherwise","Disappear")
```

To achieve full backward compatibility, change `"Otherwise"` to `"OtherwiseForMixed"` and set the value to one of `"Disappear"`, `"Freeze"`, `"Stick"`, and `"Pass"`.

Battery Design Module

New Functionality in Version 6.4

AQUEOUS ELECTROLYTE TRANSPORT INTERFACE

A new Aqueous Electrolyte Transport interface is now available. The interface computes the potential and species concentration fields in a dilute aqueous electrolyte. The new interface is specialized for modeling aqueous electrolytes featuring weak acids, weak bases, ampholytes, and generic complex species and can, for instance, be used for mechanistic corrosion modeling, electrochemical models of biological systems, and electrochemical sensor modeling.

Due to its more efficient handling of equation reactions and easier model setup, the new interface may be more preferable to use in some cases than the more generic Tertiary Current Distribution, Nernst–Planck interface. The transport is defined by the Nernst–Planck equations, including diffusion, migration, and convection, in combination with electroneutrality and the self-ionization equilibrium reaction of water (autoprotolysis).

POWER LOSS VARIABLES

New power loss variables have been introduced in the Electrochemistry interfaces. By using the new variables, it is possible to evaluate the magnitude of the total power losses in a battery cell and compare the losses of the individual components (such as the separator, electrode, and current conductor). By integrating the power losses over time, the new variables may also be used to compute the round-trip energy efficiency of a battery cell when subject to a charge–discharge load cycle.

The power losses are defined by considering the losses in the Gibbs free energy of all reacting and transported species, which allows for differentiation between ohmic, concentration, and activation losses. In battery interfaces that support particle intercalation, separate intercalation transport loss variables are also defined.

The variables are available both locally on domains and boundaries, as integrated variables for the whole cell, or integrated per individual model-tree feature node.

The power loss variables can be accessed during results visualization under **Power losses** (in the relevant Electrochemistry interface submenu) when clicking either **Insert expression** or **Replace expression**.

LOAD CYCLE NODE

A new **Load Cycle** node has been added to the Primary Current Distribution, Secondary Current Distribution, Tertiary Current Distribution and the battery interfaces. The node may be used to define arbitrary charge–discharge load cycles by adding **Voltage**, **Power**, **Current**, **C Rate**, and **Rest** child nodes, which are executed in sequence.

For each node in the load cycle sequence, the user may define one or multiple dynamic continuation or break (switching) criteria, which may be based on time, voltage, or current limits, as well as user-defined conditions using arbitrary variable expressions. By using the **Subloop** child node, dynamic switching between different cycling schemes is also possible. For instance, you can insert one or more reference performance tests for certain criteria within a longer cycling scheme.

The **Load Cycle** node also allows for automatic definitions of current and voltage probes, as well as solver stop conditions.

The **Load Cycle** node is available both as a boundary node for porous electrode and current conductor domains and as a child node of the **Electrode Surface**, **Highly Conductive Porous Electrode**, **Thin Porous Electrode**, **Perforated Electrode Surface**, **Sacrificial Anode Surface**, and **Thin Electrode Surface** nodes.

INITIAL VALUES FOR THE ION-EXCHANGE MEMBRANE NODE IN THE TERTIARY CURRENT DISTRIBUTION, NERNST–PLANCK INTERFACE

The **Ion-Exchange Membrane** node in the Tertiary Current Distribution, Nernst–Planck interface has a new **Add Donnan shift to initial values** option, which is enabled by default when creating a new model. The new option automatically shifts the initial concentration and potential values specified by the user in the **Initial Values** node for the active **Ion-Exchange Membrane** domain, assuming that the user-defined values represent the values for a bulk liquid electrolyte in equilibrium with the membrane. The shifted initial values are then used as initial values for the solver.

The new option facilitates easier model setup since it typically excludes the need to sweep the fixed space charge of the membrane to a desired nonzero value using an additional study step.

The new option is enabled by default in new models, which means that Java API backward compatibility may be affected by the change.

DEFAULT SOLVER UPDATES

Reaction forces are no longer stored by default by the time-dependent solver. This improves performance for 1D models storing all solutions.

The recovery damping factor in the default solver for stationary study steps in most Electrochemistry interfaces has been reduced to 0.1 (from 0.75). This may improve convergence in certain cases.

DEFAULT PLOTS

The default plots have been changed for improved display of results.

New and Updated Models, Apps, and Add-ins in Version 6.4

The following new tutorials has been added to the Battery Design Module Application Library:

- Liquid-Cooled Prismatic Battery Pack
- Open Circuit Voltage and Differential Voltage Modeling of a Lithium-Ion Battery
- Voltage Hysteresis in a Lithium Iron Phosphate (LFP) Electrode
- Minimizing the Charging Time of a Lithium-Ion Battery
- Power Losses in a Lithium-Ion Battery

The following tutorials where revised and updated to make use of the new **Load Cycle** node: li_battery_1d, lib_base_model_1d, lib_diffusion_induced_stress, lib_rate_capability, lib_drive_cycle, lib_single_particle, li_sigr_hysteresis, li_battery_multiple_materials_1d, li_plating_with_deformation, zn_ago_battery_1d, lithium_sulfur, li_battery_thermal_2d_axi, li_battery_1d_for_thermal_models, li_battery_thermal_2d_axi, li_battery_thermal_3d and li_battery_pack_3d

The Vanadium Redox Flow Battery model has been updated to use the new default **Add Donnan shift to initial values** checkbox.

The Cooling of a Prismatic Battery tutorial model has been updated to use the new **Flat-sided oval (prismatic)** layer configuration option in the **Battery Layers** node.

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

New Functionality in Version 6.4

- 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® version 6.4 includes an upgraded version of the Parasolid kernel. As a result, a number of stability issues have been fixed, which makes geometry modeling operations and the import of CAD models more robust.
- The CAD file import and export functionality included with these products has been extended to support new versions for some of the supported file formats (for details, see under Read from File, CAD on www.comsol.com/products/specifications/cad/).
- When importing CAD assemblies saved in the file formats of the CATIA® V5, Inventor®, NX®, PTC Creo Parametric™, PTC Pro/ENGINEER®, and SOLIDWORKS® software, it is now possible to display a 3D preview of the assembly components in the Graphics window. You can use this preview to choose the components to import from the assembly.
- The **Import** feature for CAD files has been updated with the new **Fix errors** list with the choices **Automatic**, **On**, and **Off** for detecting and repairing geometric errors during importing. The following checkboxes have also been added for simplifying and repairing: **Delete small details**, **Heal edges**, **Minimize tolerances**.
- The settings for the **Repair** feature now include the following new checkboxes for simplifying and repairing objects: **Fix errors**, **Delete small details**, **Heal edges**, **Minimize tolerances**.

New Functionality in the Design Module in Version 6.4

The Part Library now offers female and male human-like manikin geometry parts with parameterized joint angles and predefined sitting, standing, and running positions. The parts also allow for scalable body height while preserving the basic proportions.

New Functionality in LiveLink™ for AutoCAD® in Version 6.4

- The LiveLink™ interface now supports AutoCAD® 2026.
- The **LiveLink for AutoCAD** feature has been updated with the new **Fix errors** list with the choices **Automatic**, **On**, and **Off**, as well as the **Delete small details** checkbox for repairing and simplifying the synchronized geometry.

New Functionality in LiveLink™ for Inventor® in Version 6.4

- The LiveLink™ interface now supports Inventor® 2026.
- The **LiveLink for Inventor** feature has been updated with the new **Fix errors** list with the choices **Automatic**, **On**, and **Off**, as well as the **Delete small details** checkbox for repairing and simplifying the synchronized geometry.

New Functionality in LiveLink™ for PTC Creo Parametric™ in Version 6.4

- The LiveLink™ interface now supports PTC Creo Parametric™ 12.
- The **LiveLink for PTC Creo Parametric** feature has been updated with the new **Fix errors** list with the choices **Automatic**, **On**, and **Off**, as well as the **Delete small details** checkbox for repairing and simplifying the synchronized geometry.

New Functionality in LiveLink™ for Revit® in Version 6.4

- The LiveLink™ interface now supports Revit® 2026.
- The **LiveLink for Revit** feature has been updated with the new **Fix errors** list with the choices **Automatic**, **On**, and **Off**, as well as the **Delete small details** checkbox for repairing and simplifying the synchronized geometry.

New Functionality in LiveLink™ for Solid Edge® in Version 6.4

The **LiveLink for Solid Edge** feature has been updated with the new **Fix errors** list with the choices **Automatic**, **On**, and **Off**, as well as the **Delete small details** checkbox for repairing and simplifying the synchronized geometry.

New Functionality in LiveLink™ for SOLIDWORKS® in Version 6.4

The **LiveLink for SOLIDWORKS** feature has been updated with the new **Fix errors** list with the choices **Automatic**, **On**, and **Off**, as well as the **Delete small details** checkbox for repairing and simplifying the synchronized geometry.

CFD Module

New Functionality in Version 6.4

ROTATING FRAME PROPERTY

A **Rotating frame** property has been added to the **Physical Model** section of the settings for the Creeping Flow, Laminar Flow, Turbulent Flow, Large Eddy Simulation, Detached Eddy Simulation, and Brinkman Equations interfaces. When selected, the velocity field is defined relative to a rotating frame by including the fictitious Coriolis, centrifugal, and Euler forces in the momentum equation and a global **Rotating Frame** feature, with settings for the **Axis of rotation**, **Rotation speed**, and **Rotational direction**, is automatically added to the physics interface.

ELLIPTIC BLENDING REYNOLDS STRESS MODEL

The new turbulence model **Elliptic Blending R-ε** has been added under **Turbulence model** when the **Turbulence model type** is chosen as **RANS-RSM**. Model Wizard entries have been added for the Single-Phase Flow, Nonisothermal Flow, and High Mach Number Flow interfaces. This Reynolds stress model applies blending of near-wall and far-field expressions for the pressure–strain correlation and the turbulence dissipation rate to generate more accurate results for the Reynolds stresses in the near-wall region.

NEW REYNOLDS-STRESS DIFFUSION MODELS FOR RANS-RSM

The Reynolds-stress diffusion model in the RANS-RSM interfaces has four predefined options: **Daly–Harlow**, **Hanjalic–Launder**, **Mellor–Herring**, and **Lumley**.

SCALE-ADAPTIVE TURBULENT FLOW SIMULATIONS

The Turbulent Flow, SST interface has a new option to **Include scale-adaptive simulations**. By including the von Kármán length scale, a broader range of turbulent scales can be resolved in unsteady RANS simulations.

ROTATION–CURVATURE CORRECTION FOR EDDY VISCOSITY MODELS

The two- and three-equation **RANS-EVM** models have a new option to **Include rotation–curvature correction**. Streamline-curvature and system-rotation effects are taken into account by modifying the production rates for the turbulence kinetic energy, and the turbulence dissipation rate or the specific turbulence dissipation rate.

ALGEBRAIC TURBULENCE MODELS IN ROTATING MACHINERY, HIGH MACH NUMBER FLOW INTERFACES

Two new Turbulent Flow interfaces, High Mach Number Flow, Algebraic yPlus and High Mach Number Flow, L-VEL, have been added under Rotating Machinery, High Mach Number Flow branch in the Model Wizard. Solutions obtained with algebraic turbulence models can often provide good initial values for simulations with more complex turbulence models.

ANISOTROPIC THERMAL TURBULENCE MODELS

In the Nonisothermal Flow and Reacting Flow multiphysics couplings as well as in the High Mach Number Flow interfaces, it is now possible to use anisotropic thermal turbulence models. The Generalized Gradient-Diffusion Hypothesis (GGDH) model and its high-order extension (HO-GGDH) are now available to model the anisotropy of the turbulence with respect to the heat equation. These models rely on the Reynolds stress tensor and therefore require a RANS-RSM model to account for turbulence anisotropy in the momentum equation.

NONISOTHERMAL REACTING FLOW LARGE EDDY SIMULATIONS

Three Large Eddy Simulation interfaces, LES RBVM, LES RBVMWV, and LES Smagorinsky, have been added to the Model Wizard in the Chemical Species Transport branch under Nonisothermal Reacting Flow. These physics interfaces simulate fluid flow with mass and heat transfer in the turbulent regime and apply large eddy simulation (LES) modeling for the unresolved scales of all three physics. In addition to a license for the CFD Module, these interfaces require a license for either the Chemical Reaction Engineering Module, the Battery Design Module, or the Fuel Cell & Electrolyzer Module.

FLUID–STRUCTURE INTERACTION COUPLING FOR BOUNDARIES BETWEEN FREE AND POROUS DOMAINS

The Fluid–Structure Interaction multiphysics coupling node is now also applicable on boundaries between free-flow domains and porous-media domains. This enables studies of the interaction between free-flow and poroelastic domains.

MARANGONI EFFECT FOR CONCENTRATION GRADIENTS

In the **Free Surface**, **Stationary Free Surface**, and **Fluid-Fluid Interface** features, the **Surface tension coefficient** list has the new option **Library coefficient, liquid mixture/gas interface**, with a selection of predefined binary liquid mixtures. When a Transport of Concentrated Species or Transport of Diluted Species interface is active in the component, the appropriate interface selection can alternatively be selected to apply

the surface tension values of each corresponding species, which applies the mass fraction or concentration from the corresponding species transport interface.

PREDEFINED LOSS OPTIONS IN THE GRILLE BOUNDARY CONDITION

The **Flow condition** setting in the **Grille** boundary condition has the new option **Grille type**, with predefined models for **Wire gauze**, **Square mesh**, and **Perforated plate**.

PRESSURE JUMP IN THE FREE AND POROUS MEDIA FLOW COUPLING

The **Free and Porous Media Flow Coupling** multiphysics coupling node has a new option to **Include pressure jump across free–porous boundary**.

USER-DEFINED INERTIAL RESISTANCE COEFFICIENT FOR NON-DARCIAN FLOW

When the **Flow model** in the **Porous Medium** feature is set to **Non-Darcian**, the **Porous Matrix** subnode has a new option to specify a user-defined **Inertial resistance coefficient**.

CORRECTED IMPLEMENTATION OF CONTACT ANGLES IN THE WETTED WALL FEATURE IN THE TERNARY PHASE FIELD INTERFACE

An error in the implementation of the Ternary Phase Field interface's **Wetted Wall** feature, which resulted in the wrong contact angle between the phases A and B when the three surface-tension coefficients were not identical, has been corrected. As a result, the definitions of the contact angles now agree with the figure shown in the feature's **Settings** window.

New Models in Version 6.4

REVERSE OSMOSIS WATER DESALINATION

This model considers a small reverse-osmosis water desalination unit consisting of a spirally wound semipermeable membrane through which salt water is forced under high pressure.

AXIALLY ROTATING PIPE AND SWIRLING JET TURBULENT FLOWS

The Turbulent Flow, SSG-LRR interface is used to analyze fully developed turbulent flow in an axially rotating pipe and in the swirling jet emanating from the pipe.

SPAN-WISE, ROTATING TURBULENT CHANNEL FLOW

The new Turbulent flow, Elliptic Blending R-ε interface is used to analyze fully developed turbulent flow in a span-wise rotating channel. Due to the influence of the

Coriolis force on the turbulence, profiles of the flow variables are skewed at moderate rotation rates.

Backward Compatibility with Version 6.3 and Earlier

The expression for **Gas transport** > **Isotropic diffusion** under the **Inconsistent Stabilization** section of the settings of the Bubbly Flow interfaces has been replaced by an expression similar to those used for the other dependent variables. The **Tuning** parameter can be reduced to avoid excessive smearing.

Chemical Reaction Engineering Module

New Functionality in Version 6.4

AGGREGATION, BREAKAGE, AND DISSOLUTION SUPPORTED FOR PRECIPITATION AND CRYSTALLIZATION

The **Size-Based Population Balance** interface has been updated with added support for aggregation, breakage, and dissolution of particles. A new logarithmic size discretization option is available and is recommended when modeling either aggregation or breakage.

Predefined size variables are now available in the **Size-Based Population Balance** interface and can be used in many of the available input fields. When defining a size distribution using a size-dependent input, the discretized representation now more accurately represents the continuous function.

The **Precipitation and Crystallization in Fluid Flow** multiphysics interface has been improved to account for turbulent particle transport defined by the turbulent viscosity and a turbulent Schmidt number.

NONISOTHERMAL REACTING FLOW LARGE EDDY SIMULATIONS

Three Large Eddy Simulation interfaces, LES RBVM, LES RBVMWV, and LES Smagorinsky, have been added to the Model Wizard in the Chemical Species Transport branch under Nonisothermal Reacting Flow. These physics interfaces simulate fluid flow with mass and heat transfer in the turbulent regime and apply large eddy simulation (LES) modeling for the unresolved scales of all three physics. In addition to a Chemical Reaction Engineering Module license, these interfaces require a license for the CFD Module.

REACTION FEATURES WITH IMPROVED CHEMISTRY COUPLING

Reaction features in mass transfer interfaces are now automatically linked to the reactions in a selected **Chemistry** interface. When these are coupled, the species matching in the **Chemistry** interface is used to assign mass sources in accordance with the chemical equations. Furthermore, the reaction features now include functionality to specify and report the resulting heat of reaction as a heat source, using the geometry

selection where the reaction occurs. The heat source, which is automatically defined when coupling to a **Chemistry** interface, can be applied in a **Heat Transfer** interface.

In the **Transport of Diluted Species** interface, this improved coupling functionality is available in the following features:

- **Reactions**
- **Heterogeneous Reactions**
- **Equilibrium Reaction**

In the **Transport of Concentrated Species** interface, this improved coupling functionality is available in the following features:

- **Reactions**
- **Equilibrium Reaction**

SUPPORT FOR MOVING PACKED BEDS

The **Packed Bed** feature has been extended to support bed translation relative to the reactor. When enabled, the name of the feature changes to **Moving Packed Bed**. The new functionality is available in the **Transport of Diluted Species** interface and the **Transport of Concentrated Species** interface.

Three new multiphysics interfaces supporting translation of the packed bed have been added:

- **Reacting Flow in Moving Packed Beds, Transport of Diluted Species**
- **Reacting Flow in Moving Packed Beds, Transport of Concentrated Species**
- **Reacting Flow in Moving Packed Beds, Transport of Diluted Species with Shrinking Core Model**

These all combine the functionality of the **Brinkman Equations** interface, a mass transfer interface for a dilute or concentrated solution, and a reacting flow multiphysics coupling. When added, the mass transfer interface includes a **Moving Packed Bed** feature.

SUPPORT FOR USING NONIDEAL THERMODYNAMICS MODELS IN VAPOR–LIQUID EQUILIBRIA

Support has been added for nonideal mixtures from the **Thermodynamics** node for vapor–liquid features (**Vapor Inflow**, **Vapor–Liquid Interface**, and **Vapor–Liquid-Solution Interface**). This allows the user to leverage the COMSOL® software's database of

thermodynamic properties (vapor pressure, heat of vaporization, and more) for multicomponent solutions.

IMPROVED MULTIPHYSICS SETUP FOR VAPOR–LIQUID EQUILIBRIA IN TWO-PHASE FLOW

When adding a **Laminar Two-Phase Flow** multiphysics interface from the **Vapor–Liquid Equilibrium** branch, a **Fluid–Fluid Interface** node is now automatically added to the **Laminar Flow** interface. Additionally, two nodes are now automatically added to the **Transport of Concentrated Species in Vapor** interface: a **Vapor–Liquid-Solution Interface** node for internal boundaries and a **Vapor–Liquid Interface** node for external boundaries.

ENHANCED SUPPORT FOR THE CHEMKIN® FORMAT

The cost of computing transport properties for mixtures scales quadratically with the number of species; for this reason, there is now a customizable threshold for the maximum number of species for these expressions to be generated.

Chemical species with special characters in their names are now automatically renamed, with their original name available in the shape function's description.

OTHER IMPROVEMENTS

- The **Reaction** node, which implements a single reaction in laminar or turbulent flow, has been reintroduced in the **Transport of Concentrated Species** interface.
- The **Size-Based Population Balance** interface has a new **Number of Intervals** input field, available in the **Size intervals** section together with the settings for the minimum and maximum particle sizes.
- The **Vapor–Liquid-Mixture Interface** feature in the **Transport of Concentrated Species** interface is now named **Vapor–Liquid-Solution Interface**.
- The **Species Source** feature in the **Transport of Diluted Species** interface has been renamed **Heterogeneous Reactions** and is used to describe phase-transfer reactions in a system.
- The **Feed Inlet** feature in the **Reaction Engineering** interface has been improved to allow for temperature-dependent specification of enthalpies.
- The **Reaction Thermodynamics** subfeature of the **Reversible Reaction Group** and **Equilibrium Reaction Group** features in the **Reaction Engineering** interface has been removed due to being redundant and susceptible to misconfiguration.

TURBULENT AGGREGATION OF NANOPARTICLES

Nanoparticles have become widely used in a range of applications. For example, in pharmaceuticals, particles on the nanoscale, such as nanocrystals, are often used as an effective means of drug delivery. This model solves for the turbulent aggregation of particles in the range of a few hundred nanometers in a beaker fitted with a square blade impeller in order to predict the evolution of the size distribution.

CHEMICAL VAPOR DEPOSITION OF TITANIUM NITRIDE ON MACHINING TOOL INSERTS

Machining tools are often coated with titanium nitride (TiN) to improve their wear resistance, reduce friction, and decrease their oxidation. This model simulates the process of coating metal inserts with TiN through chemical vapor deposition (CVD). The system studied consists of a horizontal tray in a cylindrical CVD reactor. A gas containing titanium chloride ($TiCl_4$) is fed into the system at the center of the reactor, flowing over the inserts placed on the tray and exiting the system through openings in the outer reactor wall. The titanium chloride in the feed gas reacts with nitrogen in the gas phase to form solid TiN that deposits on the inserts. The model considers fluid flow and mass transfer in the gas phase together with the buildup of solid TiN on the inserts. Both the deposition rate and the solid layer buildup are analyzed as a function of time. With the aim of minimizing the variation of the solid layer thickness, the impact of rotating the inserts is also investigated.

CHARGING OF A METAL HYDRIDE TANK

Metal hydride tanks offer safe hydrogen storage, thanks to their low reactivity, and a relatively high hydrogen density. When developing metal hydride tanks, modeling and simulation is useful for optimizing operating conditions, such as gas composition, pressure, and temperature, based on specific design parameters like metal composition, porosity, and geometric dimensions.

In this example, the charging process of a metal hydride tank is described. It includes the thermodynamics and absorption kinetics and computes the flow, concentrations, and temperature fields.

CONVECTIVE EVAPORATION OF A WATER–ACETONE DROPLET

In this model, the evaporation of a water–acetone droplet on a marble substrate is studied in detail. The droplet consists of a concentrated aqueous solution that initially contains 20% acetone by weight. The model describes the coupled phenomena of mass

and heat transfer across the vapor–liquid interface and accounts for multiphase flow by solving for the velocity in both the vapor and the liquid. As the droplet evaporates, the model tracks the position of the vapor–liquid interface.

Practical applications where detailed modeling of evaporation is valuable are found in a wide range of industries such as the pharmaceutical industry, fine chemicals production, and other industries where separation and drying processes need to be investigated.

Backward Compatibility with Version 6.3 and Earlier

- In version 6.4, the **Size-Based Population Balance** interface uses a new input field for entering the number of discretized intervals. Java scripts using code from version 6.3 need to be updated. A command of the kind:

  ```
  model.component("comp1").physics("pbsb").field("populationnumberd
  ensity").component(new String[]{"n1", "n2", "n3", "n4", "n5", "n6",
  "n7", "n8", "n9", "n10"});
  ```

  needs to be changed to:

  ```
  model.component("comp1").physics("pbsb").prop("ParticleSizeInterv
  als").set("NumberOfIntervals", 10);
  ```

- The improved discretized representation of the continuous size distributions in the **Size-Based Population Balance** interface can lead to a slight change in results.

Composite Materials Module

New Functionality in Version 6.4

LAYERED MATERIAL IMPROVEMENT

The rotation angle input in the **Layered Material** node under **Global Definitions** can now be defined as a variable. This enhancement enables the modeling of variable-angle tow laminates using the Layered Shell and Shell interfaces.

LAYERED SHELL IMPROVEMENTS

The default plot now displays the stress tensor component in the fiber direction instead of the von Mises stress. Two results templates now show lamina or laminate stress tensor components instead of the von Mises stress.

The formulations for the **LaRC03** and **Hashin** failure criteria have been improved.

The foldline constraint handling has been improved when a **Layered Material Stack** material node is used.

Laminate stiffness and flexibility matrices have been added as results variables for the **Linear Elastic Material** and **Piezoelectric Material** material models.

LAYERED-SHELL–STRUCTURE TRANSITION IMPROVEMENTS

An improved formulation for structural connections has been implemented. The variable for the normal direction has been corrected.

The new default constraint method is set to **Nodal**. In earlier versions, the default was **Elemental**.

New and Updated Models, Apps, and Add-ins in Version 6.4

PARAMETRIC STUDY ON CROSS-PLY LAMINATE FAILURE

This example demonstrates how to perform a failure analysis of laminated composite shells using the Layered Shell interface. The laminate is subjected to multiple load cases and loaded to failure, considering in-plane stress states. Both classical and advanced failure criteria are compared.

Backward Compatibility with Version 6.3 and Earlier

The user inputs related to the Young's modulus, shear modulus, and Poisson's ratio have been removed from the settings in the **LaRC03** failure criterion. When the parent material is isotropic or orthotropic, the **LaRC03** failure parameters are taken from the parent node. The new behavior is the same as that of the **From parent** option found in the earlier versions.

Corrosion Module

New Functionality in Version 6.4

AQUEOUS ELECTROLYTE TRANSPORT INTERFACE

A new Aqueous Electrolyte Transport interface is now available. This interface computes the potential and species concentration fields in a dilute aqueous electrolyte. The new interface is specialized for modeling aqueous electrolytes featuring weak acids, weak bases, ampholytes, and generic complex species and can, for instance, be used for mechanistic corrosion modeling, electrochemical models of biological systems, and electrochemical sensor modeling.

Due to its more efficient handling of equation reactions and easier model setup, the new interface may be more preferable to use in some cases than the more generic Tertiary Current Distribution, Nernst–Planck interface. The transport is defined by the Nernst–Planck equations, including diffusion, migration, and convection, in combination with electroneutrality and the self-ionization equilibrium reaction of water (autoprotolysis).

LOAD CYCLE NODE

A new **Load Cycle** node has been added to the Primary Current Distribution, Secondary Current Distribution, and Tertiary Current Distribution interfaces. The node may be used to define arbitrary charge–discharge load cycles by adding **Voltage**, **Current**, and **Rest** child nodes, which are executed in sequence.

For each node in the load cycle sequence, the user may define one or multiple dynamic continuation or break (switching) criteria, which may be based on time, voltage, or current limits, as well as user-defined conditions using arbitrary variable expressions.

The **Load Cycle** node also allows for automatic definitions of current and voltage probes, as well as solver stop conditions.

The **Load Cycle** node is available both as a boundary node for porous electrode and current conductor domains and as a child node of the **Electrode Surface**, **Highly Conductive Porous Electrode**, **Thin Porous Electrode**, **Perforated Electrode Surface**, **Sacrificial Anode Surface**, and **Thin Electrode Surface** nodes.

POWER LOSS VARIABLES

New power loss variables have been introduced in the Electrochemistry interfaces. By using the new variables, it is possible to evaluate the magnitude of the total power losses in a battery cell and compare the losses of the individual components (such as the separator, electrode, and current conductor).

The power losses are defined by considering the losses in the Gibbs free energy of all reacting and transported species, which allows for differentiation between ohmic, concentration, and activation losses.

The variables are available both locally on domains and boundaries, as integrated variables for the whole cell, or integrated per individual model-tree feature node.

The power loss variables can be accessed during results visualization under **Power losses** (in the relevant Electrochemistry interface submenu) when clicking either **Insert expression** or **Replace expression**.

INITIAL VALUES FOR THE ION-EXCHANGE MEMBRANE NODE IN THE TERTIARY CURRENT DISTRIBUTION, NERNST–PLANCK INTERFACE

The **Ion-Exchange Membrane** node in the Tertiary Current Distribution, Nernst–Planck interface has a new **Add Donnan shift to initial values** option, which is enabled by default when creating a new model. The new option automatically shifts the initial concentration and potential values specified by the user in the **Initial Values** node for the active **Ion-Exchange Membrane** domain node, assuming that the user-defined values represent the values for a bulk liquid electrolyte in equilibrium with the membrane. The shifted initial values are then used as initial values for the solver.

The new option facilitates easier model setup since it typically excludes the need to sweep the fixed space charge of the membrane to a desired nonzero value using an additional study step.

The new option is enabled by default in new models, which means that Java API backward compatibility may be affected by the change.

DEFAULT SOLVER UPDATES

Reaction forces are no longer stored by default by the time-dependent solver. This improves performance for 1D models storing all solutions.

The recovery damping factor in the default solver for stationary study steps in most Electrochemistry interfaces has been reduced to 0.1 (from 0.75). This may improve convergence in certain cases.

CHANGED DEFAULT VALUES IN THE SACRIFICIAL EDGE ANODE AND SACRIFICIAL ANODE SURFACE NODES

The default value of mass-based initial anode capacity has been changed from 3[A*h/kg] to 2000[A*h/kg]. The new default value may impact Java API backward compatibility.

New and Updated Models, Apps, and Add-ins in Version 6.4

The following models in the Corrosion Module Application Library have been revised and updated to use the new Aqueous Electrolyte Transport interface:

- Accelerated Corrosion Test of a Scratched Galvanized Steel Sample
- Atmospheric Corrosion with Mass Transport
- Crevice Corrosion of Iron in an Acetic Acid/Sodium Acetate Solution
- Biodegradation of a Magnesium Stent
- Corrosion Under an Evans Droplet

ECAD Import Module

New Functionality in Version 6.4

IMPORT OF IPC-2581 AND ODB++ FILES

- A new **Preview** button is available in the toolbar of the **Settings** window for the **Import** node. It can be used to display a 2D preview of the copper and drill layers in the PCB file. The layers are displayed in the **Graphics** window, and the layers and nets are also listed in a separate **Preview** window. The preview functionality can be used to select layers and nets to import.
- A new **Nets to Import** section in the settings of the **Import** node contains the list of nets available in the selected PCB file. From this list, you can select the nets to import from the file. Doing so can reduce the time it takes to generate the PCB geometry when you know in advance that only a few of the nets are needed for modeling.
- A new **Pads** section in the settings of the **Import** node can be used to set up the creation of pad domains or faces. Pads can be created for selected components and pins. Attributes are automatically attached to the created geometric entities for pads and make it easy to set up physics definitions when leveraging **Logical Expression Selection** features.
- In the **Selection of Resulting Entities** section of the settings of the **Import** node, the new **Net selections per layer** checkbox controls the generation of selections that form the intersection of layer and net selections.
- The **Reload Layers** button has been moved from under the **Layers** table to next to the **Import** button in the **Source** section, and it has been renamed **Refresh Data from File** since it now loads both layer and net information from the file.

New and Updated Models, Apps, and Add-ins in Version 6.4

PCB IMPORT TUTORIAL SERIES

This new tutorial series demonstrates the import of a PCB file for generating a geometry for simulation. Key topics include previewing layers, generating shell or solid objects for copper traces, handling of vias, and creating simplified 3D components and pads. The tutorial models also demonstrate the removal of small details and how the geometry finalization method can influence the choice mesh generators.

PCB TO MATERIAL DATA ADD-IN

This new add-in simplifies a PCB geometry for simulation by creating a space-dependent interpolation function for the material properties. The interior structure of the board is represented by the interpolation function, so that the traces and vias do not need to be included in the geometry. The simplified PCB geometry is generated by extruding the board outline according to the layer arrangement in the file.

Electric Discharge Module

New Functionality in Version 6.4

MORE EFFICIENT SWITCHING ARC SIMULATION

Several systematic improvements have been made to the **Arc Discharge** multiphysics interface, including an improved moving mesh and remeshing solver for curl elements, as well as optimized material data and model formulations for improved convergence. A fully coupled 3D switching arc model — integrating magnetohydrodynamics, moving mesh, and external circuit effects — can now be completed within just a few hours on a standard PC.

You can explore these improvements in the tutorial model Switching Arc Discharges in Low-Voltage Circuit Breakers.

MORE ROBUST ELECTRIC DISCHARGE SIMULATION

Enhancements to numerical formulation, initial values, and boundary conditions in the **Electric Discharge** interface provide markedly improved stability and faster, more efficient computations. For time dependent studies, a new option **Include time steps effect on stabilization time scale** for Streamline diffusion **Consistent Stabilization** is available. The initial values and boundary conditions have also been adjusted to provide better initialization for solvers.

You can explore these enhancements in tutorial models such as Dielectric Barrier Discharge in Air and Surface Dielectric Barrier Discharge.

AUTOMATIC MATERIAL INTERFACE BOUNDARY CONDITION

The **Dielectric Interface, Surface Transport** feature enables automatic handling of arbitrary interfaces between insulators, conductors, and charge transport domains. You can also include additional physics effects such as surface conduction and surface charge density for specific material interfaces.

You can explore this new feature in the tutorial model Partial Discharge Inside Solid Dielectrics.

FLOATING ELECTRODE FEATURE

The new **Floating Electrode** feature introduces dedicated functionality for modeling floating electrodes and their interactions with charge transport.

You can explore this new feature in the tutorial model Streamers Initialized from Suspended Metal Particles.

DISSOCIATION FEATURE

The new **Dissociation** feature adds support for modeling the Wien effect in dielectric liquids. This is useful for modeling electrohydrodynamic flow in dielectric liquids.

You can explore this new feature in the tutorial model Electrohydrodynamic Flow in Dielectric Liquids.

EQUILIBRIUM DISCHARGE MATERIAL LIBRARY

The **Equilibrium Discharge** material library includes several updates: material properties now exhibit weak dependence on pressure, the stabilization electric conductivity has been adjusted from 1 S/m to 10^{-3} S/m, and a new material, SF_6, is added.

New Models in Version 6.4

DC BREAKDOWN VOLTAGE OF PARALLEL ELECTRODES IN AIR

This model calculates the DC breakdown voltage between parallel electrodes in air using a detailed charge transport approach. It is currently one-dimensional for simplicity but can be extended to other gases and dimensions. The results closely match experimental data found in the literature.

AC BREAKDOWN VOLTAGE OF PARALLEL ELECTRODES IN AIR

The model estimates the AC breakdown voltage between parallel electrodes in air by simulating charge transport dynamics. To keep computations efficient, it is implemented in one dimension, though it can be adapted for different gases and extended to higher-dimensional setups. The simulation results show strong agreement with experimental values reported in the literature.

SURFACE DIELECTRIC BARRIER DISCHARGE

This model demonstrates the simulation of surface dielectric barrier discharges (DBDs) using the Electric Discharge interface. The formulation incorporates a comprehensive set of physical processes, including charge transport, impact ionization, electron attachment, recombination, and surface charge accumulation, all fully coupled with the Poisson's equation.

DIELECTRIC BARRIER DISCHARGE IN AIR

This model demonstrates DBD in air subjected to an AC applied voltage. As the voltage amplitude increases, the discharge intensity correspondingly strengthens. Furthermore, the dominant component of the discharge current transitions from displacement current to conduction current.

ELECTROHYDRODYNAMIC FLOW IN DIELECTRIC LIQUIDS

This model simulates the electrohydrodynamic (EHD) flow of a dielectric liquid around a wire electrode positioned between two parallel flat-plate electrodes. The ion transport is described using the Poisson–Nernst–Planck equations, while fluid motion is governed by the Navier–Stokes equations. The model incorporates the Onsager effect to more accurately capture ion dynamics. Simulation results show strong agreement with experimental data reported in the literature.

SWITCHING ARC DISCHARGES IN LOW-VOLTAGE CIRCUIT BREAKERS

Unwanted electrical arcing poses significant risks to the reliability and safety of electrical and electronic systems. To enhance predictive capabilities and deepen understanding of arc behavior, this comprehensive numerical model simulates transient arc discharge phenomena in a circuit breaker environment. The model employs a full 3D magnetohydrodynamics-based **Arc Discharge** multiphysics interface, capturing the complex interplay between thermal, electromagnetic, and fluid dynamic effects during arc formation and evolution. A dynamic moving mesh with remeshing is implemented to accurately resolve topological changes associated with mechanical switching actions. Furthermore, the model is coupled to an external electrical circuit, enabling realistic simulation of arc-circuit interactions under operational switching conditions. This integrated approach provides a robust framework for investigating arc dynamics and supports the development of more effective arc mitigation and control strategies in power systems.

STREAMERS INITIALIZED FROM SUSPENDED METAL PARTICLES

This model simulates the initiation of streamers from suspended metal particles, their propagation under a high electric field, and their subsequent merging. The discharge current flows into the metal particles, which are maintained at equal potential. These suspended particles enhance the local electric field, thereby accelerating both the initiation and propagation of streamer discharges.

PARTIAL DISCHARGE INSIDE SOLID DIELECTRICS

The electrical insulation strength of gases is generally much lower than that of solids. Under normal operating conditions, electrical discharges — known as partial

discharges — can occur in voids or cracks within a solid dielectric. This model simulates partial discharges in a spherical air void embedded in a solid insulator under a 50-Hz power frequency. It incorporates detailed charge transport processes coupled with electrostatics while automatically accounting for surface charge accumulation at the interface. The simulation produces the phase-resolved partial discharge (PRPD) pattern.

Backward Compatibility with Version 6.3 and Earlier

- The charge transport formulations in **Gas**, **Liquid**, and **Solid** features in the **Electric Discharge** interface now include a background ionization with a rate of 10 $\mathrm{cm}^{-3}\ \mathrm{s}^{-1}$ by default. From the physics standpoint, this background ionization serves as a crucial source of seed electrons. In numerical simulations that use a logarithmic formulation for charge carriers, it also contributes to improved numerical stability. However, since the background ionization rate is several orders of magnitude lower than that of dominant source terms — such as impact ionization — it has a negligible effect in models where those stronger processes prevail.
- The **Initial Values** and default values of the **Number Density** boundary condition for charge carriers have been changed depending on the dielectric medium (**Gas**, **Liquid**, or **Solid**). The new values can help to improve model initialization and convergence.

Electrochemistry Module

New Functionality in Version 6.4

AQUEOUS ELECTROLYTE TRANSPORT INTERFACE

A new Aqueous Electrolyte Transport interface is now available. This interface computes the potential and species concentration fields in a dilute aqueous electrolyte. The new interface is specialized for modeling aqueous electrolytes featuring weak acids, weak bases, ampholytes, and generic complex species and can, for instance, be used for mechanistic corrosion modeling, electrochemical models of biological systems, and electrochemical sensor modeling.

Due to its more efficient handling of equation reactions and easier model setup, the new interface may be more preferable to use in some cases than the more generic Tertiary Current Distribution, Nernst–Planck interface. The transport is defined by the Nernst–Planck equations, including diffusion, migration, and convection, in combination with electroneutrality and the self-ionization equilibrium reaction of water (autoprotolysis).

POWER LOSS VARIABLES

New power loss variables have been introduced in the Electrochemistry interfaces. By using the new variables, it is possible to evaluate the magnitude of the total power losses in a battery cell and compare the losses of the individual components (such as the separator, electrode, and current conductor).

The power losses are defined by considering the losses in the Gibbs free energy of all reacting and transported species, which allows for differentiation between ohmic, concentration, and activation losses.

The variables are available both locally on domains and boundaries, as integrated variables for the whole cell, or integrated per individual model-tree feature node.

The power loss variables can be accessed during results visualization under **Power losses** (in the relevant Electrochemistry interface submenu) when clicking either **Insert expression** or **Replace expression**.

LOAD CYCLE NODE

A new **Load Cycle** node has been added to the Primary Current Distribution, Secondary Current Distribution, and Tertiary Current Distribution interfaces. The node may be

used to define arbitrary charge–discharge load cycles by adding **Voltage**, **Current**, and **Rest** child nodes, which are executed in sequence.

For each node in the load cycle sequence, the user may define one or multiple dynamic continuation or break (switching) criteria, which may be based on time, voltage, or current limits, as well as user-defined conditions using arbitrary variable expressions.

The **Load Cycle** node also allows for automatic definitions of current and voltage probes, as well as solver stop conditions.

The **Load Cycle** node is available both as a boundary node for porous electrode and current conductor domains and as a child node of the **Electrode Surface, Highly Conductive Porous Electrode**, **Thin Porous Electrode**, **Perforated Electrode Surface**, **Sacrificial Anode Surface**, and **Thin Electrode Surface** nodes.

INITIAL VALUES FOR THE ION-EXCHANGE MEMBRANE NODE IN THE TERTIARY CURRENT DISTRIBUTION, NERNST–PLANCK INTERFACE

The **Ion-Exchange Membrane** node in the Tertiary Current Distribution, Nernst–Planck interface has a new **Add Donnan shift to initial values** option, which is enabled by default when creating a new model. The new option automatically shifts the initial concentration and potential values specified by the user in the **Initial Values** node for the active **Ion-Exchange Membrane** domain node, assuming that the user-defined values represent the values for a bulk liquid electrolyte in equilibrium with the membrane. The shifted initial values are then used as initial values for the solver.

The new option facilitates easier model setup since it typically excludes the need to sweep the fixed space charge of the membrane to a desired nonzero value using an additional study step.

The new option is enabled by default in new models, which means that Java API backward compatibility may be affected by the change.

DEFAULT SOLVER UPDATES

Reaction forces are no longer stored by default by the time-dependent solver. This improves performance for 1D models storing all solutions.

The recovery damping factor in the default solver for stationary study steps in most Electrochemistry interfaces has been reduced to 0.1 (from 0.75). This may improve convergence in certain cases.

The Capacitive Deionization of Saline Water model in the Electrochemistry Module Application Library has been updated to use the new **Load Cycle** node.

The Desalination in an Electrodialysis Cell model in the Electrochemistry Module Application Library has been updated to make use of the new **Add Donnan shift to initial values** option in the **Ion-Exchange Membrane** node.

Electrodeposition Module

New Functionality in Version 6.4

AQUEOUS ELECTROLYTE TRANSPORT INTERFACE

A new Aqueous Electrolyte Transport interface is now available. This interface computes the potential and species concentration fields in a dilute aqueous electrolyte. The new interface is specialized for modeling aqueous electrolytes featuring weak acids, weak bases, ampholytes, and generic complex species and can, for instance, be used for mechanistic corrosion modeling, electrochemical models of biological systems, and electrochemical sensor modeling.

Due to its more efficient handling of equation reactions and easier model setup, the new interface may be more preferable to use in some cases than the more generic Tertiary Current Distribution, Nernst–Planck interface. The transport is defined by the Nernst–Planck equations, including diffusion, migration, and convection, in combination with electroneutrality and the self-ionization equilibrium reaction of water (autoprotolysis).

LOAD CYCLE NODE

A new **Load Cycle** node has been added to the Primary Current Distribution, Secondary Current Distribution, and Tertiary Current Distribution interfaces. The node may be used to define arbitrary charge–discharge load cycles by adding **Voltage**, **Current**, and **Rest** child nodes, which are executed in sequence.

For each node in the load cycle sequence, the user may define one or multiple dynamic continuation or break (switching) criteria, which may be based on time, voltage, or current limits, as well as user-defined conditions using arbitrary variable expressions.

The **Load Cycle** node also allows for automatic definitions of current and voltage probes, as well as solver stop conditions.

The **Load Cycle** node is available both as a boundary node for porous electrode and current conductor domains and as a child node of the **Electrode Surface**, **Highly Conductive Porous Electrode**, **Thin Porous Electrode**, **Perforated Electrode Surface**, **Sacrificial Anode Surface**, and **Thin Electrode Surface** nodes.

POWER LOSS VARIABLES

New power loss variables have been introduced in the Electrochemistry interfaces. By using the new variables, it is possible to evaluate the magnitude of the total power losses in a battery cell and compare the losses of the individual components (such as the separator, electrode, and current conductor).

The power losses are defined by considering the losses in the Gibbs free energy of all reacting and transported species, which allows for differentiation between ohmic, concentration, and activation losses.

The variables are available both locally on domains and boundaries, as integrated variables for the whole cell, or integrated per individual model-tree feature node.

The power loss variables can be accessed during results visualization under **Power losses** (in the relevant Electrochemistry interface submenu) when clicking either **Insert expression** or **Replace expression**.

INITIAL VALUES FOR THE ION-EXCHANGE MEMBRANE NODE IN THE TERTIARY CURRENT DISTRIBUTION, NERNST–PLANCK INTERFACE

The **Ion-Exchange Membrane** node in the Tertiary Current Distribution, Nernst–Planck interface has a new **Add Donnan shift to initial values** option, which is enabled by default when creating a new model. The new option automatically shifts the initial concentration and potential values specified by the user in the **Initial Values** node for the active **Ion-Exchange Membrane** domain node, assuming that the user-defined values represent the values for a bulk liquid electrolyte in equilibrium with the membrane. The shifted initial values are then used as initial values for the solver.

The new option facilitates easier model setup since it typically excludes the need to sweep the fixed space charge of the membrane to a desired nonzero value using an additional study step.

The new option is enabled by default in new models, which means that Java API backward compatibility may be affected by the change.

DEFAULT SOLVER UPDATES

Reaction forces are no longer stored by default by the time-dependent solver. This improves performance for 1D models storing all solutions.

The recovery damping factor in the default solver for stationary study steps in most Electrochemistry interfaces has been reduced to 0.1 (from 0.75). This may improve convergence in certain cases.

New and Updated Models, Apps, and Add-ins in Version 6.4

A new Aperture Shape Optimization for Electroplating of a Printed Circuit Board tutorial model has been added to the Electrodeposition Module Application Library.

Fuel Cell & Electrolyzer Module

New Functionality in Version 6.4

AQUEOUS ELECTROLYTE TRANSPORT INTERFACE

A new Aqueous Electrolyte Transport interface is now available. This interface computes the potential and species concentration fields in a dilute aqueous electrolyte. The new interface is specialized for modeling aqueous electrolytes featuring weak acids, weak bases, ampholytes, and generic complex species and can, for instance, be used for mechanistic corrosion modeling, electrochemical models of biological systems, and electrochemical sensor modeling.

Due to its more efficient handling of equation reactions and easier model setup, the new interface may be more preferable to use in some cases than the more generic Tertiary Current Distribution, Nernst–Planck interface. The transport is defined by the Nernst–Planck equations, including diffusion, migration, and convection, in combination with electroneutrality and the self-ionization equilibrium reaction of water (autoprotolysis).

POWER LOSS VARIABLES

New power loss variables have been introduced in the Electrochemistry interfaces. By using the new variables, it is possible to evaluate the magnitude of the total power losses in a battery cell and compare the losses of the individual components (such as the separator, electrode, and current conductor).

The power losses are defined by considering the losses in the Gibbs free energy of all reacting and transported species, which allows for differentiation between ohmic, concentration, and activation losses.

The variables are available both locally on domains and boundaries, as integrated variables for the whole cell, or integrated per individual model-tree feature node.

The power loss variables can be accessed during results visualization under **Power losses** (in the relevant Electrochemistry interface submenu) when clicking either **Insert expression** or **Replace expression**.

LOAD CYCLE NODE

A new **Load Cycle** node has been added to the Primary Current Distribution, Secondary Current Distribution, and Tertiary Current Distribution interfaces as well as the

Hydrogen Fuel Cell and Water Electrolyzer interfaces. The node may be used to define arbitrary charge–discharge load cycles by adding **Voltage**, **Power**, **Current**, and **Rest** child nodes, which are executed in sequence.

For each node in the load cycle sequence, the user may define one or multiple dynamic continuation or break (switching) criteria, which may be based on time, voltage, or current limits, as well as user-defined conditions using arbitrary variable expressions. By using the **Subloop** child node, dynamic switching between different cycling schemes is also possible.

The **Load Cycle** node also allows for automatic definitions of current and voltage probes, as well as solver stop conditions.

The **Load Cycle** node is available both as a boundary node for porous electrode and current conductor domains and as a child node of the **Electrode Surface**, **Highly Conductive Porous Electrode**, **Thin Porous Electrode**, **Perforated Electrode Surface**, **Sacrificial Anode Surface**, and **Thin Electrode Surface** nodes.

INITIAL VALUES FOR THE ION-EXCHANGE MEMBRANE NODE IN THE TERTIARY CURRENT DISTRIBUTION, NERNST–PLANCK INTERFACE

The **Ion-Exchange Membrane** node in the Tertiary Current Distribution, Nernst–Planck interface has a new **Add Donnan shift to initial values** option, which is enabled by default when creating a new model. The new option automatically shifts the initial concentration and potential values specified by the user in the **Initial Values** node for the active **Ion-Exchange Membrane** domain node, assuming that the user-defined values represent the values for a bulk liquid electrolyte in equilibrium with the membrane. The shifted initial values are then used as initial values for the solver.

The new option facilitates easier model setup since it typically excludes the need to sweep the fixed space charge of the membrane to a desired nonzero value using an additional study step.

The new option is enabled by default in new models, which means that Java API backward compatibility may be affected by the change.

DEFAULT SOLVER AND STUDY STEP UPDATES

Reaction forces are no longer stored by default by the time-dependent solver. This improves performance for 1D models storing all solutions.

The recovery damping factor in the default solver for stationary study steps in most Electrochemistry interfaces has been reduced to 0.1 (from 0.75). This may improve convergence in certain cases.

The **Free and Porous Media Flow** interface and **Reacting Flow, H2 Gas Phase** and **Reacting Flow, O2 Gas Phase** multiphysics nodes now support the **Frequency-Domain Perturbation** study step.

HYDROGEN FUEL CELL AND WATER ELECTROLYZER INTERFACES

- **O2** may now be enabled as an active species in the **H2 Gas Mixture** settings.
- **H2** may now be enabled as an active species in the **O2 Gas Mixture** settings.
- In the **H2 Inlet** and **O2 Inlet** nodes, when using the Mass flow rate as an Inlet flow type, a new **Stoichiometric feed** checkbox has been added (enabled by default). The option is used to model the common situation in an operating fuel cell or electrolyzer where the inlet flow rate of the gas mixture is set proportionally to the cell current in order to ensure excess species in the cell is consumed. The new default may impact Java API backward compatibility.
- The partial pressure variables are no longer regularized; that is, the nonnegative lower limit has been removed. This generally improves convergence due to nonvanishing gradients and may affect the convergence behavior of models created in previous versions.
- A **Solve for electrolyte phase potential** checkbox has been added in the (renamed) **Electrolyte and Membrane Transport** section of the settings for the main (top) interface node. When this checkbox is cleared, user-defined models for the electrolyte ion transport can be incorporated.
- The settings for the **Out-of-plane thickness** (2D and 1D with axial symmetry) and the **Cross-sectional area** (1D) properties have been moved from being defined on individual boundary nodes to a common edit field in the settings of the main (top) interface node.

FUEL CELL AND ELECTROLYZER MATERIAL LIBRARY

- The **Potassium Hydroxide, KOH** material has been modified to use an analytic function, which allows for easy plotting of the electrolyte conductivity as a function of T and C.
- A new polymer electrolyte material, **PiperION®, Liquid Equilibrated, Hydroxide Form**, has been added, with data for the electrolyte conductivity as a function of T.

New and Updated Models, Apps, and Add-ins in Version 6.4

The following new tutorial models have been added to the Fuel Cell & Electrolyzer Module Application Library:

- Ohmic and Activation Losses in a Polymer Electrolyte Membrane Water Electrolyzer Cell
- Carbon Corrosion in a PEM Fuel Cell
- Liquid Alkaline Electrolyzer with Concentrated Electrolyte Transport

In addition, the Mass Transport and Electrochemical Reaction in a Fuel Cell Cathode model and the Current Density Distribution in a Solid Oxide Fuel Cell model have been updated to make use of the new power loss variables.

Geomechanics Module

New Functionality in Version 6.4

PHASE-FIELD DAMAGE

A new multiphysics interface has been added, Phase-Field Damage, Explicit Dynamics. In terms of functionality, the new interface is equivalent to the Phase-Field Damage multiphysics interface, but the new interface couples the (also new) Solid Mechanics, Explicit Dynamics interface to the Phase Field in Solids interface. This new functionality is showcased in the model Phase-Field Modeling of Dynamic Crack Branching.

The **Phase-Field Damage** node has been extended with new options for modeling of quasibrittle fractures.

- The Rankine criterion has been added as a crack driving force.
- A new rational damage evolution function has been added.

The linear, exponential, and Cornelissen softening laws, common for modeling concrete, are available as rational degradation functions for the phase-field regularized cohesive zone model (PFCZM).

IMPROVED TIME DISCRETIZATION

The Euler steps used in the discretized equations to compute creep, viscoplasticity, polymer viscoplasticity, and adiabatic heating have been revamped.

The improvements enable faster evaluation of variables and help resolve issues with the initialization of state variables when continuing the solution in subsequent study steps.

ELASTOPLASTIC SOIL MODELS

All of the material models available in the **Elastoplastic Soil Material** node have been revamped. The user interface is new, and the implementation is more robust.

In the **Cap and Cutoff** subnode, a **Linear** option has been added to the **Hardening model** setting and a **Cam-clay** option has been added to the **Cap model** setting.

A new option, **None**, has been added to the **Load Reversal Points** section of the **Small Strain Overlay** and **Hardening Soil Small Strain** models.

New and Updated Models, Apps, and Add-ins in Version 6.4

TRIAXIAL AND OEDOMETER TESTS WITH HARDENING SOIL MATERIAL MODEL

In this example model, triaxial and oedometer tests are simulated using the Hardening Soil material model. The model has been updated to include an oedometer test, allowing a direct comparison between the two test types. Both tests display a hyperbolic stress–strain relationship. In the triaxial test, the axial stress is shown to approach the analytical failure stress asymptotically. In the oedometer test, loading and unloading cycles are performed to evaluate the robustness and consistency of the material model.

Backward Compatibility with Version 6.3 and Earlier

SMALL STRAIN OVERLAY MODEL

In the presence of initial strains, the measure of the historical elastic shear strain variable, `item.gammaeeHist`, starts as nonzero at the first parametric step. The correction in version 6.4 is to match the equivalent elastic shear strain variable, `item.gammaee`, when initial strains are added to the model. This correction was not implemented in version 6.3 and prior versions.

Heat Transfer Module

New and Improved Functionality in Version 6.4

PERFORMANCE IMPROVEMENTS FOR THE DISCRETE ORDINATES METHOD

In the **Radiation in Participating Media** and **Radiation in Absorbing–Scattering Media** interfaces, a new **Jacobian Contribution** section has been added when the **Radiation discretization method** is set to **Discrete ordinates method**. To display this new section, click the **Show More Options** button and select **Advanced Physics Options** in the **Show More Options** dialog.

The checkbox in this section, **Use reduced Jacobian**, can be used to activate or deactivate the contributions of several terms to the Jacobian matrix. When it is selected (the default), in combination with the improved solver settings of version 6.4, CPU time and memory requirements are significantly decreased, especially for quadrature sets with many directions. The previously used approach remains available by disabling **Use reduced Jacobian**. To benefit from the new default solver settings for models created in previous versions, reset the solvers to default.

Because this new approach is drastically different and shows great improvements, the **Performance index** option has been deprecated. The **Stabilization** section has been renamed **Inconsistent Stabilization**, and the **Use alternative solver with stabilization** option now does not have any effect on the solver sequence; it has therefore been renamed **Isotropic diffusion** since it only adds a stabilization term.

WEIGHTED SUM OF GRAY GASES MODEL

In the **Radiation in Participating Media** interface, a new option, **Weighted sum of gray gases model**, has been added to the **Wavelength dependence of radiative properties** input. This option enables modeling of nongray gases as a sum of gray gases. Its main application is in the domain of combustion.

Because of the underlying hypotheses of the model, not all features of the **Radiation in Participating Media** interface are available with this option.

HANDLING OF REFRACTION IN SURFACE-TO-SURFACE RADIATION INTERFACES

In the **Surface-to-Surface Radiation** and **Orbital Thermal Loads** interfaces, refraction is now handled by the already-existing **Semitransparent Surface** feature and the new

Refractive Interface and **Dielectric Window** features. When these features are applied on boundaries between domains of different refractive indices, the change of angle of the incident radiation is calculated through Snell's law. For the **Refractive Interface** and **Dielectric Window** features, the directional dependence of surface properties is automatically calculated from the Fresnel relations.

ANISOTROPIC THERMAL TURBULENCE MODELS

In the **Nonisothermal Flow** and **Reacting Flow** multiphysics couplings as well as in the **High Mach Number Flow** interfaces, it is now possible to use anisotropic thermal turbulence models. The Generalized Gradient-Diffusion Hypothesis (GGDH) model, and its high-order extension (HO-GGDH) are now available to model the anisotropy of the turbulence with respect to the heat equation. These models rely on the Reynolds stress tensor and therefore require a RANS-RSM model to account for turbulence anisotropy in the momentum equation. The RANS-RSM models and anisotropic thermal turbulence models are available with the CFD Module.

RADIATION SHIELD FEATURE FOR SURFACE-TO-SURFACE RADIATION INTERFACES

In the **Surface-to-Surface Radiation** and **Orbital Thermal Loads** interfaces, it is now possible to use the **Radiation Shield** feature. This feature makes it possible to model single-layer and multilayer radiation insulation. Typical application is for modeling shielding of spacecraft and cryogenic systems.

DIELECTRIC WINDOW FEATURE FOR SURFACE-TO-SURFACE RADIATION INTERFACES

In the **Surface-to-Surface Radiation** and **Orbital Thermal Loads** interfaces, it is now possible to use the **Dielectric Window** feature when the **Surface-to-surface radiation method** is set to **Ray shooting**. This feature makes it possible to model thin dielectric layers that reflect, absorb, and transmit radiation depending on the angle of incidence. Refraction is also handled by this feature. A typical application is for modeling the wavelength-dependent transmission and loss through windows that can be present in ovens, furnaces, spacecraft, and the built environment.

IMPROVEMENT OF THE REACTING FLOW MULTIPHYSICS COUPLING

In the **Reacting Flow** multiphysics coupling feature, the definition of the chemical reaction heat source has been updated, simplifying the setup of nonisothermal reacting flow models. This new approach removes the previous limitation on the number of **Reaction Source** features used in the coupled **Transport of Concentrated Species** interface

and increases modeling flexibility. The total reaction heat source can now either be user defined or automatically synchronized with the **Chemistry** interface.

Additionally, the **Reacting Flow** interface now includes heat contributions from equilibrium reactions defined using the **Equilibrium Reaction** features within the **Transport of Concentrated Species** interface.

API backward compatibility is not maintained for models that include the **Transport of Concentrated Species** interface with **Reaction Source** features in combination with the **Reacting Flow** coupling.

MISCELLANEOUS IMPROVEMENTS

- In the **Heat Transfer** interfaces, energy and heat balance variables evaluated on the material frame are now available. For models where the spatial and material frames are different — for example, when a **Moving Mesh** interface or a physics interface controlling the deformation is used — the balance variables of the material frame are more representative of the computation than the spatial frame's balance variables. When the frames are identical, so are these two sets of variables.
- In the **Surface-to-Surface Radiation** and **Orbital Thermal Loads** interfaces, new variables and **Results Templates** are available to help with checking the consistency of the model and the details of the radiation method, such as the number of adaptions required by the **Hemicube** method. These help variables can be found in the list of predefined quantities, in the **Consistency check** and **Radiation method** sections, when entering an expression for a plot.
- In the **Orbital Thermal Loads** interface, new **Results Templates** have been added to facilitate the analysis of results, showing the orbit in different planes.

New Models, Apps, and Add-ins in Version 6.4

AIR-COOLED BATTERY ENERGY STORAGE SYSTEM

In this model, a battery energy storage system (BESS) is cooled by forced convection of air in the cabinet. The batteries generate heat, which spreads through conduction. To avoid thermal runaway, the system is cooled using fans to force convective heat exchange with cold air. The model features grilles, screens, and fans that generate and modify the airflow through the system.

Updated Models, Apps, and Add-ins in Version 6.4

The tutorials have been updated with simplified step-by-step instructions for model construction that take advantage of the latest geometry, mesh, and solver features. In addition to these updates, some tutorials have been updated regarding physical modeling:

- The Radiative Heat Transfer in a Utility Boiler, Radiative Cooling of a Glass Plate, and Radiative Heat Transfer in Finite Cylindrical Media models now use the new reduced Jacobian approach as well as the new default solver sequence of the **Radiation in Participating Media** interface.
- The Forced Convection Cooling of an Enclosure with Fan and Grille model now uses the new **Perforated plate** option of the **Grille** feature.
- The Nonisothermal Turbulent Flow over a Flat Plate model has been updated to increase the Reynolds number and provide an analysis of the friction coefficient on top of the heat transfer coefficient.
- The Hepatic Tumor Ablation model now uses the **Use local time integration** option in the **Thermal Damage** feature.

Backward Compatibility with Version 6.3 and Earlier

- In COMSOL Multiphysics® version 6.4, the **Radiation in Participating Media** and **Radiation in Absorbing–Scattering Media** interfaces, when solved with the discrete ordinates method, use a new lumped Jacobian approach and a new default solver. Additionally, some options related to the solver sequence have been deprecated (**Performance index**) or modified (**Use alternative solver with stabilization**). The resulting solver of API scripts will thus be in a different state than before. If further manual changes to the solver sequence were made, the scripts may fail.
- In the **Radiation in Participating Media** and **Radiation in Absorbing–Scattering Media** interfaces, the **Semitransparent Surface** feature has been deprecated when the P1 approximation is used. If a **Semitransparent Surface** condition is necessary for the model, it is recommended to switch the **Radiation discretization method** to **Discrete ordinates method**. Backward compatibility is ensured for models in version 6.3 and earlier, but API code must be updated for the semitransparent condition to be active.
- The **External Temperature** subnode available for the **Thin Layer**, **Thin Film**, and **Fracture** features has been extended for the **Thermally thin approximation**. Note that this configuration was already possible in version 6.3 for the **Thin Layer** feature.

Backward compatibility with models created in previous versions is ensured. Some API scripts need to be updated to achieve the same behavior as in previous versions since **External Temperature** may now be active when it was not in previous versions.

MEMS Module

New Functionality in Version 6.4

SYMMETRY FACTORS FOR THE TERMINAL FEATURE

In the **Advanced Settings** section of the **Terminal** feature in the Electrostatics interface, there is now support for a **Terminal area multiplication factor**. This multiplication factor can be used to model the effects of symmetry. A connected load or circuit will then perceive the full device, even though the model includes only part of it. For details, see Symmetry Factors for the Terminal Feature in the release notes for the AC/DC Module.

GENERAL CONTACT

A new feature, **General Contact**, is available in the Solid Mechanics interface. This contact formulation is particularly useful when there are many different boundaries that can come into contact with each other. For details, see General Contact in the release notes for the Structural Mechanics Module.

SMOOTH PENALTY FUNCTION FOR CONTACT

A smooth penalty function has been added to the **Contact** and **Interior Contact** nodes. It is also available in the new **General Contact** feature. For details, see Smooth Penalty Function for Contact in the release notes for the Structural Mechanics Module.

LOGARITHMIC STRAINS

The control over the formulation of geometric nonlinearity in the **Geometric Nonlinearity** settings of the **Linear Elastic Material** node havs been updated to speed-up large strain analysis while giving virtually the same results as using full multiplicative decomposition. For details, see Logarithmic Strains in the release notes for the Structural Mechanics Module.

LINEAR BUCKLING FOR PERIODIC STRUCTURES

The Solid Mechanics interfaces now support the **Linear Buckling** study and study step. For details, see Linear Buckling for Periodic Structures in the release notes for the Structural Mechanics Module.

WARPAGE

It is now possible to compute warpage with respect to a reference plane defined by the user. For details, see Warpage in the release notes for the Structural Mechanics Module.

REACTION FORCES

New variables to compute reaction force and moments are available in the Solid Mechanics interfaces. For details, see Reaction Forces in the release notes for the Structural Mechanics Module.

CONTINUITY

A new checkbox, **Correct for mismatching boundaries**, has been added to the settings in the **Continuity** node. For details, see Continuity in the release notes for the Structural Mechanics Module.

DEFAULT PLOT OF TOTAL STORED ENERGY

A new default plot of total stored energy, `Wh_tot`, is available for **Frequency Domain** studies with electromechanics or piezoelectricity coupling. The default plot provides a visualization of the frequency response and resonance peaks of an electromechanical system.

New and Updated Models, Apps, and Add-ins in Version 6.4

BIRDBATH RESONATOR GYROSCOPE

This tutorial shows how to model and analyze a birdbath resonator gyroscope (BRG), a type of Coriolis vibratory gyroscope invented at the University of Michigan and commercialized by Enertia Microsystems as a millimeter-scale navigation-grade gyroscope. This tutorial model demonstrates how its shape allows for efficient coupling between drive and sense modes and how the Coriolis force is measured to realize a design that minimizes cost, weight, size, and power consumption while delivering higher accuracy and stability.

SHAPE OPTIMIZATION OF MICROPHONE DIAPHRAGM

This tutorial model demonstrates how to tune the natural frequency of a MEMS microphone diaphragm using the **Shape Optimization** interface and study. In this example, **Stationary** and **Eigenfrequency** study steps are used in succession to tune the natural frequency to a precise value by deforming the 2D shape of the suspension springs in accordance with standard MEMS fabrication specifications.

UNCERTAINTY QUANTIFICATION ANALYSIS OF PIEZOELECTRIC ENERGY HARVESTER

This tutorial model demonstrates how to use the Uncertainty Quantification Module to analyze the performance variation of a piezoelectric energy harvester that converts vibrational energy to electric energy. See how variations in manufacturing processes translate to variation in the device's output power as the quantity of interest (QoI). Different **Uncertainty Quantification** (UQ) studies provide design-of-experiments data from finite element method (FEM) simulations and a surrogate model to evaluate the QoI.

VELOCITY CALCULATION FOR SAW UNIT CELL

FEM simulations can be used to obtain the surface acoustic wave (SAW) velocity and other related parameters, such as the squared electromechanical coupling coefficient, or k^2, and reflectivity, or k_p, for different configurations. Such parameters are used as inputs for various analytical and semianalytical methods of SAW device design. This tutorial model shows how the unit cell and the **Periodic Condition** features are used in an **Eigenfrequency** study to investigate a 128° *YX*-cut lithium niobate ($LiNbO_3$) crystal sized for a center frequency of about 400 MHz with various electrode configurations.

SAW EULER ANGLE ROTATION

In general, surface acoustic wave (SAW) devices use different crystal cuts of piezoelectric materials that do not coincide with the reference crystal axes used in COMSOL® software. This tutorial shows how the 128° *YX*-cut lithium niobate ($LiNbO_3$) material property matrices are derived from default material properties using Euler angle rotation when the cut-specific data is not available. Euler angle rotations are applied in three configurations to calculate eigenfrequencies and eigenmodes.

Metal Processing Module

New Functionality in Version 6.4

MULTIPHYSICS INTERFACE FOR INDUCTION HARDENING

A new **Induction Hardening** multiphysics interface has been introduced for modeling of induction hardening processes for steel parts. This multiphysics interface combines the functionality of the Heat Transfer in Solids, Austenite Decomposition, Solid Mechanics, and Magnetic Fields interfaces. It requires both the AC/DC Module and the Metal Processing Module

PHASE TRANSFORMATION MODELS FOR AUSTENITIZATION

Two new phase transformation models have been introduced to model the transformation of ferritic phases of steel into austenite during heating.

- Using the **Linear** phase transformation model is a simple way to model austenitization. In this model, the rate of austenite formation is taken to be active between lower and upper temperature limits and proportional to the rate of temperature.
- The **Oddy–McDill–Karlsson** phase transformation model can be used to simulate heating of hypoeutectoid steels. Its mathematical form is based on the Johnson–Mehl–Avrami–Kolmogorov (JMAK) model, but it has a specific form for the time constant in the expression, and the equilibrium phase fraction is replaced by the eutectoid fraction of austenite.

STRESS-DEPENDENT MARTENSITIC TRANSFORMATION

You can now let stresses affect the onset of the phase transformation through a shift of the martensite start temperature, M_s. The start temperature is shifted depending on the pressure and the effective (von Mises) stress.

IMPROVED PHASE TRANSFORMATION COMPUTATIONS

The performance of phase transformation computations has been improved by forcing the fraction of one phase to be algebraically prescribed by the remaining phase fractions and the requirement that they add to unity.

IMPROVED RESULTS PROCESSING OF PHASE FRACTIONS

The visualization of computed phase fractions has been improved though the introduction of new Gauss-point-based phase fraction variables. For example: The variable corresponding to `audc.phase1.xi` is called `audc.phase1.xiGp`. Using these variables generally produces smoother plots.

TRANSFORMATION CONDITIONS FOR AUSTENITE DECOMPOSITION INTERFACES

When one of the Austenite Decomposition interfaces is added, **Transformation Condition** subnodes requiring isothermal or cooling conditions are now added to the ferritic, pearlitic, and bainitic phase transformations. This is also done for the corresponding phase transformations during import of phase transformation data from JMatPro®.

New and Updated Models, Apps, and Add-ins in Version 6.4

INDUCTION HARDENING OF A CYLINDRICAL PIN

This example shows how to model induction hardening of a small steel pin. The ferritic–pearlitic pin is heated using a coil with a high-frequency alternating current. The surface of the pin is thereby austenitized, and the pin is then water cooled to produce a hard, martensitic surface, retaining the softer interior ferritic–pearlitic composition.

Microfluidics Module

New Functionality in Version 6.4

ROTATING FRAME PROPERTY

A **Rotating frame** property has been added to the **Physical Model** properties in the Creeping Flow, Laminar Flow, and Brinkman Equations interfaces. When selected, the velocity field is defined relative to a rotating frame by including the fictitious Coriolis, centrifugal, and Euler forces in the momentum equation and a global **Rotating Frame** feature, with settings for the **Axis of rotation**, **Rotation speed**, and **Rotational direction**, is automatically added to the physics interface.

MARANGONI EFFECT FOR CONCENTRATION GRADIENTS

In the **Free Surface** and **Fluid–Fluid Interface** features, the **Surface tension coefficient** list has the new option **Library coefficient, liquid mixture/gas interface** with a selection of predefined binary liquid mixtures. When a Transport of Diluted Species interface is active in the component, the appropriate interface selection can alternatively be selected to apply the surface tension values of each corresponding species, which applies the concentration from the Transport of Diluted Species interface.

PRESSURE JUMP IN THE FREE AND POROUS MEDIA FLOW COUPLING

The Free and Porous Media Flow Coupling has a new option to **Include pressure jump across free–porous boundary**.

USER-DEFINED INERTIAL RESISTANCE COEFFICIENT FOR NON-DARCIAN FLOW

When the **Flow model** in the **Porous Medium** feature is set to **Non-Darcian**, the **Porous Matrix** subnode has a new option to specify a user-defined **Inertial resistance coefficient**.

CORRECTED IMPLEMENTATION OF CONTACT ANGLES IN THE WETTED WALL FEATURE IN THE TERNARY PHASE FIELD INTERFACE

An error in the implementation of the Ternary Phase Field interface's **Wetted Wall** feature, which resulted in the wrong contact angle between the phases A and B when the three surface-tension coefficients were not identical, has been corrected. As a result, the definitions of the contact angles now agree with the figure shown in the feature's **Settings** window.

Mixer Module

New Functionality in Version 6.4

MIXER PARTS WITH THICKNESS

The Mixer Module Part Library has been updated with new parts with finite thickness for the impeller blades and tanks. The **Impellers** section has been divided into **Impellers, Solid Blade Representation** (requiring a license for the Design Module) and **Impellers, Surface Blade Representation**.

Backward Compatibility with Version 6.3 and Earlier

Due to the changes in the structure of the Mixer Module Part Library, Java files using parts need to be updated. For example if a Rushton impeller was used, the line

```
model.geom().load(new String[]{"part1"},
"Mixer_Module\\Impellers,_Radial\\rushton_impeller.mph",
new String[]{"part1"});
```

should be replaced by

```
model.geom().load(new String[]{"part1"},"Mixer_Module\\
Impellers,_Surface_Blades\\Impellers,_Radial\\rushton_impeller.mph",
new String[]{"part1"});
```

Multibody Dynamics Module

New Functionality in Version 6.4

PHYSICS INTERFACE FOR JOINTS

A new physics interface named Joints has been added. It is intended for analysis of mechanical assemblies connected by various joints. The parts in the assembly can be rigid or flexible. Different structural elements can be connected through joints by defining **Attachment** nodes, and flexible parts can be defined using solid, shell, or beam elements.

The joints are the same as those available in the Multibody Dynamics interface. The main purpose of the new Joints interface is to facilitate couplings in other physics interfaces without having to add the full Multibody Dynamics interface.

NEW SELECTIONS IN ROLLER CHAIN PARTS

In addition to the existing selections, the following predefined domain selections are available in roller chain geometry parts:

- **Rollers**
- **Pins**
- **Link Plates**

GENERAL CONTACT

A new feature, **General Contact**, is available in the Multibody Dynamics interface. This contact formulation is particularly useful when there are many different boundaries that can come into contact with each other. Using this feature, you can avoid setting up many different contact pairs.

SMOOTH PENALTY FUNCTION FOR CONTACT

A smooth penalty function has been added to the **Contact** node. It is also available for the new **General Contact** feature. The new option uses a nonlinear (smooth ramp) penalty function for the contact pressure. This may improve the convergence significantly.

POINT MASS

A new **Point Mass** node is now available in the Multibody Dynamics interface.

SPRING–DAMPER

A **Loss factor** damping option is now available in the **Spring–Damper** node.

REACTION FORCES

New variables to compute reaction force and moments are available in the Multibody Dynamics interface. It is now possible to directly obtain the reactions for a single boundary condition, both for pointwise and weak constraints. The following nodes have this capability:

- **Fixed**
- **Prescribed Displacement**
- **Prescribed Velocity**
- **Prescribed Acceleration**
- **Roller**

CONTINUITY

A new checkbox, **Correct for mismatching boundaries**, has been added to the settings in the **Continuity** node. The purpose is to create a more accurate connection for cases when there is a gap or an overlap between the joined boundaries.

TRACTION VARIABLE

The traction variable, `phys.Trac`, is now always available on external and internal boundaries in the Multibody Dynamics interface.

QUATERNIONS

The variable for the real part of the quaternion for the **Rigid Connector**, **Rigid Material**, and **Attachment** nodes has changed. The internal dependent variable for the quaternion has been modified in order to be initialized with a zero solution.

ADHESION AND DECOHESION

Resetting of adhesion and decohesion was previously not possible. The adhesion condition and damage states are stored as internal degrees of freedom (DOFs) that keep track of the maximum value over time. These variables can now be reset to their initial values.

STRUCTURAL CONNECTIONS

The transformation matrix for the flexible option in the **Solid–Thin-Structure Connection** multiphysics coupling has been improved for geometric nonlinear analyses.

The implementation of the **Solid–Beam Connection** multiphysics coupling has been improved for geometric nonlinear analyses.

AVERAGE ROTATION

The internal dependent variable for the quaternion has been modified in order to be safely initialized with a zero solution.

The expression for the computation of the angular velocity vector has been corrected to account for large rotations.

ORTHONORMAL COORDINATE SYSTEMS

Only orthonormal coordinate systems defined in the spatial frame are allowed in the **Gravity**, **Base Excitation**, and **Linearly Accelerated Frame** nodes in all structural mechanics interfaces. This implies that such accelerations must always act in a space-fixed direction.

New and Updated Models, Apps, and Add-ins in Version 6.4

MODELING TRANSMISSION RATIO VARIATIONS IN A PLANETARY GEAR TRAIN

Planetary gears, also known as epicyclic gear systems, are a type of mechanism commonly used in applications that require compact design and high torque transmission. They are characterized by their ability to provide multiple gear ratios within a small space, making them ideal for use in automatic transmissions, electric drills, and aerospace applications.

This example illustrates the modeling of a helical planetary gear train consisting of a sun gear, a ring gear, a carrier and five planet gears. Three cases are analyzed by fixing different components.

DOOR SLAM ANALYSIS

This example illustrates the dynamics of a car door assembly during a slam event. The system consists of a car body, a door, and rubber seals. Contact pairs are defined between the car body, rubber seals, and the door. The rubber seal is modeled using a hyperelastic material model. A transient analysis is performed to compute accelerations and joint forces. The results are analyzed during the impact, the rebounding of the door as it locks, and the steady-state.

QUATERNION VARIABLES

In the **Attachment**, **Average Rotation**, **Rigid Connector**, and **Rigid Material** nodes in 3D, the internal dependent variable for the quaternion has been modified in order to be safely initialized with a zero solution. When opening an old model, this rescaling is done automatically to maintain full compatibility.

When building a model using the API, the new behavior will be used. This is true even if, for example, a Java file created from an older version is used.

If you, in the solver sequence, have manually changed the scale of the DOF for the quaternion defined by an **Average Rotation** node, you need to replace lines similar to

```
model.sol("sol1").feature("v1").feature("comp1_solid_avgr1_q")
.set("scaleval", 1);
```

with a line similar to

```
model.sol("sol1").feature("v1").feature("comp1_solid_avgr1_qd").s
et("scaleval", 1);
```

For **Attachment**, **Rigid Connector**, and **Rigid Material**, you only need to update existing code if you have deselected grouping of ODE variables in the physics interface (this is on by default). In that case, you need to replace lines similar to

```
model.sol("sol1").feature("v1").feature("comp1_solid_att1_a")
.set("scaleval", 1);
model.sol("sol1").feature("v1").feature("comp1_solid_a_rig1")
.set("scaleval", 1);
model.sol("sol1").feature("v1").feature("comp1_solid_rd1_a")
.set("scaleval", 1);
```

with lines similar to

```
model.sol("sol1").feature("v1").feature("comp1_solid_att1_a0")
.set("scaleval", 1);
model.sol("sol1").feature("v1").feature("comp1_solid_a0_rig1")
.set("scaleval", 1);
model.sol("sol1").feature("v1").feature("comp1_solid_rd1_a0")
.set("scaleval", 1);
```

SPRING–DAMPER

In the **Spring–Damper** node, available in the Multibody Dynamics interfaces, the energy dissipated in the damper is computed only when it is indicated in the **Energy Dissipation** section in the settings of the physics interface.

Nonlinear Structural Materials Module

New Functionality in Version 6.4

PHASE-FIELD DAMAGE

A new multiphysics interface has been added, Phase-Field Damage, Explicit Dynamics. In terms of functionality, the new interface is equivalent to the Phase-Field Damage multiphysics interface, but the new interface couples the (also new) Solid Mechanics, Explicit Dynamics interface to the Phase Field in Solids interface. This new functionality is showcased in the model Phase-Field Modeling of Dynamic Crack Branching.

The **Phase-Field Damage** node has been extended with new options for modeling of quasibrittle fractures.

- The Rankine criterion has been added as a crack driving force.
- A new rational damage evolution function has been added.

The linear, exponential, and Cornelissen softening laws, common for modeling concrete, are available as rational degradation functions for the phase-field regularized cohesive zone model (PFCZM).

IMPROVED TIME DISCRETIZATION

The Euler steps used in the discretized equations to compute creep, viscoplasticity, polymer viscoplasticity, and adiabatic heating have been revamped.

These improvements enable faster evaluation of variables and resolve issues with the initialization of state variables when continuing the solution in subsequent study steps.

POROUS PLASTICITY

The **Porous Plasticity** node has been improved: The user interface has been revamped and the implementation is now more robust. All isotropic hardening models available in the **Plasticity** node are now also available for the **Shima–Oyane**, **Gurson**, **Gurson–Tvergaard–Needleman**, **Fleck–Kuhn–McMeeking**, and **FKM–GTN** material models.

The power law isotropic hardening option has been removed; the same functionality is available with the **Swift** isotropic hardening model.

A linear hardening option and Cam-clay cap model have been added to the **Capped Drucker–Prager** model.

INTERCALATION STRAIN

A reference concentration has been added to the **Intercalation Strain** subnode of the **Hyperelastic Material** node. This addition makes it possible to define a strain-free configuration with a nonzero concentration.

HILL ORTHOTROPIC PLASTICITY

The user interface for the **Hill orthotropic** plasticity model has been revamped. The initial yield stress can either be computed from Hill's coefficients or defined as a user input parameter.

HYPERELASTIC MATERIALS

A new polynomial model has been added to the available models for the volumetric strain energy density.

VISCOPLASTICITY AND CREEP

A new subnode, **Set Variables**, has been added to the **Viscoplasticity** and **Creep** nodes. The purpose of this node is to reset inelastic states to model cases like stress relieving and annealing.

New and Updated Models, Apps, and Add-ins in Version 6.4

A LARGE-STRAIN POROVISCOELASTIC MODEL OF BRAIN TISSUE

This example demonstrates how to set up a fully coupled poroviscoelastic model of biological tissues. The model is benchmarked by simulating a cyclic uniaxial tension–compression test on human brain tissue.

INDENTATION OF A CYLINDRICAL BATTERY CELL

Indentation tests are commonly used to evaluate the safety of battery cells by assessing the risk of internal short circuits under external mechanical loads.

This example demonstrates how to set up such a test, in which an indenter is pressed onto a battery cell, causing significant deformation of its components. The analysis is performed using an explicit dynamics approach that accounts for plasticity and contact mechanics.

PHONE DROP TEST

Drop tests are used to evaluate how consumer products respond to impacts and high accelerations. In this example, the Solid Mechanics, Explicit Dynamics interface is used to numerically simulate a drop test of a mobile phone, highlighting the permanent deformation of the aluminum case and the damage sustained by the glass screen.

NEEDLE PENETRATION

Modeling the penetration of a needle into soft materials, such as gelatin, involves several complex mechanical challenges, including large deformations, high friction, contact forces, and material failure. This example simulates the needle insertion process into tissue-like materials, using a cohesive zone model to capture the fracture behavior.

CRIMPING OF A CABLE TERMINAL

Crimping is the plastic deformation process used to form an electrical joint between a stranded conductor and a terminal. The process involves complex multisurface contact, including self-contact, as the terminal sleeve is plastically deformed around the wire strands. This example analyzes the crimping of an aviation wire terminal using an explicit dynamics simulation.

Optimization Module

New Functionality in Version 6.4

OPTIMIZATION OF FINAL TIME

The time-dependent solver can be terminated based on a condition during the optimization, and the gradient computation will account for this, making it possible to minimize the duration of a process that stops once a certain condition is met.

PERFORMANCE FOR CONSTRAINTS

Gradient-based optimization of stationary and time-dependent problems now recycles the factorization and the Jacobian for the sensitivity computation of constraints, which can lead to dramatic performance improvements for problems with many optimization constraints. For transient models, the forward model is no longer recomputed when performing adjoint sensitivity analysis for the constraints, and the checkpointing is also only done once because the adjoints are computed in parallel.

PERFORMANCE FOR GCMMA

Gradient-based optimization with the globally convergent method of moving asymptotes (GCMMA) optimization solver now only computes sensitivities for outer iterations.

PARAMETER OPTIMIZATION STUDY STEP

Scaling of control variables is done based on the distance between the bounds in the **Parameter Optimization** study step. The optimization study step used in previous versions has been renamed **General Optimization**.

EXPORT OF OPTIMIZED PARAMETERS

Optimized parameters can be used in further analysis because control parameters can now be exported to parameter cases using a checkbox in the **General Optimization** and **Parameter Optimization** study steps.

CONTROL FUNCTION IMPROVEMENTS

The **Control Function** feature has also gained support for export of the optimized values to an **Analytic** or **Interpolation** function. The function type depends on the discretization of the function, and export is triggered using a checkbox in the feature. A variable for the average function value has also been added to the feature.

RANDOMIZATION OF BOUNDED GLOBAL CONTROLS

The **Parameter Optimization** and **General Optimization** study steps allow randomization of the initial values for bounded global controls based on a Sobol sequence. There is a shift property that can be changed using a **Parametric Sweep** so that different local optima can be identified.

P-NORM AND STANDARD DEVIATION

The maximum operator is not differentiable, so it cannot be used with gradient based optimization, but it can be approximated using a p-norm, and this functionality has been built into the **P-norm** feature. Similarly, it is possible to homogenize a field using the **Standard Deviation** feature.

SNOPT HAS BEEN DEPRECATED

The sparse nonlinear optimizer (SNOPT) optimization solver has been deprecated, so second-order convergence is limited to the interior point optimizer (IPOPT) and Levenberg–Marquardt solvers.

MMA SOLVER AT STUDY LEVEL

Previously, choosing the method of moving asymptotes (MMA) solver at the study level triggered the GCMMA solver, unless the solver configuration was manually changed. Now, it is possible to choose both the GCMMA and MMA solvers on the study level.

EXTRUSION MANUFACTURING CONSTRAINT

The **Density Model** feature has gained support for extrusion constraints, so incorporating an extrusion manufacturing constraint no longer requires manual setup with a **General Extrusion** operator. This functionality is also compatible with the **Topology Link** and **Prescribed Density** features.

FAR-FIELD OPERATOR FOR GRADIENT-BASED OPTIMIZATION

Operators have been added to the **Electromagnetic Waves** interface for evaluating the electromagnetic far field in a way that is compatible with gradient-based optimization. The operators — for computing gain, ratio, and other quantities — are available in 2D, 2D axisymmetry, and 3D. The norm of the electromagnetic field can be accessed as `comp#.emw#.normEfar_opt(x,y,z)`.

NONANALYTIC OPERATORS IN OBJECTIVES FOR EIGENVALUE SENSITIVITY AND OPTIMIZATION

It is now possible to use nonanalytic operators in the expression for the objective when performing eigenvalue sensitivity and optimization. Examples of nonanalytic operators include `real()`, `imag()`, and `abs()`.

ACCURATE BOUNDARY FLUX VARIABLES IN OBJECTIVES WITH ADJOINT SENSITIVITY

Gradient-based optimization can now be performed when objectives and/or constraints contain accurate boundary flux variables.

STABILITY IMPROVEMENTS FOR AUTOMATIC GRADIENT METHOD

Stability improvements have been implemented for the automatic gradient method, which selects either the forward or adjoint method depending on the number of objectives and controls, so that the number of model evaluations is minimized.

MINIMUM AND MAXIMUM CENTER OF SCALING IN TRANSFORMATION

Previously, the center of scaling and rotation in the **Transformation** feature was limited to **Average** and **User defined**, but now it is also possible to choose **Minimum** and **Maximum**, which is useful whenever an object is next to a **Symmetry/Roller** feature.

New and Updated Models, Apps, and Add-ins in Version 6.4

- The new 3D model Designing a Metasurface Beam Deflector Using Shape Optimization maximizes the deflection by changing the shape and position of the pillars.
- The new Aperture Shape Optimization for Electroplating of a Printed Circuit Board 3D model minimizes the inhomogeneity in an electroplating process.
- The Optimal Control for Heating of a Rod tutorial model has been updated to use condition-based final time.
- The Topology Optimization of a Beam with Milling Constraints tutorial model has been updated to reflect that the MMA solver can now be selected at the study level.
- The Topology Optimization of District Heating Network tutorial model is now easier to set up because the MMA solver can be selected at the study level.
- The Optimizing a Flywheel Profile tutorial model has been simplified due to the introduction of the **P-norm** feature.

- The Optimization of a Waveguide Iris Bandpass Filter — Transformation Version tutorial model has been updated to use randomization of the initial controls to identify several optima. The model also uses the maximum y-coordinate for the center of scaling in the **Transformation** feature.
- The Minimizing the Charging Time of a Lithium-Ion Battery tutorial model demonstrates optimization of final time by minimizing the charging time of a lithium-ion battery subject to a constraint on the battery degradation.
- The new Minimizing the Drying Time of a Wood Particle tutorial model extends the Superheated Steam Drying of a Wood Particle model with optimal control of the temperature.
- The Optimization of an Extruded MBB Beam tutorial model has been modified to use the **Free Shape Shell** feature so that equation-based modeling can be avoided.
- The Roundness Evaluation of a Deformed Hole tutorial model evaluates the roundness of some data points by computing the minimum circumscribed circle and maximum inscribed circle.
- The Shape Optimization of Coils tutorial model has been simplified due to the introduction of the **Standard Deviation** feature.

Backward Compatibility with Version 6.3 and Earlier

- The `optsolver` property has been replaced with an `optmethod` property in all optimization study steps. This supports both MMA and GCMMA, so the `gcmma` property on the **Optimization Solver** has been deprecated.
- Models using the now-removed SNOPT optimization solver been migrated to IPOPT, and API code choosing SNOPT now triggers the use of IPOPT; that is, `optsolver = snopt` is translated to `optmethod = ipopt`.
- The `millingActive` property has been replaced by the `manufacturingConstraint` property in the **Density Model** feature. Models are migrated accordingly, and there is API migration which translates `millingActive = Active` to `manufacturingConstraint = Milling`.

Particle Tracing Module

Improved Functionality in Version 6.4

IMPROVED NONRESONANT CHARGE EXCHANGE FEATURE

A new checkbox **Change primary ion species** has been added to the **Collision Frequency** settings of the **Nonresonant Charge Exchange** feature in the Charged Particle Tracing interface. The checkbox is selected by default and can be cleared when the ion species as a result of collision have the same properties as the primary particles. In such cases, the primary particles are reused instead of releasing new secondary particles.

IMPROVED HEAT SOURCE FEATURE

A new section, **Accumulator Settings**, has been added to the **Heat Source** feature, which improves the functionality of the feature when dealing with elastic collisions of the particles with the walls. The accumulated variable can now be affected by either just the particles that stick or freeze to the walls (the default) or by all particles that interact with the walls.

SIMPLIFIED OPERATOR NAMES

The names of the operators used by the physics features listed below have been simplified.

OLD NAME	NEW NAME
`intop`	`sum`
`maxop`	`max`
`minop`	`min`
`aveop`	`ave`

The following physics features utilize these updated operators:

- **Accumulator (Boundary)**
- **Boundary Load**
- **Current Density**
- **Erosion**
- **Etch**
- **Heat Source**
- **Mass Deposition**
- **Mass Flux**

- **Nonlocal Accumulator**
- **Particle Beam**
- **Surface Charge Density**

Backward Compatibility with Version 6.3 and Earlier

HEAT SOURCE

In versions 6.3 and earlier, the default behavior was to allow all particles interacting with the walls to influence the accumulated heat source variables. This behavior corresponds to the **Particle–wall interactions** option in the **Accumulate over** list. However, the default selection for the **Accumulate over** list in version 6.4 is **Particles in boundary elements**, with which the accumulated heat source is only affected by the particles that freeze or stick to the walls.

If a model built in version 6.3 or earlier is opened in version 6.4, the old default settings are maintained for the **Heat Source** features that already exist in the model. However, if a new **Heat Source** feature is added to the same model in version 6.4, the new default settings will be applied.

When building a model using the API, the new behavior is used. This is true even if, for example, a Java file created from an older version is used. If you want to achieve full backward compatibility using the API and you have code such as

```
model.component("comp1").physics("cpt").feature("wall1")
.create("hts1","HeatSource", 2);
```

the following line must be added:

```
model.component("comp1").physics("cpt").feature("wall1")
.feature("hts1").set("AccumulateOver","ParticleWallInteractions");
```

SIMPLIFIED OPERATOR NAMES

If a model built in version 6.3 or earlier is opened in version 6.4, the physics features will be automatically updated to use the new operators. However, if the operators are used as part of a user-defined expression — for example, in features for visualizing and evaluating results, such as **Global Evaluation** or **Color Expression** — the expressions will need to be updated.

Pipe Flow Module

New Functionality in Version 6.4

JUNCTION FEATURE FOR TRANSPORT OF DILUTED SPECIES IN PIPES

A new **Junction** feature is available for the Transport of Diluted Species in Pipes interface. It enables simulations of concentration profiles in complex piping networks by representing mass transfer at fittings with three or more connected pipes. The **Junction** node supports reversible flow in each branch by applying conditions that depend on the flow direction.

OPEN PIPE END FEATURE FOR TRANSPORT OF DILUTED SPECIES IN PIPES

The new **Open Pipe End** node makes it possible to connect a pipe to an external system with a specified concentration, for example, a tank or reservoir. The node supports reversible flow by automatically applying boundary conditions that adapt to the flow direction at the pipe end.

PIPE CONNECTION, DILUTED SPECIES

Connecting a pipe segment modeled with the Transport of Diluted Species in Pipes interface to a 3D domain modeled with the Transport of Diluted Species interface is now significantly easier thanks to the introduction of the new multiphysics coupling: **Pipe Connection, Diluted Species**. This feature enables direct coupling of concentration and mass flux between 1D pipe segments and 3D domains modeled with the Transport of Diluted Species interface, streamlining the model setup.

ORIFICE

A new **Orifice** feature is available in the Pipe Flow interface for modeling pressure drops caused by flow through orifices. Both sharp-edged and rounded orifices are supported, providing accurate representation of flow restrictions in piping systems.

CONTROL VALVE

A new **Control Valve** feature for modeling pressure losses introduced by adjustable valves is available in the Pipe Flow interface. The valve characteristics can be specified to represent different valve types and operating conditions, enabling accurate simulation of flow regulation in piping systems. The valve setup can be defined either

through a dimensionless loss coefficient or by using industry-standard flow coefficients, such as Cv and Kv, allowing direct use of manufacturer data.

3-WAY VALVE

A new **3-Way Valve** feature for modeling flow through valves with three connected pipes is available in the Pipe Flow interface. Both mixing and diverting regulated flow can be modeled. Parameterization can be done using a loss coefficient or with standard flow coefficients, such as Cv and Kv.

New and Updated Models, Apps, and Add-ins in Version 6.4

BOREHOLE HEAT EXCHANGER WITH PIPE FLOW

A new model, Borehole Heat Exchanger with Pipe Flow, demonstrates how to compute an array of borehole heat exchangers (BHEs) for shallow geothermal energy production. The BHEs are represented as simplified line elements using the Nonisothermal Pipe Flow interface, and the array is embedded in a layered subsurface model that includes groundwater flow in one of the layers.

Backward Compatibility with Version 6.3 and Earlier

For consistency with the Transport of Diluted Species in Pipes interface, the **Open Boundary** feature in the Heat Transfer in Pipes interface has been renamed **Open Pipe End**.

Plasma Module

New Functionality in Version 6.4

ENHANCED MODELING OF ETCHING AND DEPOSITION

A range of new functionalities has been introduced to enhance the modeling of etching and deposition processes in semiconductor applications. Surface species are now supported within the Stationary solver and the Plasma, Time Periodic interface, enabling faster parameterization. They can also be designated as site-constraint species to efficiently capture self-limiting reaction mechanisms. In addition, surface reactions can now be defined with energy-dependent reaction yields. View these functionalities in th following models:

- Modeling of a Silane/Argon ICP Reactor for Plasma-Enhanced Chemical Vapor Deposition of Amorphous Silicon
- Model of a CF4/O2 Inductively Coupled Plasma Reactor with RF Bias for Silicon Etching

NEW PLASMA CHEMISTRY IMPORT FEATURE

The new **Plasma Chemistry Import** feature can automatically create a complete plasma chemistry from a text file for models using the Plasma interface and the Plasma, Time Periodic interface. The contents of the text file can specify the aspects of a plasma chemistry, including species properties such as thermodynamic parameters, electron impact reactions from cross sections and rate constants, heavy-species reactions, and surface reactions. This feature is especially useful for plasma chemistries involving a large number of species and reactions. The following tutorial models use this new feature:

- Global Model of a CF4/O2 Plasma Reactor for Silicon Etching
- Model of an SF6/Argon Inductively Coupled Plasma Reactor

NEW SURFACE REACTION GROUP FEATURE

The **Surface Reaction Group** feature allows for the input of large sets of surface reactions in a table. Reaction data can be entered manually, imported from a file, or added

automatically using the **Plasma Chemistry import** feature. The following new models demonstrate the use of this feature:

- Modeling of a Silane/Argon ICP Reactor for Plasma-Enhanced Chemical Vapor Deposition of Amorphous Silicon
- Model of a CF4/O2 Inductively Coupled Plasma Reactor with RF Bias for Silicon Etching

NEW INDUCTIVE AND CAPACITIVE STOCHASTIC HEATING FEATURES

Two new features, **Inductive Stochastic Heating** and **Capacitive Stochastic Heating**, introduce an additional contribution to the electron collision frequency for momentum transfer, accounting for stochastic heating effects. This stochastic collision frequency is based on analytical models and can be applied to user-defined regions of the plasma domain.

GENERAL

- For the Plasma, Time Periodic interface, it is now possible to specify species mobility using the Dalgarno mobility model in the **Mobility Specification** section.
- Surface species are now supported in both the Stationary solver and the Plasma, Time Periodic interface.
- The **Metal Contact** feature can now be applied to both plasma-facing boundaries and nonplasma boundaries.
- The **Species** feature now includes a **User defined** option for specifying species thermodynamic parameters in the **Species Thermodynamic Parameters** section.
- The **Heavy Species Flux** feature now has an option to add convection to the prescribed flux.
- The **Inflow** feature has a new option that allows you to specify the mass flow rate.

New Models in Version 6.4

MODELING OF A SILANE/ARGON ICP REACTOR FOR PLASMA-ENHANCED CHEMICAL VAPOR DEPOSITION OF AMORPHOUS SILICON

This tutorial model studies the deposition of amorphous silicon using an inductively coupled plasma reactor with a silane/argon gas mixture. It examines how the deposition rate varies across the wafer as a function of silane mole fraction and input power.

MODEL OF A CF4/O2 INDUCTIVELY COUPLED PLASMA REACTOR WITH RF BIAS FOR SILICON ETCHING

This tutorial model studies the etching of silicon using an inductively coupled plasma reactor with an RF bias in a mixture of CF_4/O_2. The etching rate is computed along the wafer as a function of the RF bias voltage.

GLOBAL MODEL OF A CF4/O2 PLASMA REACTOR FOR SILICON ETCHING

In this example, the etching of silicon in a CF_4/O_2 plasma reactor is studied using a global model. Parametric sweeps for ion energy and oxygen mole fraction are computed.

MODEL OF A CF4/O2 INDUCTIVELY COUPLED PLASMA REACTOR

This tutorial model simulates an inductively coupled plasma reactor operating with a CF_4/O_2 gas mixture. The model solves plasma transport, magnetic fields, fluid flow, and heat transfer fully self-consistently.The oxygen mole fraction of the reactor feed is parameterized.

MICROWAVE PLASMA TORCH

This tutorial model explores the behavior of a microwave plasma torch. The plasma is generated within a dielectric tube inserted into a rectangular waveguide. The simulation solves a fully self-consistent set of coupled equations, including plasma transport and heating, Maxwell's equations, fluid flow, and heat transfer.

Backward Compatibility with Version 6.3 and Earlier

The **Inlet**, **Outflow**, **Flux**, and **Symmetry** subfeatures are deprecated. When opening models created with version 6.2 and earlier, the following warning message will appear: "This feature is obsolete and will be removed in a future version. Consider using the [Inflow/Outflow/Heavy Species Flux/Symmetry] feature instead."

Polymer Flow Module

New Functionality in Version 6.4

CURING REACTION

The Curing Reaction interface now includes a **Solid** node for modeling curing processes in solid domains. The Curing Reaction Heating multiphysics coupling has been extended to couple the Heat Transfer in Solids interface with the Curing Reaction interface in solids. In addition, the **Curing Kinetics** node has been renamed to **Fluid** for clarity.

CORRECTED IMPLEMENTATION OF CONTACT ANGLES IN THE WETTED WALL FEATURE IN THE TERNARY PHASE FIELD INTERFACE

An error in the implementation of the Ternary Phase Field interface's **Wetted Wall** feature, which resulted in the wrong contact angle between the phases A and B when the three surface-tension coefficients were not identical, has been corrected. As a result, the definitions of the contact angles now agree with the figure shown in the feature's **Settings** window.

New and Updated Models, Apps, and Add-ins in Version 6.4

PASTA EXTRUSION MODEL

The Pasta Extrusion model has been updated to use a parameterized geometry, making it easier to adjust dimensions and explore design variations.

Porous Media Flow Module

New Functionality in Version 6.4

PERIODIC CONDITION IN DARCY'S LAW INTERFACE

A **Periodic Condition** feature has been added to specify periodicity between two or more boundaries. In the **Periodic Condition** node, it is possible to enter either a **Pressure difference** or a **Mass flow** between source and destination boundaries.

GENERAL

- The **Inlet** boundary condition in the Darcy's Law and Richards' Equation interfaces now has the option to specify a **Standard flow rate** or **Standard Flow rate (SCCM)**.
- A new **Power law** permeability model specifies the permeability as a function of porosity.
- A user-defined non-Darcian flow option has been added to specify the resistance coefficient β directly.
- It is now possible to specify a pressure jump across a free-porous boundary within the **Free and Porous Media Flow, Darcy** multiphysics coupling.
- A new **Velocity Streamlines** result template is available for all flow interfaces. Additionally, existing result templates have been updated to improve visual clarity.
- Convergence has been improved for the Richards' Equation interface.
- The **Fluid** subnode, under the **Porous Medium** node, allows concentration as a **Model Input** to account for concentration-dependent material properties.

New and Updated Models, Apps, and Add-ins in Version 6.4

MODELING A LFA RAPID DETECTION TEST

This example models a lateral flow assay for the detection of coronavirus antibodies in a 2D geometry. The chemical species are homogeneously distributed over the membrane thickness.

A LARGE-STRAIN POROVISCOELASTIC MODEL OF BRAIN TISSUE

This example demonstrates how to set up a fully coupled poroviscoelastic model of biological tissues. The model is benchmarked by simulating a cyclic uniaxial tension–compression test on human brain tissue.

LINEAR BIPHASIC POROELASTICITY

This example demonstrates how to use the Poroelasticity multiphysics coupling between the Solid Mechanics and Darcy's Law interfaces to model linear poroviscoelastic behavior of soft biological tissues.

THE MANDEL–CRYER EFFECT

Cryer's problem is a three-dimensional consolidation benchmark where a porous sphere is subjected to a uniform boundary pressure. The pore pressure at the center of the sphere rises due to the Mandel–Cryer effect, which is modeled using the Poroelasticity multiphysics interface.

Ray Optics Module

New and Improved Functionality in Version 6.4

NEW SCATTER MODE IN SCATTERING DOMAIN FEATURE

The **Scattering Domain** feature now has a new **Scatter** option in the **Effect on primary ray** list in the **Scattered and Absorbed Rays** section, which can be used to trace the rays scattered by the particles in the medium. When you choose the **Scatter** option, you can either select **Henyey–Greenstein**, which is typically used for the asymmetric scattering, or **Rayleigh**, which is usually suited for small particles from the **Scattering phase functions** list.

SIMPLIFIED OPERATOR NAMES

The names of the operators used by some physics features have been simplified. The old and updated names are listed below:

OLD NAME	NEW NAME
intop	sum
maxop	max
minop	min
aveop	ave

The updated operators are used by the following physics features:

- **Accumulator (Boundary)**
- **Blackbody Radiation**
- **Release from Electric Field**

COMPLETE ABSORPTION–REFLECTION–TRANSMISSION (ART) MODELING ENABLED

The **Transmission coefficient** t field has been added in the **Absorption Coefficient** section of the **Mirror**, **Wall**, and **Illuminated surface** features. This field allows you to specify the transmission coefficient in addition to the reflection and absorption coefficients, enabling more accurate intensity and power computations at reflective boundaries.

RAYLEIGH SURFACE ROUGHNESS MODEL

You can now take into account the reduction of the intensity and power of the reflected rays due to surface roughness using the **Raleigh roughness** option (from the **Surface roughness model** list in the **Surface Roughness** section). This section is available in the **Mirror** and **Wall** features when ray intensity or power are computed. In the **Wall** feature, this new section appears when **Specular reflection** is selected from the **Wall condition** list in the **Wall Condition** section.

RAY RELEASE INSIDE GRADED AND DISSIPATIVE MEDIUM

A new checkbox called **Release inside graded and dissipative medium** appears in the **Ray Direction Vector** section of the release features when

- **Compute intensity in graded media** or **Compute intensity and power in graded media** is selected from the **Intensity computation** list in the physics interface node's **Intensity Computation** section and
- The **Store ray status data** checkbox is selected in the physics interface node's **Additional Variables** section.

Selecting this checkbox enables release of the rays inside media with spatially varying and complex refractive indices either at a single time or at multiple release times.

OPTICAL MATERIAL LIBRARY UPDATES

In the Optical material library, 12 new glasses from Schott AG and 10 new glasses from CDGM Glass Co., Ltd have been added. These added glasses have all the material properties required to perform structural-thermal-optical performance (STOP) analysis.

Backward Compatibility with Version 6.3 and Earlier

WALL

In the **Wall** feature, when the **Wall condition** is **Mixed diffuse and Specular reflection** and the **Primary ray condition** is either **Probability** or **Expression,** the **Otherwise** list has been updated to support nonreflected conditions. Therefore, if you have the following lines in the API code:

```
model.component("comp1").physics("gop").feature("wall1")
.set("WallCondition", "MixedDiffuseSpecular")
model.component("comp1").physics("gop").feature("wall1")
.set("Otherwise", "Disappear")
```

To achieve full backward compatibility, change `"Otherwise"` to `"OtherwiseForMixed"` and set its corresponding condition to one of `"Disappear"`, `"Freeze"`, `"Stick"`, or `"Pass"`.

SIMPLIFIED OPERATOR NAMES

If a model built in version 6.3 or earlier is opened in version 6.4, the physics features will be automatically updated to use the new operators. However, if the operators are used as part of a user-defined expression, for example, in postprocessing features such as **Global Evaluation** or **Color Expression**, the expressions will need to be updated.

RF Module

New Functionality in Version 6.4

PHYSICS INTERFACE: TRANSMISSION LINE, PARAMETERS

This new interface transitions from a complex multiphysics setup with two interfaces to a simpler, user-friendly configuration with minimal boundary conditions. It provides series resistance, series inductance, shunt conductance, and shunt capacitance per unit length, as well as characteristic impedance and propagation constant for two-conductor transmission lines. The computation is performed using frequency-domain modeling in 2D. This interface is introduced with the library example `transmission_line_coaxial`.

UPDATES IN TRANSMISSION LINE INTERFACE

Multiconductor transmission lines are now supported in the **Transmission Line** interface through multiple dependent variables. The distributed element parameters are defined as square matrices, with sizes corresponding to the number of dependent variables. The **Transmission Line** interface also supports series and shunt element features.

PHYSICS FEATURE: PERIODIC STRUCTURE

Designing metamaterials is now more streamlined with the **Periodic Structure** feature, which provides the default **Periodic Port** and **Floquet Periodic Condition** features. It is available in the **Electromagnetic Waves, Frequency Domain** interface. Several library examples demonstrate how to use this feature, including `fresnel_equations` and `frequency_selective_surface_csrr`.

PHYSICS FEATURE: IMPEDANCE BOUNDARY CONDITION AND TRANSITION BOUNDARY CONDITION

Lossy or conductive boundaries can be modeled in time-domain analysis. For electrically thin interior boundaries, use the **Transition Boundary Condition**, while for exterior boundaries or for surfaces of domains where thickness is electrically large (much greater than the skin depth) use the **Impedance Boundary Condition**. The example `dual_band_antenna_transient` demonstrates how to use the **Transition Boundary Condition**.

PHYSICS FEATURE: FAR-FIELD DOMAIN, INHOMOGENEOUS

Far-field radiation in the presence of a substrate can be analyzed with this new feature. It currently supports far-field computation for structures consisting of a superstrate (air) and a homogeneous substrate (dielectric). See the `embedded_scatterer_on_substrate` example.

ENHANCED FAR-FIELD FUNCTIONALITIES

Far-field functions are available for optimization, as demonstrated in a new library example that performs shape optimization to enhance the antenna gain. They are accessible from **Component** > **Definitions** > **Functions** in the **Results** context menu. The example `conical_horn_lens_antenna_shape_optimization` uses one of these far-field functions. New variables for circular polarization named `EfarLHCP` and `EfarRHCP` are also introduced for left-handed and right-handed circular polarization, respectively.

AUTOMATIC PORT PROBE

When a single port or lumped port is excited, a port probe is automatically added to monitor S-parameters during computation.

TOOLBAR BUTTONS

The toolbar buttons enable faster and easier modeling by analyzing the given materials and geometry and by configuring appropriate physics features and settings. The **Convert to a Half-Symmetry Model** button is available in the **Electromagnetic Waves, Frequency Domain** and **Electromagnetic Waves, Boundary Elements** interfaces. It creates a half-geometry and assigns a PEC- or PMC-type **Symmetry Plane** feature (Frequency Domain interface) or sets symmetry properties (Boundary Elements interface), depending on the user's selection among six available options. These buttons operate only in 3D. The examples `rcs_sphere` and `rcs_sphere_bem` demonstrate this functionality.

NEW ELECTRICALLY VERY THIN LAYER OPTION FOR TRANSITION BOUNDARY CONDITION FEATURE

In the **Transition Boundary Condition**, there is a new option in the **Type** parameter called **Electrically very thin layer**. This option represents the case of a very thin layer, where the electric field on the two sides of the boundary are almost the same. Thus, no slit of the dependent variable is used at the boundary.

PML DOMAINS NOT INCLUDED IN DEFAULT FIELD PLOTS

When **Perfectly Matched Layer** (PML) nodes are part of the model, the default field plots define a plot group selection that only includes the non-PML domains. Thereby, only the fields in the non-PML domains are visible in the plots.

PLOTS AT SYMMETRY PLANES ADDED TO DEFAULT FIELD PLOTS

When **Symmetry Plane** features are added, the default field plots replace slice plots parallel to the **Symmetry Plane** boundaries with **Surface** plots of the field at the **Symmetry Plane** boundaries.

EVALUATION GROUPS ADDED BY DEFAULT INSTEAD OF GLOBAL EVALUATION NODES

By default, in version 6.3 and earlier, when performing a study where there was no parametric sweep, **Global Evaluation** nodes were added for evaluation of S-parameters. Now, those **Global Evaluation** nodes are instead added to **Evaluation Group** nodes. This means that the evaluated expression values can be automatically updated after a study has been completed.

ELECTRIC POINT DIPOLE POWER VARIABLE ADDED IN 2D

A variable for the power radiated by the dipole has been added to the **Electric Point Dipole** feature in 2D (previously only available in 3D).

PART LIBRARY

With coplanar waveguide (CPW) trace parts, it is easier to build CPW geometries for modeling quantum computing devices. The example `cpw_resonator` uses these CPW trace parts.

MATERIAL LIBRARY

Substrate material properties from Rogers Corporation have been updated. An absorbing material from Hymag'in and a 3D-printing material for mmWave applications from Inkbit have also been added.

New and Updated Models, Apps, and Add-ins in Version 6.4

OPTIMIZING THE FRONT END OF A CONICAL HORN LENS ANTENNA

In this example, a parameterized lens geometry is transformed into a more effective lens shape, resulting in improved realized gain for the antenna. Shape optimization

removes the need for explicit parameterization and can achieve comparable or even better performance in a fraction of the time compared to multiple parametric sweeps.

MICROSTRIP PATCH ANTENNA SURROGATE MODEL

This example builds surrogate models with deep neural network (DNN) training to quickly estimate the performance of a microstrip patch antenna based on four design parameters: patch length, tuning stub length, dielectric constant of a substrate, and frequency. The model also simulates a full finite element method (FEM) model using asymptotic waveform evaluation of the computed S-parameters to efficiently evaluate the frequency response with fine frequency resolution.

EMBEDDED SCATTERER ON A SUBSTRATE

A plane transverse electric (TE)–polarized electromagnetic wave is incident on a metallic sphere embedded on a substrate. In this electromagnetic scattering problem, the far-field variables are computed for a few different elevation angles of incidence.

MRI IMPLANT HEATING

This MRI implant heating tutorial model shows the RF-induced heating on a passive conductive polyaxial screw implant inserted into C3-C4-C5 anatomical vertebrae and plunged into a jelly phantom brick so as to mimic the tissue enclosing the spine. The phantom has standard physical properties and geometry as reported in the ASTM-F2182-19e2 standard.

EFFICIENT MODELING OF A SPHERICAL RADOME

This model demonstrates an efficient approach to simulate a thin, spherical, large radome using a 2D axisymmetric formulation with cubic discretization. Using higher-order elements such as cubic element on a coarse mesh can significantly reduce computational cost while preserving accuracy.

TRANSIENT ANALYSIS OF A PRINTED DUAL-BAND STRIP ANTENNA

This model demonstrates how to represent lossy conductive surface using a **Transition Boundary Condition** in the time domain.

COMPUTING THE RADAR CROSS SECTION OF A PERFECTLY CONDUCTING SPHERE

This model uses toolbar buttons to automatically set up a quarter-sized geometry and user-specified **Symmetry Plane** features.

RCS OF A METALLIC SPHERE USING THE BOUNDARY ELEMENT METHOD

This model utilizes a toolbar button to build a half-sized geometry and configure symmetry properties in the physics interface settings.

FREQUENCY SELECTIVE SURFACE, PERIODIC COMPLEMENTARY SPLIT RING RESONATOR

The new physics feature, **Periodic Structure**, is demonstrated in this example. The modeling workflow is now more streamlined for building periodic models such as metamaterials.

FRESNEL EQUATIONS

This model uses the **Periodic Structure** feature to quickly set up a virtually infinite structure in the *xy*-plane.

IMPEDANCE MATCHING OF A LOSSY FERRITE 3-PORT CIRCULATOR

The steps for building the geometry in the model has been simplified. The model utilizes a **Frequency Domain Source Sweep** study to construct the full S-parameter matrix instead of using the conventional port sweep.

TRANSMISSION LINE PARAMETERS OF A COAXIAL CABLE

The new physics interface, **Transmission Line, Parameters,** is used to compute the distributed parameters of a coaxial cable. Previously, the computing was done with two physics interfaces through a multiphysics coupling. It is now possible to do it through a single dedicated physics interface with less boundary conditions.

Backward Compatibility with Version 6.3 and Earlier

DEFAULT VALUE FOR SLIT TYPE CHANGED TO DOMAIN-BACKED

The **Slit Type** parameter in the settings for the **Port** feature has changed from **PEC-backed** to **Domain-backed**. This change has consequences for models generated by API code. If the code previously did not explicitly set the **Slit Type** parameter for the port, the slit type would be PEC-backed. Now, the same code will get the Domain-backed slit type. To get PEC-backed slit type behavior, manually add a Java statement similar to this one:

```
model.component("comp1").physics("ewfd").feature("port1")
.set("SlitType", "PECBacked");
```

Rotordynamics Module

New Functionality in Version 6.4

ROTATING FRAME

A new default node, **Rotating Frame**, has been added to the Solid Rotor and Solid Rotor, Fixed Frame interfaces. The new node simplifies the workflow and improves the performance when performing rotordynamics simulations.

In previous versions, the rotational velocity and inclusion/exclusion of spin-softening effects were defined either in the physics interface settings or in the **Change Rotor Speed** node. In previous versions, the rotor axis was defined in the **Rotor Axis** node, along with its associated subnodes. With the new **Rotating Frame** default node, all rotational properties are defined in one dedicated node.

UPDATED LOCAL COORDINATE SYSTEM

The Solid Rotor and Solid Rotor, Fixed Frame interfaces now define a local coordinate system. In previous versions, the local x direction was used as the axis of rotation. In version 6.4, the local z direction is used instead for consistency.

SERENDIPITY SHAPE FUNCTIONS

Serendipity shape functions are now available in the Solid Rotor and Solid Rotor, Fixed Frame interfaces. The following options are now available: **Quadratic serendipity**, **Cubic serendipity**, and **Quartic serendipity**, in addition to the previously available Lagrange shape functions.

NEW DEFAULT DISCRETIZATION

The default discretization used for the displacement field in the Solid Rotor and Solid Rotor, Fixed Frame interfaces has been changed to **Quadratic serendipity**.

CHANGE ROTOR SPEED DEPRECATED

The **Change Rotor Speed** feature has been deprecated in the Solid Rotor and Solid Rotor, Fixed Frame interfaces. The rotational velocity is now defined in the **Rotating Frame** feature.

ROTOR AXIS DEPRECATED

The **Rotor Axis** node has been deprecated. The same functionality is available in the new **Rotating Frame** node.

FIRST SUPPORT AND SECOND SUPPORT DEPRECATED

The **First Support** and **Second Support** features, subfeatures to the **Rotor Axis** feature, have been deprecated. The same functionality is available using the **Rotating Frame** feature with the **Axis of rotation** selected as **From point selection**.

AXIS DEPRECATED

The **Axis** subnode under the deprecated **Rotor Axis** node has been deprecated. The functionality is now available in the new **Rotating Frame** node when **Axis of rotation** is selected as **From edge selection**.

WHIRL RESULT TEMPLATE

The **Whirl** result template has been updated. Unique **Whirl** result templates are now available for each **Rotating Frame** node in the model. This make it easier to analyze the results if multiple rotors are modeled simultaneously.

CAMPBELL AND STABILITY DIAGRAMS

The result templates for **Campbell Diagram** and **Stability Diagram** have been updated. The creation time has been reduced.

EXTENDED FUNCTIONALITY FOR DYNAMIC COEFFICIENTS

The functionality for determining the stiffness and damping dynamic coefficients in the Hydrodynamic Bearing interface has been improved and extended to support additional boundary conditions. The extended functionality is also valid when using:

- The **Symmetry** node
- The **Wall** node
- The **Inlet** edge features with the **Normal inflow velocity** condition
- The **Outlet** edge features with the **Normal outflow velocity** condition
- The **Border** feature with the **Border flow** condition
- The **Inlet** and **Outlet** nodes

MODE FOLLOWING

You can now use *mode following* when performing parametric eigenfrequency studies. The new functionality can be used to track eigenmodes as functions of a parameter. This is particularly useful when creating Campbell and stability diagrams.

New and Updated Models, Apps, and Add-ins in Version 6.4

VALIDATION OF DYNAMIC COEFFICIENTS IN HYDRODYNAMIC BEARINGS

In this verification example, the built-in functionality for calculating dynamic coefficients in hydrodynamic bearings is validated. The results are compared with those obtained using the finite perturbation method to ensure consistency and accuracy.

ROTORDYNAMIC ANALYSIS OF A CRANKSHAFT

The updated model uses the new **Rotating Frame** node.

COMPARISON OF CAMPBELL DIAGRAMS USING DIFFERENT ROTOR INTERFACES

The updated model uses the new **Rotating Frame** node and the new mode following functionality for eigenfrequency studies. The example model also showcases the updated **Whirl** and **Campbell Diagram** result templates.

SIMPLY SUPPORTED BEAM ROTOR

The updated model uses the new mode following functionality for eigenfrequency studies. It also showcases the updated **Campbell Diagram** and **Stability Diagram** result templates.

CRITICAL SPEED OF A DUAL ROTOR SYSTEM

The updated model uses the new mode following functionality for eigenfrequency studies. It also showcases the updated **Campbell Diagram** result template.

ROTORS CONNECTED BY A SPLINE COUPLING

The updated model uses the new mode following functionality for eigenfrequency studies. It also showcases the updated **Campbell Diagram** result template.

Backward Compatibility with Version 6.3 and Earlier

NEW ROTATING FRAME FEATURE

In the **Solid Rotor** and **Solid Rotor, Fixed Frame** interfaces, the **Rotor Axis** and **Change Rotor Speed** nodes have been deprecated. When opening a model created in an earlier version of COMSOL Multiphysics, all **Rotor Axis** nodes are automatically replaced by corresponding **Rotating Frame** nodes.

If the model contains any **Change Rotor Speed** nodes with matching domain selections, the settings will automatically be updated in the corresponding **Rotating Frame** node. Any unmatched **Change Rotor Speed** nodes must be updated manually.

For those using the COMSOL API *for* use with Java, manual migration of old models is required. This can be done by replacing old code such as:

```
model.component("comp1").physics("rotsld").prop(RotorProperties")
.set("freqr","1[Hz]");
model.component("comp1").physics("rotsld").prop("SpinSoftening").
set("SpinSoftening",false);
```

with

```
model.component("comp1").physics("rotsld").feature("rotf1")
.set("freqr","1[Hz]");
model.component("comp1").physics("rotsld").feature("rotf1")
.set("includeSpinSoftening",false);
```

In addition, code for setting up the axis of rotation in a **Rotor Axis** node with an edge or point selection such as:

```
model.component("comp1").physics("rotsld").feature("raxi1")
.set("specifiedBy","Edge");
model.component("comp1").physics("rotsld").feature("raxi1")
.feature("axis1").selection().set(1);
```

and

```
model.component("comp1").physics("rotsld").feature("raxi1")
.set("specifiedBy","Supports");
model.component("comp1").physics("rotsld").feature("raxi1")
.set("Location","SelectedPoints");
model.component("comp1").physics("rotsld").feature("raxi1")
.feature("fsup1").selection().set(1,2);
model.component("comp1").physics("rotsld").feature("rotf1")
.feature("ssup1").selection().set(3,4);
```

must be replaced with

```
model.component("comp1").physics("rotsld").feature("rotf1")
.set("axisOfRotation","edgeSelection");
model.component("comp1").physics("rotsld").feature("rotf1")
.selection("axisEdgeSelection").set(1);
```

and

```
model.component("comp1").physics("rotsld").feature("rotf1")
.set("axisOfRotation","pointSelection");
model.component("comp1").physics("rotsld").feature("rotf1")
.selection("axisPointSelection1").set(1,2);
```

```
model.component("comp1").physics("rotsld").feature("rotf1")
.selection("axisPointSelection2").set(3,4);
```

If you do not want to use the new and improved **Rotating Frame** node, the previously default **Rotor Axis** node must be added manually. You can do this by adding the following lines of code:

```
model.component("comp1").physics("rotsld")
.create("raxi1","RotorAxis",3);
model.component("comp1").physics("rotsld").feature("raxi1")
.selection().all();
```

UPDATED LOCAL COORDINATE SYSTEM

For consistency reasons, the local *z* direction is used as the axis of rotation in the **Solid Rotor** and **Solid Rotor, Fixed Frame** interfaces. In most cases, the migration is handled automatically.

However, if a model contains features with manually typed in expressions that rely on the previous local coordinate system, manual migration of the variables is required. One example is a **Journal Bearing** node with the **Bearing model** selected as **Total force and moment**. If the first entry in the **Total force** user input was manually defined as `1[N/m]*rotsld.jrb1.du2 + 2[N/m]*rotsld.jrb1.du3`, the same entry must be updated to `1[N/m]*rotsld.jrb1.du1 + 2[N/m]*rotsld.jrb1.du2`.

Also, if you are using the **Misalignment** node (subnode to, for example, the **Journal Bearing** node), along with COMSOL API *for* use with Java, manual migration of old models is required. This can be done by replacing old code such as:

```
model.component("comp1").physics("rotsld").feature("jrb1")
.feature("mlgn1").set("u0y", "1[um]");
model.component("comp1").physics("rotsld").feature("jrb1")
.feature("mlgn1").set("u0z", "2[um]");
model.component("comp1").physics("rotsld").feature("jrb1")
.feature("mlgn1").set("th0y", "1[mrad]");
model.component("comp1").physics("rotsld").feature("jrb1")
.feature("mlgn1").set("th0z", "2[mrad]");
```

with

```
model.component("comp1").physics("rotsld").feature("jrb1")
.feature("mlgn1").set("u0x", "1[um]");
model.component("comp1").physics("rotsld").feature("jrb1")
.feature("mlgn1").set("u0y", "2[um]");
model.component("comp1").physics("rotsld").feature("jrb1")
.feature("mlgn1").set("th0x", "1[mrad]");
model.component("comp1").physics("rotsld").feature("jrb1")
.feature("mlgn1").set("th0y", "2[mrad]");
```

WHIRL RESULT TEMPLATE

If you were using the default plot **Whirl**, manual migration is needed. You can either delete the existing **Whirl** plot (for example **Whirl (rotsld)**) and add the new **Whirl** result template, or follow these steps:

1. Expand the **Datasets** node under **Results**.
2. Select the **Cut Line 3D** dataset (labeled **Axis**).
3. From the **Line entry method** list, select **Point and direction**.
4. In the **X** text field under **Position**, type `rotsld.rotf1.rbdx`.
5. In the **Y** text field under **Position**, type `rotsld.rotf1.rbdy`.
6. In the **Z** text field under **Position**, type `rotsld.rotf1.rbdz`.
7. In the **X** text field under **Direction**, type `rotsld.rotf1.e3x`.
8. In the **Y** text field under **Direction**, type `rotsld.rotf1.e3y`.
9. In the **Z** text field under **Direction**, type `rotsld.rotf1.e3z`.

Semiconductor Module

New Functionality in Version 6.4

SEMICONDUCTOR ELECTROSTATICS COUPLING MULTIPHYSICS FEATURE

The new **Semiconductor Electrostatics Coupling** multiphysics feature overrides the built-in electrostatics equation defined in the **Semiconductor** interface with the one defined in a separate **Electrostatics** interface. This feature is useful when an advanced electrostatics formulation, such as **Charge Conservation, Ferroelectric**, is needed, making it especially well suited for simulating ferroelectric and piezoelectric semiconductors.

MULTIPHYSICS MODELING OF NOVEL SEMICONDUCTOR ARCHITECTURES

The **Transport of Charge Carriers** interface included in the Semiconductor Module enables seamless integration of customized charge carrier transport models with thermal and electromagnetic simulations.

You can explore this multiphysics modeling workflow in the Memristor tutorial model.

INDIRECT OPTICAL TRANSITIONS MODEL

The **Indirect Optical Transition** feature includes a new model that supports defining the optical transitions directly from optical materials.

New Models in Version 6.4

MEMRISTOR

This model captures the dynamic resistive switching behavior of an oxide-based memristor. The device features a thin metal oxide layer sandwiched between two metal electrodes. When a voltage is applied, oxygen vacancies within the oxide layer migrate, acting as charge carriers and enabling resistive switching.

The model incorporates key physical phenomena, including oxygen vacancy drift–diffusion, current continuity, and heat transfer, all fully coupled through a multiphysics framework. These interactions are essential to accurately reproduce the device's behavior. Simulation results reveal the characteristic pinched hysteresis curve of memristors, closely aligning with experimental data reported in the literature.

METAL–INSULATOR–METAL (MIM) DIODE

This example shows how to model a simple Metal–Insulator–Metal (MIM) diode. The two metal electrodes are defined on each side using a **Metal Contact** feature. Two studies are performed: one without quantum tunneling across the potential barrier and another using the **WKB Tunneling Model** feature to include it. The resulting I–V characteristics are then compared between two studies: with and without tunneling.

SHOCKLEY DIODE

This example shows how to model a simple Shockley diode — a four-layer PNPN semiconductor device. An **Analytic Doping Model** node is used to define the doping profiles for each domain. A time-dependent study is employed to calculate the I–V characteristics of the diode.

TUNNEL DIODE

This tutorial demonstrates how to model the band-to-band tunneling across a p–n junction. The tunneling effect is imitated by defining the **User-Defined Recombination** domain feature, which makes the electrons disappear from the conduction band on the n-side and makes the holes disappear from the valence band on the p-side. The resulting J–V curve under forward bias is derived from the model.

Structural Mechanics Module

New Functionality in Version 6.4

PHYSICS INTERFACES FOR EXPLICIT DYNAMIC ANALYSIS

Two new physics interfaces intended for explicit dynamic analysis have been added. These are Solid Mechanics, Explicit Dynamics and Truss, Explicit Dynamics. In terms of functionality, the new interfaces are similar to the corresponding Solid Mechanics and Truss interfaces but have a set of specializations and some limitations in the available features and options.

The new interfaces contain two new unique domain-level nodes, **Mass Scaling** and **Artificial Viscosity**.

There is also a new dedicated **Explicit Dynamics** study to automate explicit dynamics analyses.

MAGNETOMECHANICS FOR SHELL AND MEMBRANE

Four new multiphysics interfaces are available to model the magnetomechanics effect in shells and membranes:

- Magnetomechanics, Shell
- Magnetomechanics, No Currents, Shell
- Magnetomechanics, Membrane
- Magnetomechanics, No Currents, Membrane

A new multiphysics coupling node, **Magnetomechanics, Boundary**, allows the coupling of the Shell and Membrane interfaces to the Magnetic Fields and Magnetic Fields, No Currents interfaces.

The new multiphysics interfaces require both the AC/DC Module and the Structural Mechanics Module.

GENERAL CONTACT

A new **General Contact** feature is available in the Solid Mechanics interface. This contact formulation is particularly useful when there are many different boundaries that can come into contact with each other. Using this feature, you can avoid setting up many different contact pairs.

Damping and **Friction** subnodes can be added to **General Contact**.

SMOOTH PENALTY FUNCTION FOR CONTACT

A smooth penalty function has been added to the **Contact** and **Interior Contact** nodes. It is also available for **General Contact**. The new option uses a nonlinear (smooth ramp) penalty function for the contact pressure. This may improve the convergence significantly.

LOGARITHMIC STRAINS

The control over the formulation of geometric nonlinearity in the **Geometric Nonlinearity** section of the settings in the **Linear Elastic Material** node has been updated. It now includes the possibility to define an additive decomposition of logarithmic (Hencky) strains. This option can give significant speedups in large strain analysis while giving virtually the same results as using full multiplicative decomposition.

TRANSPORT IN SOLID INTERFACE

There are several improvements in the Transport in Solids interface.

Three new subnodes are available under the **Solid** material model:

- **Thermophoresis**
- **Electromigration**
- **Stress Migration**

Their purpose is to account for specific transport mechanisms. The new flux contributions are showcased in the following models:

- Vacancy Electromigration in IC Interconnect Lines
- Vacancy Electromigration in an Interconnect

A **Periodic Condition** feature has been added.

Serendipity shape functions have been added.

The sign convention for the stoichiometric coefficient in the **Electrode Surface Coupling** node has been changed. It now follows the convention used in the Battery Design Module.

The new sign convention is showcased in the model Heterogeneous Lithium-Ion Battery.

PHASE FIELD IN SOLIDS

New predefined phase field models have been added to the **Phase Field in Solids** interface. The new models are:

- AT1, Ambrosio-Tortorelli type 1
- AT2, Ambrosio-Tortorelli type 2
- PF-CZM, phase-field regularized cohesive zone model

The new functionality is showcased in the model Phase-Field Modeling of Dynamic Crack Branching.

FLUID–STRUCTURE INTERACTION WITH POROELASTIC MATERIALS

The **Free and Porous Media Flow Coupling** node sets the coupling between Laminar Flow and Darcy's Law interfaces. The **Poroelasticity** node sets the coupling between the Darcy's Law and Solid Mechanics interfaces. The **Fluid–Structure Interaction** (FSI) coupling node can now be applied at the interface between free flow domains and porous domains.

The new multiphysics coupling requires the Structural Mechanics Module together with either the Porous Media Flow Module or the Polymer Flow Module.

LINEAR BUCKLING FOR PERIODIC STRUCTURES

The Shell and Solid Mechanics interfaces now support the **Linear Buckling** analysis together with **Floquet Periodicity** and **Cyclic Symmetry**.

The new functionality is shown in the model Buckling Analysis of a Corrugated Conical Shell.

DATASET FOR PERIODIC CONDITIONS

A new predefined sector dataset and plot templates are added when **Cyclic symmetry** is selected in the **Periodic Condition** node.

The new result templates are added for the Solid Mechanics and Shell interfaces.

This functionality is showcased in the models:

- Vibrations of an Impeller
- Eigenfrequency Analysis of a Free Cylinder

PIEZOMAGNETIC MATERIAL

The options **Maximum loss factor** and **Loss factor for compliance sH** have been added to the **Mechanical Damping** subnode under the **Piezomagnetic Material** node.

SPRING–DAMPER

A **Loss factor** damping option is now available in the **Spring–Damper** node.

The **Spring–Damper** node in the Shell interface now includes the possibility to add rotational stiffnesses beside the translational stiffnesses. Viscous damping can also be added to the translational and rotational components.

WARPAGE

It is now possible to compute warpage with respect to a reference plane defined by the user.

There is now support for symmetry and antisymmetry boundaries conditions when computing warpage.

The warping displacement variable, `dispn_warp`, now uses the normal direction from the reference plane rather than the material normal of the selected boundaries.

The following models showcase the new functionality:

- Heating Circuit
- Heating Circuit — Layered Shell Version
- Thermal Stresses in a Layered Plate

INTERCALATION STRAIN

A reference concentration has been added to the **Intercalation Strain** node. The addition allows you to define a strain-free configuration with a nonzero concentration.

EXTERNAL STRAIN

An option named **Volumetric strain** has been added to the **External Strain** feature. Its purpose is to account for external volumetric strains.

EXTERNAL STRESS

In the Shell interface, the ability to enter stress tensor components defined in the spatial frame (deformed configuration) has been added to the **External Stress** node.

POINT MASS

The **Point Mass** node is now also available in the Solid Mechanics interface.

MANUAL SEARCH METHOD IN COUPLINGS

A manual search method has been added to the extrusion operator in some structural multiphysics nodes in order to improve the performance of the coupling.

It can be accessed in the **Advanced** section of the following multiphysics coupling nodes:

- **Solid–Thin Structure Connection**
- **Solid–Beam Connection**
- **Shell–Beam Connection**
- **Layered Shell–Structure Transition**
- **Structure–Pipe Connection**

JOINT FORCES AND MOMENTS

A new **Joint Forces and Moments** section has been added to the settings of the **Fixed Joint** node. The components connected by a **Fixed Joint** are rigidly attached to each other, and the forces or moments required to restrict the relative motion can now be computed.

REACTION FORCES

New variables to compute reaction force and moments are available in the Solid Mechanics and Shell interfaces. It is now possible to directly obtain the reactions for a single boundary condition, both for pointwise and weak constraints.

In the Solid Mechanics interface:

- **Fixed**
- **Prescribed Displacement**
- **Prescribed Velocity**
- **Prescribed Acceleration**
- **Symmetry**
- **Roller**
- **Antisymmetry**

In the Shell interface:

- **Fixed**
- **Pinned**
- **No Rotation**
- **Simply Supported**
- **Prescribed Displacement/Rotation**
- **Prescribed Velocity**

- **Prescribed Acceleration**
- **Symmetry**
- **Symmetry Plane**
- **Antisymmetry**

CONTINUITY

A new checkbox, **Correct for mismatching boundaries**, has been added to the settings in the **Continuity** node. The purpose is to create a more accurate connection for cases when there is a gap or an overlap between the joined boundaries.

REDUCED INTEGRATION

There are new methods available for hourglass stabilization: the **Energy sampling**, **Hessian**, and **Flanagan–Belytschko** methods. The latter method is only available in the Solid Mechanics, Explicit Dynamics interface.

It is now possible to define a different hourglass stabilization method per mesh element type, for instance, when combining hexahedral and prism elements in the same domain.

Reduced integration is now also available in the Truss and Wire interfaces. It is turned off by default.

FASTENERS

- The **Hole Selection** subnode now includes boundary selection type for the fastener hole detection.
- Fasteners are now available in the Solid Mechanics interface, including pretension load to model bolted connections.
- The tolerance inputs in **Hole Selection** subnode have changed to account for the fastener diameter.

RESULT TEMPLATE FOR FASTENERS AND SPOT WELDS

The result template now uses the new **Table Arrow** plot for the normal and shear forces.

The new functionality is showcased in the model Postbuckling Analysis of an Aircraft Fuselage.

TRACTION VARIABLE

The traction variable `phys.Trac` is now always available on external and internal boundaries in the Solid Mechanics interface.

SECTION FORCES

Section forces can now be computed accurately under geometric nonlinearity.

The definitions of the section forces using the traction variable `phys.Trac` based on the nominal stress, have been updated for geometric nonlinear analyses.

CELL PERIODICITY

It is possible to create a homogenized material for a specific parameter value after running a parametric study.

This functionality is available after pressing the **Create Material by Value** button.

TEST MATERIAL

The extra datasets created by the **Test Material** node in the presence of multiple components have been removed.

If a user adds several **Test Material** nodes in the same model, several datasets are created corresponding to the multiple components and physics. Out of these datasets, only two datasets are relevant; the others are empty.

Functions used in the **Test Material** feature must be defined under **Global Definitions**.

QUATERNIONS

The variable for the real part of the quaternion for **Rigid Connector**, **Rigid Material**, and **Attachment** has changed. The internal dependent variable for the quaternion has been modified in order to be initialized with a zero solution.

ADHESION AND DECOHESION

Resetting of adhesion and decohesion was previously not possible. The adhesion condition and damage states are stored as internal degrees of freedom that keep track of the maximum value over time. These variables can now be reset to their initial values.

STRUCTURAL CONNECTIONS

The transformation matrix for the flexible option in the **Solid–Thin Structure Connection** multiphysics node has been improved for geometric nonlinear analyses.

The implementation of the **Solid–Beam Connection** and **Shell–Beam Connection** multiphysics nodes has been improved for geometric nonlinear analyses.

RESULT TEMPLATES FOR PRINCIPAL STRESSES

Two new result templates to plot the maximum principal tensile stress and the maximum principal compressive stress have been added to the Solid Mechanics, Shell, and Plate interfaces. The maximum principal tensile stress template is also available in the Membrane interface.

NEW FRAMEWORK FOR STORING DISSIPATION

A new method to accumulate and store energy dissipation variables is available. The method is based on an implementation that uses state variables. It works with all time integrators, including the Verlet scheme for explicit dynamics.

AVERAGE ROTATION

The internal dependent variable for the quaternion has been modified in order to be safely initialized with a zero solution.

The expression for the computation of the angular velocity vector has been corrected to account for large rotations.

ORTHONORMAL COORDINATE SYSTEMS

Only orthonormal coordinate systems defined in the spatial frame are allowed in the **Gravity**, **Base Excitation**, and **Linearly Accelerated Frame** nodes in all structural interfaces. This implies that such accelerations always must act in a space-fixed direction.

The special local coordinate systems defined by the Shell and Beam interfaces are excluded from these nodes.

RANDOM PARTICULATE COMPOSITE

The actual volume fraction is computed during the creation of particles. This change affects the resulting geometry since a different volume fraction is obtained with the new formulation.

In version 6.3, when the particles are allowed to intersect, the computed volume fraction differs from the prescribed volume fraction. The mismatch depends on the particle diameters and volume fraction, so the analytical formula used for the computation of number of particles has been improved. The new functionality is showcased in the model Heterogeneous Model of a Solid-State Battery Unit Cell.

GYROID PART

The gyroid geometry part has been moved from the COMSOL Multiphysics part library to the Design Module part library.

ROUNDNESS EVALUATION OF A DEFORMED HOLE

During manufacturing processes, or during use, the geometrical shape of structural parts can deviate from what was intended. For instance, a hole can deviate from an ideal circle, and it can be important to measure and control this deviation. The process of obtaining this information is often called roundness (or circularity) error evaluation.

This model showcases different methods to estimate the radius and the center of circles that approximate a deformed hole.

BUCKLING ANALYSIS OF A CORRUGATED CONICAL SHELL

This example illustrates a linear buckling analysis conducted for a corrugated conical shell, utilizing a single sector of the geometry along with cyclic symmetry periodicity edge conditions. The results are compared to those using the full shell geometry.

NONLINEAR FERROMAGNETIC DIAPHRAGM

A magnetic diaphragm is a flexible, thin structure that interacts with magnetic fields to perform mechanical or sensing functions. When subjected to an external magnetic field, the diaphragm deforms due to magnetomechanical interactions, converting magnetic energy into mechanical displacement or, conversely, mechanical deformation into changes in the magnetic field distribution. In this model, the magnetomechanics of a magnetic diaphragm is studied using the Magnetomechanics, Shell multiphysics interface.

BRACKET, EXPLICIT DYNAMICS

This is a tutorial example showing how to perform a drop test using explicit dynamic analysis.

SEISMIC EVENT: SCATTERING OFF A SMALL MOUNTAIN

The propagation of elastic waves in the ground after a seismic event is simulated using a 2D model. The effect of the ground surface topology on the wave propagation is illustrated when an ideal half-space is modified with the presence of a small mountain. The model is a variation of Lamb's problem. The propagation of the elastic waves is modeled using the Solid Mechanics, Explicit Dynamics interface.

OBJECTS FALLING IN A BOX

This example demonstrates how to set up a dynamic simulation that includes several contact interactions, including friction, using a single general contact pair. The dynamics is solved using explicit time stepping.

Backward Compatibility with Version 6.3 and Earlier

QUATERNION VARIABLES

In the **Attachment**, **Average Rotation**, **Rigid Connector**, and **Rigid Material** nodes in 3D, the internal dependent variable for the quaternion has been modified in order to be safely initialized with a zero solution. When opening an old model, this rescaling is done automatically to maintain full compatibility.

When building a model using the API, the new behavior will be used. This is true even if, for example, a Java file created from an older version is used.

If you have manually changed the scale of the DOF in the solver sequence for the quaternion defined by an **Average Rotation** node, you need to replace lines similar to

```
model.sol("sol1").feature("v1").feature("comp1_solid_avgr1_q")
.set("scaleval", 1);
```

with a line similar to

```
model.sol("sol1").feature("v1").feature("comp1_solid_avgr1_qd").s
et("scaleval", 1);
```

For the **Attachment**, **Rigid Connector**, and **Rigid Material** nodes, you only need to update existing code if you have deselected grouping of ODE variables in the physics interface (this is on by default). In that case, you need to replace lines similar to

```
model.sol("sol1").feature("v1").feature("comp1_solid_att1_a")
.set("scaleval", 1);
model.sol("sol1").feature("v1").feature("comp1_solid_a_rig1")
.set("scaleval", 1);
model.sol("sol1").feature("v1").feature("comp1_solid_rd1_a")
.set("scaleval", 1);
```

with lines similar to

```
model.sol("sol1").feature("v1").feature("comp1_solid_att1_a0")
.set("scaleval", 1);
model.sol("sol1").feature("v1").feature("comp1_solid_a0_rig1")
.set("scaleval", 1);
model.sol("sol1").feature("v1").feature("comp1_solid_rd1_a0")
.set("scaleval", 1);
```

FASTENERS

The **Fasteners** node and its associated **Hole Selection** subnode in the Shell interface are now defined as global features. A new fastener hole detection criterion is used to incorporate the fastener diameter as the reference length. This change of criterion requires a slightly different tolerance value. When opening an old model, the conversion is done automatically, and the automatic hole detection provides the same hole selection as in previous versions. You may consider adjusting the diameter tolerance as this is the new filtering criterion.

When building a model using the API, the new behavior will be used. This is true even if, for example, a Java file created from an older version is used. Some older syntax, however, need manual conversion.

The source and destination selections for the **Fasteners** node and **Hole Selection** subnode that were defined with lines similar to

```
model.physics("shell").feature("fst1").selection().set(1);
model.physics("shell").feature("fst1").selection("bndDst")
.set(2);
model.physics("shell").feature("fst1").feature("hsel1")
.selection().all();
```

need to be replaced with

```
model.physics("shell").feature("fst1").selection("fstSrc")
.set(1);
model.physics("shell").feature("fst1").selection("fstDst")
.set(2);
model.physics("shell").feature("fst1").feature("hsel1")
.selection("edgSelSrc").all();
```

To convert the center offset tolerance to the new criterion, change the line

```
model.physics("shell").feature("fst1").feature("hsel1")
.selection("epsilon_c",0.1);
```

to

```
model.physics("shell").feature("fst1").feature("hsel1")
.selection("epsilon_c",0.1/2);
```

Finally, to enable the former radius difference criterion and discard the newly added diameter tolerance criterion, include the following line:

```
model.physics("shell").feature("fst1").feature("hsel1")
.set("hasEpsilon_r",true);
model.physics("shell").feature("fst1").feature("hsel1")
.set("epsilon_d",10);
```

SPRING–DAMPER

In the **Spring–Damper** node, available in the Solid Mechanics, Shell, Plate, Membrane and Truss interfaces, the energy dissipated in the damper is computed only when it is indicated in the **Energy Dissipation** section in the settings of the physics interface.

Subsurface Flow Module

New Functionality in Version 6.4

PERIODIC CONDITION IN DARCY'S LAW INTERFACE

A **Periodic Condition** feature has been added to specify periodicity between two or more boundaries. In the **Periodic Condition** node, is it possible to enter either a **Pressure difference** or a **Mass flow** between source and destination boundaries.

GENERAL

- The **Inlet** boundary condition in the Darcy's Law and Richards' Equation interfaces now has the option to specify a **Standard flow rate** or **Standard flow rate (SCCM)**.
- A new **Power law** permeability model specifies the permeability as a function of porosity.
- A user-defined non-Darcian flow option has been added to specify the resistance coefficient β directly.
- It is now possible to specify a pressure jump across a free-porous boundary within the **Free and Porous Media Flow, Darcy** multiphysics coupling.
- A new **Velocity Streamlines** result template is available for all flow interfaces. Additionally, existing result templates have been updated to improve visual clarity.
- Improved convergence for the Richards' Equation interface.
- The **Fluid** subnode (under the **Porous Medium** node) allows concentration as a **Model Input** to account for concentration-dependent material properties.

New and Updated Models, Apps, and Add-ins in Version 6.4

ARTIFICIAL GROUND FREEZING

This example shows the thermo-hydro-mechanical (THM) effect to model artificial ground freezing. It couples heat transfer, groundwater flow, and solid mechanics to capture ice formation, pore-pressure changes, and ground deformation during a tunnel excavation.

BOREHOLE HEAT EXCHANGER

This example shows how to model an array of borehole heat exchangers (BHEs) for shallow geothermal energy production. The BHEs are simplified as line heat sinks with

a uniform heat extraction rate. The array is embedded into a layered subsurface model with groundwater flow in one of the layers.

PERMEABILITY ESTIMATION

This updated model uses a parametric sweep over different porosities to compute the permeability. In a subsequent step, a Least-Squares Fit function is used to determine the parameters of a power law relationship between porosity and permeability.

Wave Optics Module

New Functionality in Version 6.4

NEW FEATURE TO CALCULATE THE FAR FIELD FROM MODELS WITH TWO DIFFERENT FAR-FIELD DOMAINS

The new **Far-Field Domain, Inhomogeneous** feature and its **Far-Field Calculation** subnode calculate far-field radiation patterns from structures having a substrate with one set of homogeneous material properties and a superstrate with another set of homogeneous material properties. The **Substrate** and **Superstrate** features are defined as subnodes to the **Far-Field Domain, Inhomogeneous** feature.

The new feature is available in 3D in the **Electromagnetic Waves, Frequency Domain** interface. When defined in the physics, it adds a default 2D **Radiation Pattern** plot.

The feature is demonstrated in the *Scatterer on Substrate* model.

DE-EMBEDDING FUNCTIONALITY AVAILABLE FOR THE PORT FEATURE

In the **Port** feature, it is now possible to multiply the S-parameter with a user-defined phase shift. The phase shift is calculated from the **Port** propagation constant and the user-defined offset distance.

NEW VARIABLES FOR CALCULATION OF COUPLING BETWEEN DIFFERENT BOUNDARIES

The **Port** and **Scattering Boundary Condition** features define variables for the input power. In addition, on **Port** and **Scattering Boundary Condition** boundaries and on boundaries toward perfectly matched layers (PMLs), variables for the outgoing waves are defined. For example, there are variables for output power integrated over the feature boundary and the outcoupling efficiency. That is, the ratio between the output power and the input power.

The output power and outcoupling efficiency variables form a hierarchy, where variables defined for feature nodes (with feature scope) are accumulated to variables with physics scope.

For some of the boundaries, like the boundaries toward PMLs, scattering loss variables are also defined.

There are also new variables for calculating the integrated power loss in domains and at boundaries. The variables are defined by the wave equation domain features and by boundary features like the **Transition** and **Impedance** boundary conditions. Also, these variables are hierarchically defined, so it is easy to both find the power loss per feature and the power loss for the whole physics interface. When the input power is defined, there are also similar variables for the absorptance, defined by the ratio of the power loss and the input power.

NEW ELECTRICALLY VERY THIN LAYER OPTION FOR TRANSITION BOUNDARY CONDITION FEATURE

In the **Transition Boundary Condition**, there is a new option in the **Type** parameter called **Electrically very thin layer**. This option represents the case of a very thin layer, as the electric field on the two sides of the boundary are almost the same. Thus, no slit of the dependent variable is used at the boundary.

NEW VARIABLES FOR THE TOTAL ELECTRIC AND MAGNETIC PORT MODE FIELDS

The **Port** feature now defines variables like `ewfd.Emodex_1 = ewfd.tEmodex_1 + ewfd.Enmode*ewfd.nx`, where `ewfd` is the physics tag, and `ewfd.Emodex_1`, `ewfd.tEmodex_1`, and `ewfd.nx` are the x-components of the total electric mode field, the tangential component of the electric mode field, and the port boundary normal, respectively. `ewfd.Enmode` is the normal component of the electric mode field. Similar variables are also defined for the magnetic port mode field.

CHANGED DEFINITION OF TOTAL MODE FIELD AFTER BOUNDARY MODE ANALYSIS

In version 6.3 and earlier, the total **Port** mode field, after performing a **Boundary Mode Analysis** study, was defined as

$$\mathbf{E} = \mathbf{E}_T - E_n\mathbf{n},$$

where $\mathbf{E}_T$ and E_n are the solved for tangential and normal mode field components, respectively, and $\mathbf{n}$ is the port boundary normal, pointing out from the physics.

From this release, the total field is instead formed by

$$\mathbf{E} = \mathbf{E}_T + E_n\mathbf{n}.$$

PML DOMAINS NOT INCLUDED IN DEFAULT FIELD PLOTS

When **Perfectly Matched Layer** (PML) nodes are part of the model, the default field plots define a plot group selection that only includes the non-PML domains. Thereby, only the fields in the non-PML domains are visible in the plots.

PLOTS AT SYMMETRY PLANES ADDED TO DEFAULT FIELD PLOTS

When **Symmetry Plane** features are added, the default field plots replace slice plots parallel to the **Symmetry Plane** boundaries with **Surface** plots of the field at the **Symmetry Plane** boundaries.

NEW NORMALIZATION OPTION TO POLARIZATION PLOTS

The **Polarization** plot now has an option to normalize the polarization ellipse size to each diffraction order (**Normalization** is set to **With respect to current order**) or to the largest diffraction efficiency (**Normalization** is set to **With respect to all orders**). The first option gives the behavior in version 6.3 and earlier, when all polarization ellipses were of the same size. With the second option, the sizes of the polarization ellipses scale with the diffraction efficiency.

In addition, the default polarization plot group adds plots indicating the limit for when the diffraction orders represent propagating waves. These additional plots are enabled when polarization states of higher diffraction orders are plotted.

CROSS SECTION CALCULATION ADDS DEFAULT PLOTS AND EVALUATION GROUPS

When a **Cross Section Calculation** node is part of the model, it automatically generates default plots or evaluation groups, including the cross sections.

The Scatterer on Substrate and Optical Scattering off a Gold Nanosphere models demonstrate this new functionality.

EVALUATION GROUPS ADDED BY DEFAULT INSTEAD OF GLOBAL EVALUATION NODES

By default, in version 6.3 and earlier, when performing a study where there was no parametric sweep, **Global Evaluation** nodes were added for instance for evaluation of reflectance, transmittance, and absorptance. Now, instead those **Global Evaluation** nodes are added to **Evaluation Group** nodes. That means that the evaluated expression values can be automatically updated after a study has been completed.

ELECTRIC POINT DIPOLE POWER VARIABLE ADDED IN 2D

A variable for the power radiated by the dipole has been added to the **Electric Point Dipole** feature in 2D (previously only available in 3D).

New and Updated Models, Apps, and Add-ins in Version 6.4

THERMAL HEATING OF A SEMICONDUCTOR SATURABLE ABSORBER MIRROR (SESAM)

The thermal effects of a laser beam incident on a Semiconductor Saturable Absorber Mirror (SESAM) are simulated under a range of incident beam powers.

A multiphysics simulation is performed of the saturated absorption of the beam on a SESAM, the resultant thermal heating, the structural deformation of the SESAM surface, and the effect this has on the reflected beam.

SILICON ON INSULATOR OPTICAL GRATING COUPLER

This model simulates the coupling of free-space laser radiation into waveguide modes using an optical grating coupler. The coupling efficiency and the radiation loss are calculated for different angles of incidence of the laser beam.

TOPOLOGY OPTIMIZATION OF A METALENS

This model demonstrates how to use topology optimization to design a metalens that focuses a single wavelength to a point. Equation-based modeling is used to implement a manufacturing constraint similar to milling but with a discrete number of layers corresponding to planar processing. The results are verified on another component using an explicit geometry representation.

REFLECTION OF A CIRCULARLY POLARIZED PLANE WAVE

In this model, a plane wave with circular polarization is reflected off an interface between air and glass. The model demonstrates three different ways of defining this problem. The first part shows how to define user-defined port mode fields. The second part uses the predefined circular polarization settings of the **Port** features. Finally, the last part demonstrates how the **Periodic Structure** feature greatly simplifies the modeling process.

DESIGNING A METASURFACE BEAM DEFLECTOR USING SHAPE OPTIMIZATION

This model is based on the existing *Metasurface Beam Deflector* model. The performance is improved using shape optimization by allowing both the radii and the position of the individual pillars to change.

MODELS UPDATED TO DEMONSTRATE THE NEW OUTCOUPLING AND LOSS VARIABLES

The following models have been updated to demonstrate the new outcoupling, absorptance, and scattering loss variables:

- Beam Splitter
- Silicon on Insulator Optical Grating Coupler
- Mach-Zehnder Modulator
- Modeling a Scatterer Near an Optical Waveguide
- Optical Ring Resonator Notch Filter
- Optical Ring Resonator Notch Filter 3D
- Thermal Heating of a Semiconductor Saturable Absorber Mirror (SESAM)
- Single Mode Fiber-to-Fiber Coupling
- Waveguide S-Bend

MODELS UPDATED TO DEMONSTRATE THE UPDATED POLARIZATION PLOT

The Reflection of a Circularly Polarized Plane Wave, Hexagonal Grating, Hexagonal Plasmonic Color Filter, and Metasurface Beam Deflector models have been updated to demonstrate the updated polarization plots.

MODELS MOVED TO THE BEAM PROPAGATION CATEGORY

The following models have been moved from the Optical Scattering to the Beam Propagation category in the Wave Optics Module Application Library:

- Beam Splitter
- Gaussian Beam Incident at the Brewster Angle
- Gaussian Beam Propagation Through an Optical Prism

BAND-GAP ANALYSIS OF A PHOTONIC CRYSTAL

This model has been simplified by using the nonlinear ARPACK eigenvalue solver combined with mode filtering and following.

FRESNEL EQUATIONS

The geometry in this model has been updated to be of micrometer-size, instead of meter-size.

OPTICALLY ANISOTROPIC WAVEGUIDE

This model has been updated to use the mode following functionality available in the **Mode Analysis** study settings. Thereby, dispersion curves can be plotted as solid lines — one for each mode number — when modes are computed as part of a **Parametric Sweep**.

SCATTERER ON SUBSTRATE

This model demonstrates the default plots and evaluation groups added by the new **Far-Field Domain, Inhomogeneous** feature, the physics interfaces and the **Periodic Structure** and **Cross Section Calculation** features.

WHISPERING GALLERY MODE RESONATOR

This model has been updated to demonstrate the eigenfrequency filtering functionality available in the **Eigenfrequency** study settings.

Backward Compatibility with Version 6.3 and Earlier

NEW DEFINITION FOR TOTAL ABSORPTANCE VARIABLE

The variable `ewfd.Atotal`, where `ewfd` is the physics tag, was previously defined to include both loss due to material absorption and outcoupling loss due to radiation that reached ports (of slit type) but did not match the port mode fields. From this release, those two contributions are split into two variables: `ewfd.Atotal`, includes the loss due to absorption in domains and at boundaries, whereas a new variable, `ewfd.Lsca`, accounts for the scattering loss, such as the loss due to radiation not matching the port mode fields. The `ewfd.Lsca` variable also includes the loss due to power flowing into **Perfectly Match Layer** domains.

The `ewfd.Atotal` variable was previously available from the `Ports` category in variable context menus. Now, it is available from the `Heating and losses` category.

DEFAULT VALUE FOR SLIT TYPE CHANGED TO DOMAIN-BACKED

The **Slit Type** parameter in the settings for the **Port** feature has changed from **PEC-backed** to **Domain-backed**. This change has consequences for models generated by API code. If the code previously did not explicitly set the **Slit Type** parameter for the port, the slit type would be PEC-backed. Now, the same code will get the Domain-backed slit type. To get PEC-backed slit type behavior, manually add a Java statement similar to this one

```
model.component("comp1").physics("ewfd").feature("port1")
.set("SlitType", "PECBacked")
```

LiveLink™ for MATLAB®

New Functionality in Version 6.4

MPHGRAIN

New function for extracting data from models using the Granular Flow Module. This function supports models using these datasets: Grain Trajectories, Grain1D, and GrainBin.

`mphplot` supports plotting of data coming from the Granular Flow Module.

MPHSTDDEV

New function for calculating standard deviation for expressions similar to the already existing `mphmean` wrapper function.

MPHMATRIX

When using `mphmatrix` for extracting system matrices for eigenvalue problems, the linearization value `'eigref'` may now be complex valued.

MPHDOC

When opening the Java documentation using `mphdoc`, the help is shown in a separate browser window.

Syntax:

```
mphdoc('api', <modelnode>)
```

MPHWRITEMESH

`mphwritemesh`, which is used for writing meshes as files, now supports second order elements.

MPHSTARTCOMSOLMPHSERVER

`mphstartcomsolmphserver` now supports many more options, enabling it to be used in more scenarios where more control of file locations is required. The new arguments are: `multi, timeout, forcecomsolgcc, forcegcc, mlroot, prefsdir`.

MPHRUN

When using `mphrun` on a solver sequence, the method `runAll` is used. Previous versions used the `run` method. This should not make much difference for most models.

FUNCTIONS WITH MORE OUTPUT

`mphsolinfo`, `mphmeshstats`, and `mphshowerrors` show more output than previous versions. `mphdatasetinfo` has more output for Grid datasets.

LiveLink™ for Excel®

New Functionality in Version 6.4

SUPPORT FOR 32-BIT AND 64-BIT MICROSOFT EXCEL

LiveLink™ for Excel® supports 32-bit as well as 64-bit versions of Microsoft Excel. Note that COMSOL Multiphysics is always using 64-bits. When using a 32-bit version of Excel®, an attempt to extract a large amount of data may result in Excel® running out of memory.

The COMSOL API for Use with Java

COMSOL 6.4 API Changes

CACHE SIZE CONTROL

You can now use the following methods to control the cache size and clear the cache:

`ModelUtil.ClearCache()`, which clears the memory and disk cache.

`ModelUtil.setCacheMaxSize(long fileCache, long memoryCache)`: Set the maximum size used for caching.

`getCacheMaxSizeMemory()` returns, as a long data type, the maximum amount of memory in megabytes (MB) used for caching, for example, geometries, meshes, and rendering data.

`getCacheMaxSizeDisk()` returns, as a long data type, the maximum amount of disk in megabytes (MB) used in the temp area for caching, for example, geometries, meshes, and rendering data.

MEASURING THE ANGLE BETWEEN TWO EDGES

Use the `getEdgeAngle` method when measuring two edges that are either straight or adjacent to get the angle between them.

COMSOL 6.3 API Changes

MATERIAL MODEL API METHODS

The following new methods are available:

- The `setPropertyInfo` method for specifying property information (references, temperature conditions, and so on) for the given material property:

  ```
  MaterialModel setPropertyInfo(String name, String info);
  ```

- The `getPropertyInfo` method for returning property information for the given material property:

  ```
  String getPropertyInfo(String name);
  ```

For the following methods, “parameter” has been changed to “property” in the descriptions for names and group types that are sent to and returned: `getString`,

`getStringArray`, `getType`, `getValueType`, `hasParam`, `setMixingRule`, `size`, and `getMixingRule`.

COMSOL 6.2 API Changes

MESH ENTITY NUMBERS

For meshes that define their own geometric models, such as imported meshes, the entity numbering will change if any of the following features are used in the meshing sequence: `Union`, `IntersectPlane`, `IntersectLine`, `MergeEntities`, `CreateDomains`, `CreateEdges`, `CreateFaces`, `CreateVertices`, or `FillHoles`.

COMSOL 6.1 API Changes

PERIODIC CONDITIONS — DESTINATION SELECTION

The **Destination Selection** subnode under **Periodic Condition** nodes was removed in version 6.1 and replaced by an optional **Destination Selection** section in the **Settings** windows for **Periodic Condition** nodes. It can still be added from the API for backward compatibility. The only difference in behavior for old API code when using Destination Selections is what the entities of the feature's selection are when it is set to all boundaries:

A periodic condition:

```
PhysicsFeature pc = model.physics("ec").feature("pc1");
```

Set the selection to all boundaries of the destination domains feature:

```
pc.feature("dd1").selection().all();
```

The result of this statement typically differs between versions 6.0 and 6.1:

```
int[] entities = pc.feature("dd1").selection().entities();
```

In version 6.0, only the automatic destination entities from the parent's automatic destination selection were returned. In version 6.1, the software returns all entities that overlap with the parent selections (source and destination). The interpretation of the all boundaries flag being set is still the same: making the periodic condition use the automatic destination. The only difference is what the API returns for the selection of the destination domains.

FILLETS AND CHAMFERS IN 2D GEOMETRIES

In 2D models, when adding `Fillet` and `Chamfer` features, you typically do not need to set the `selectinsketch` property, as it is set to `on` by default.

GEOMETRY EXPORT

The following method now sets the file format of the geometry export:

```
model.component(<ctag>).geom(<tag>).export().set<i>Type</i>(<format>);
```

where *<format>* can take the following values: `nativeascii`, `nativebin`, `parasolidbin`, `parasolidascii`, `acisbin`, `acisascii`, `iges`, `step`, `stlbin`, `stlascii`, or `dxf`.

To get the file format that is set for the geometry export, use:

```
String fileType = model.component(<ctag>).geom(<tag>).export().
                  get<i>Type</i>();
```

COMSOL 6.0 API Changes

PAIR FEATURES

The removal of the fallback features under pair features can break any API code that you have written that accesses these features. The presence of a default pair feature may also create a different configuration compared to what previous API runs did.

Java or MATLAB® programs that accessed fallback features under a pair feature will not work anymore. This is a necessary limitation because any attempt to support such API backward compatibility will be both unintuitive and unsafe. For most physics created from the API, the default feature from the physics will act as the fallback to the pair features. As long as the default fallback feature in 5.6 was of the same type, the behavior in 6.0 will be equivalent. If you used a special fallback feature for a pair feature in 5.6, you now place it either before or after the pair feature and use the same selection as the pair feature. Placing the fallback feature afterward makes it possible to use the pair feature's selection directly. The example below shows such code for 5.6 and how the equivalent code in 6.0 can be written:

```
// Creating a pair feature in both 5.6 and 6.0
model.component("comp1").physics("es").create("cont1",
"Continuity", 2);
// Assign some pairs to the pair feature
model.component("comp1").physics("es").feature("cont1").
set("pairs", new String[]{"ap1", "ap2"});
```

```
// Creating a fallback feature under a pair in 5.6 (this will not
work in 6.0)
model.component("comp1").physics("es").feature("cont1").
create("sfcd1", "SurfaceChargeDensity", 2);
// Accessing a settings in 5.6 (this will not work in 6.0)
model.component("comp1").physics("es").feature("cont1").
feature("sfcd1").set("rhoqs", "1e-9");

// Creating an equivalent feature in 6.0 that act as fallback to
the pair feature
model.component("comp1").physics("es").create("sfcd1",
"SurfaceChargeDensity", 2);
// Use the same selection as the pair feature
int[] ent = model.component("comp1").physics("es").
feature("cont1").selection().entities();
model.component("comp1").physics("es").feature("sfcd1").
selection().set(ent);
// Accessing a settings in 6.0
model.component("comp1").physics("es").feature("sfcd1").
set("rhoqs", "1e-9");
```

When a physics interface is created from the API in 6.0, it will also add default pair features that did not exist in 5.6. As long as the final model uses all created pairs, the behavior will be the same in 6.0; otherwise, it may be necessary to disconnect the default pair feature with an extra command. Below is an example for the Electric Currents interface, but the actual set operation is identical for all interfaces.

```
model.component("comp1").physics("ec").feature("dcont1").
set("pairDisconnect", true);
```

This concludes the release notes for COMSOL Multiphysics version 6.4.

Index