App 开发器

简介

6.1

COMSOL

App 开发器简介

© 1998–2017 COMSOL 版权所有

受列于 cn.comsol.com/patents 的美国专利 7,519,518、 7,596,474、 7,623,991、 8,457,932、 8,954,302、 9,098,106、 9,146,652、 9,323,503、 9,372,673、 9,454,625 和 10,019,544 保护。专利申请中。

本文档和本文所述的程序根据 《COMSOL 软件许可协议》 (cn.comsol.com/comsol-license-agreement) 提供，且仅能按照许可协议的条款进行使用或复制。

COMSOL、COMSOL 徽标、COMSOL Multiphysics、COMSOL Desktop、COMSOL Compiler、COMSOL Server 和 LiveLink 为 COMSOL AB 的注册商标或商标。所有其他商标均为其各自所有者的财产，COMSOL AB 及其子公司和产品不与上述商标所有者相关联，亦不由其担保、赞助或支持。相关商标所有者的列表请参见 cn.comsol.com/trademarks。

版本：COMSOL 5.6

联系信息

请访问 “联系我们”页面 cn.comsol.com/contact，以提交一般查询、联系技术支持或搜索我们的联系地址和电话号码。您也可以访问全球销售办事处页面 cn.comsol.com/contact/offices，获取更多地址和联系信息。

如需联系技术支持，请访问 COMSOL Access 页面 cn.comsol.com/support/case，创建并提交在线请求表单。其他常用链接包括：

- 技术支持中心：cn.comsol.com/support
- 产品下载：cn.comsol.com/product-download
- 产品更新：cn.comsol.com/support/updates
- COMSOL 博客：cn.comsol.com/blogs
- 用户论坛：cn.comsol.com/community
- 活动：cn.comsol.com/events
- COMSOL 视频中心：cn.comsol.com/video
- 技术支持知识库：cn.comsol.com/support/knowledgebase

文档编号：CM020011

目录

前言 9
简介 10
App 开发器桌面环境 12
App 开发器和模型开发器 19
参数、变量和范围 20
运行 App 22
在 COMSOL Multiphysics 中运行 App 22
通过 COMSOL Server 运行 App 30
编译和运行已编译的 App 35
发布 COMSOL App 39
App 开发器入门 41
表单编辑器 46
表单设置窗口 46
单独的表单设置窗口 47
表单编辑器首选项 49
表单对象 50
表单编辑器中的编辑器工具 55
按钮 57
图形 69
输入框 85
单位 92
文本标签 92
数据显示 93
表单编辑器中的数据访问 96
草图和栅格布局 102

在 App 之间进行复制.....116
在“模型开发器”中使用表单.....117
输入.....119
主窗口.....122
菜单栏和工具栏.....124
功能区.....128
事件.....129
启动和停止运行时的事件.....130
全局事件.....131
表单和表单对象事件.....134
使用局部方法.....135
声明.....136
标量.....140
一维数组.....143
二维数组.....145
选择列表.....147
文件.....149
单位集.....150
快捷方式.....155
图形数据.....157
方法编辑器.....161
将命令序列转换成方法.....161
语言元素窗口.....166
方法编辑器中的编辑器工具.....166
方法编辑器中的数据访问.....169
录制代码.....171
检查语法.....174

查找和替换 175
模型表达式窗口 176
使用快捷方式 177
语法突出显示、代码折叠和缩进 178
方法编辑器首选项 180
使用 Ctrl+/ 与 Tab 补全代码 181
创建局部变量 183
局部方法 184
带输入和输出变元的方法 186
调试 188
停止方法 190
模型对象 190
语言元素示例 190
在模型开发器中运行方法 194
库. 201
图像 201
声音 204
文件 206
附录 A - 表单对象 207
表单对象列表 207
切换按钮 208
复选框 211
组合框 215
方程 235
线 236
网页 237
图像 238
视频 238

进度条 240
日志 243
消息日志 244
结果表 245
表单 247
表单集合 249
卡片堆 251
文件导入 255
信息卡片堆 259
数组输入 262
单选按钮 266
选择输入 268
文本 272
列表框 273
表格 277
滑块 282
超链接 284
工具栏 286
隔片 287
附录 B - 在 App 之间进行复制 289
附录 C - 文件处理和文件方案语法. 290
使用 COMSOL Server 处理文件 290
文件方案语法 293
文件导入 295
文件导出 303
附录 D - 键盘快捷键. 311
附录 E - 内置方法库 313

附录 F - App 开发指南 329
附录 G - 案例库示例 332

前言

仿真软件的用户通常是拥有硕士或博士学位的专业人员，他们具有多年的建模仿真经验，并针对特定软件的使用接受过全面的培训。这些专业人员通常是大型机构研发部门的科研人员或是学术研究人员。由于仿真理论非常复杂，典型的仿真软件包含许多选项，需要用户利用自己的专业知识来验证模型和仿真。

这意味着少数仿真专业人员需要为一个较大的群体服务，后者主要是从事产品开发、生产的工作人员或是研究物理效应的学生。仿真模型常常非常复杂，能够实现模型的人员只能是可以安全提供输入数据来获取有用输出的人。因此，使用计算机进行建模和仿真在产品开发、生产和教育方面产生了瓶颈。

为了使这一小部分专业人员能够为更大的用户群体提供服务，“App 开发器”提供了相应的解决方案。借助 “App 开发器”，仿真专业人员可以为原本的通用计算机模型 （随时可用的 App）创建直观且非常独特的用户界面。通用模型可以作为多种不同 App 的起点，每个 App 都为用户提供了只针对要处理的特定任务的输入和输出选项。 App 可以包含用户文档、“在允许范围内的输入”检查以及单击按钮时生成的预定义报告。

创建一个 App 通常需要以下领域的专业人员协同合作：物理学、数值分析、编程、用户界面设计和图形设计。

COMSOL 技术支持团队可以针对您的 App 的物理场和数值分析设置在合理范围内提供一些建议。此外， COMSOL 文档和在线资源也能为您提供很大的帮助。在编程和设计方面，技术支持团队能够提供的帮助有限。在这些领域，您自己的开发工作至关重要。

“App 开发器”能帮助您的团队轻松创建设计优良的 App，从而避免出现意外的用户输入错误，并同时专注于相关的输出详细信息。

COMSOL 确信这是在世界范围内普及成功应用仿真的有效手段，我们将完全致力于帮助实现这一目标。

简介

COMSOL® App 是通过高度专业化的用户界面与 COMSOL Multiphysics® 模型进行交互的一种直观而有效的方式。本书包含 “App 开发器”桌面环境的简要概述，并通过多个示例演示了 “表单编辑器”和 “方法编辑器”的使用方法。还包含相关参考资料，其中特别列出了可用的内置方法和函数。有关如何使用 “模型开发器”的详细信息，请参阅 *《COMSOL Multiphysics 简介》*一书。

> 如果您想要在阅读本书之前先查看 App 示例，可以从任一个 App 文件夹中的 “案例库”打开并研究一个 App。在阅读本书时保持打开该 App，以进行各种尝试。仅 applications 文件夹包含带用户界面的 App，“案例库”中的其他文件夹包含的是不带用户界面的教学案例。

“App 开发器”在 Windows® 版本的 COMSOL Multiphysics 中提供，可以从 COMSOL Desktop® 环境访问。 COMSOL Multiphysics 及其附加产品用于创建 App。需要相应附加产品的许可证才能从 COMSOL Multiphysics 或 COMSOL Server™ 产品运行 App。

如需视频教程等其他资源，可通过以下链接在线获取：cn.comsol.com。

通过 COMSOL MULTIPHYSICS 运行 APP

如果您拥有 COMSOL Multiphysics 许可证，则可在 Windows®、 macOS 及 Linux® 系统的 COMSOL Desktop 中运行 App。

运行编译的 APP

通过使用 COMSOL Compiler™，您可以将 App 编译成支持 Windows®、Linux® 和 macOS 操作系统的可执行文件，并将编译后的可执行文件自由分发给其他人使用，他们无需任何许可证文件就能运行这些可执行文件。

通过 COMSOL SERVER 运行 APP

如果您拥有 COMSOL Server 许可证，则可在 Windows®、 macOS、 iOS、 Linux® 和 Android™ 等平台的主流 Web 浏览器中运行 Web 形式的 App。在 Windows® 系统中，还可以通过易于安装的 COMSOL Client （可从 cn.comsol.com 下载）连接到 COMSOL Server 来运行 COMSOL App。COMSOL Server 不包含 “App 开发器”、“物理场开发器”或 “模型开发器”工具，这些工具随 COMSOL Desktop 环境提供。

APP 开发指南

如果您没有图形用户界面开发或编程经验，建议您先阅读第 329 页的 “附录 F - App 开发指南”。

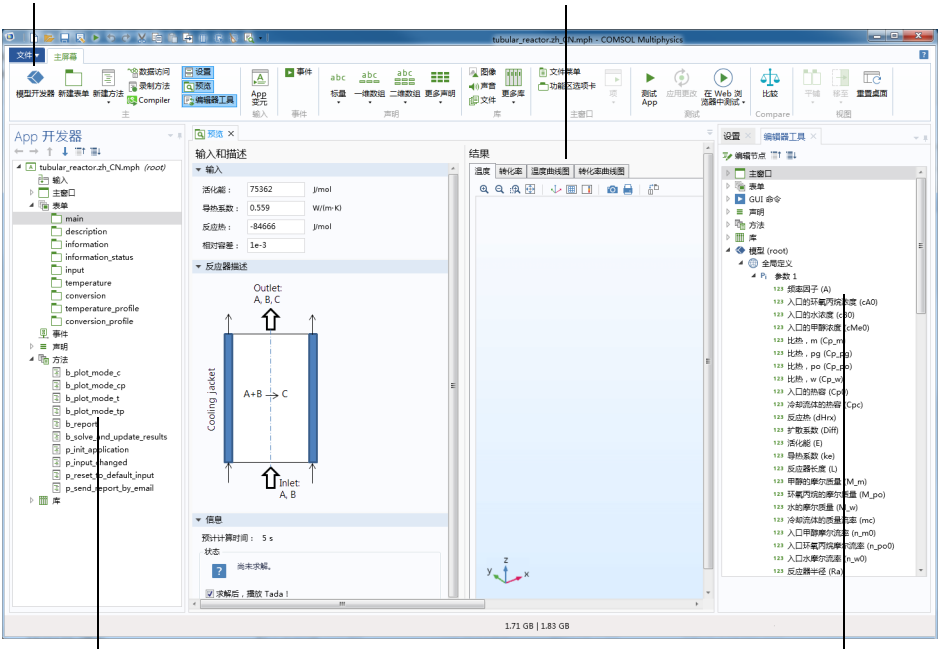
附加文档

有关构建 App 的信息，请参阅以下书籍中的附加文档：Application Programming Guide、 Application Builder Reference Manual 和 Programming Reference Manual。

App 开发器桌面环境

“模型开发器”和“App 开发器”- 通过单击此按钮，可以在“模型开发器”与“App 开发器”之间进行切换。

COMSOL DESKTOP 环境 - COMSOL Desktop 环境支持访问“App 开发器”（包括“表单编辑器”与“方法编辑器”）和“模型开发器”。



“App 开发器”窗口 - 含 App 树的“App 开发器”窗口。

“设置”窗口 - 单击 App 树中的任意节点（包括表单对象或方法节点），查看其关联的“设置”窗口。

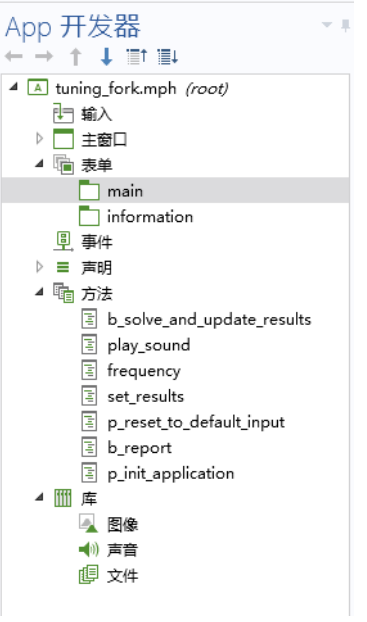
以上屏幕截图表示您在使用“App 开发器”时将看到的界面。“App 开发器”桌面环境的主要组成部分包括：

- “App 开发器”窗口和功能区选项卡
- COMSOL Desktop 环境
- 表单编辑器 （参见第 46 页）
- 方法编辑器 （参见第 161 页）

APP 树

App 树由以下节点组成：

- **输入**
- **主窗口**
- **表单**
- **事件**
- **声明**
- **方法**
- **库**



输入节点下的子节点属于 **App 变元**类型。当您从操作系统的命令行启动 App 时，可以将它们用作该 App 的输入变元。

主窗口节点表示 App 的主窗口，也是用户界面的顶级节点。它包含窗口布局、主菜单明细以及可选的功能区明细。

表单节点包含表单作为子节点。每个表单都可以包含多个表单对象，如输入框、图形对象及按钮。

事件节点包含全局事件作为子节点。这些子节点包含在更改各种数据实体 （如全局参数或字符串变量）时触发的所有事件。全局事件还可以同 App 的启动和关闭操作关联起来。

声明节点用于声明全局变量，除模型中定义的全局参数和变量外，还可以使用这些变量。

方法节点包含各种方法作为子节点。这些方法中的代码可用于执行 “模型开发器”中模型树节点的标准运行命令以外的操作。例如，方法可以执行循环、处理输入和输出以及向 App 用户发送消息和通知等。方法还可以修改正在运行的 App 的模型对象，或当前会话中由 “模型开发器”表示的模型对象。

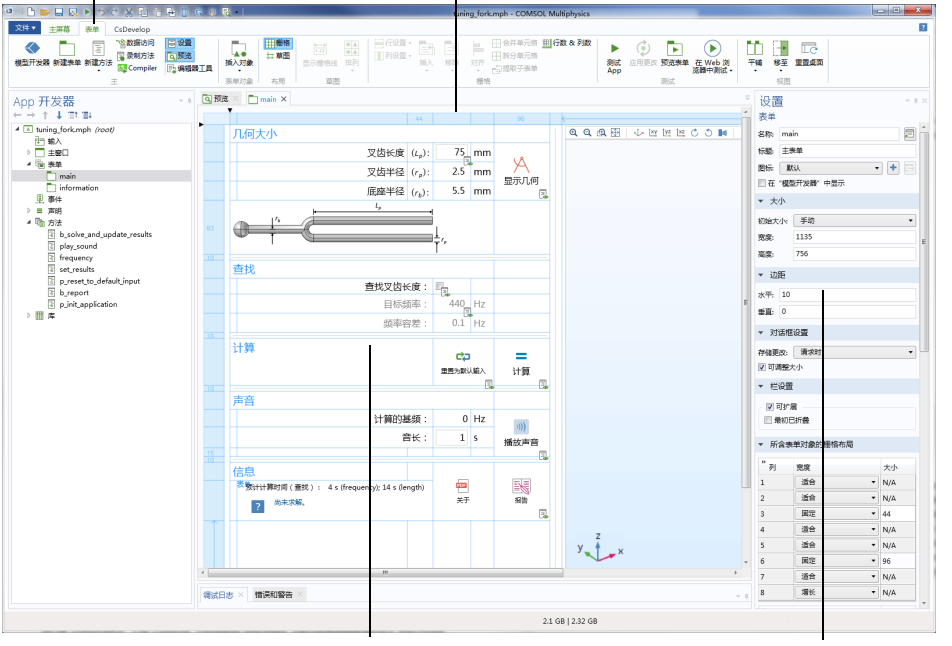
之前的版本提供两种不同类型的方法，具体取决于修改的是正在运行的 App 的模型对象，还是当前会话中由 “模型开发器” 表示的模型对象。新版本不再需要对此进行区分，方法现在支持修改这两种模型对象。

库节点包含要嵌入到 MPH 文件的图像、声音及文件，使您不必将其随 App 一起分发。此外，**库**节点还可以包含 Java® 实用程序类节点以及外部 Java® 和 C 库的节点。

表单编辑器

表单选项卡 - 功能区中的 “表单” 选项卡可供您轻松访问 “表单编辑器”。

“表单编辑器” 窗口 - 选项卡式的 “表单编辑器” 窗口可供您通过拖动来移动对象。单击对象以编辑其设置。



表单对象 - 每个表单都包含表单对象，如输入框、复选框、图形、图像、按钮等。

“设置” 和 “编辑器工具” 窗口 - 单击任意 App 树节点或表单对象以查看其关联的 “设置” 窗口。“编辑器工具” 窗口用于快速创建表单对象。

使用 “表单编辑器” 中的表单对象 （如输入框、图形和按钮）来创建表单，以设计用户界面布局。

“表单编辑器”的主要组成部分包括：

- “表单”功能区选项卡
- 含 App 树的 “App 开发器”窗口
- “表单”窗口
- “编辑器工具”窗口
- “设置”窗口

新建表单

要创建新表单，右键单击 App 树中的**表单**节点并选择**新建表单**。也可以在功能区中单击**新建表单**。在创建新表单时将自动打开**新建表单**向导。

如果您的 App 已包含一个表单，例如 **form1**，并且您想要编辑它，可通过以下两种方式来打开 “表单编辑器”：

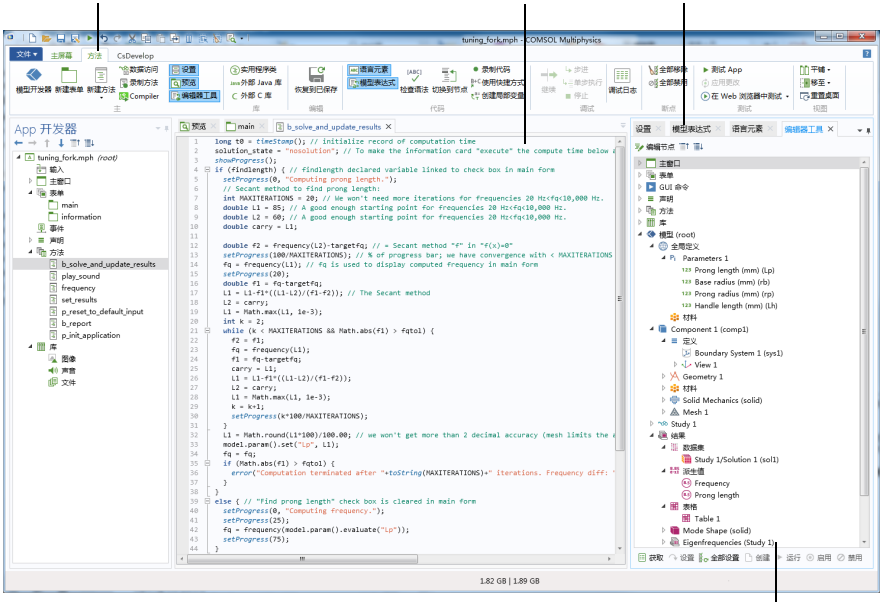
- 在 App 树中双击 **form1** 节点。
- 在 App 树中，右键单击 **form1** 节点并选择**编辑**。

方法编辑器

“方法”选项卡 - 功能区中的“方法”选项卡可供您轻松访问用于编写和调试代码的工具。

“方法”窗口 - 选项卡式的“方法”窗口可供您在不同的方法编辑窗口之间切换。

“设置”窗口 - 单击任意 App 树节点以查看其关联的“设置”窗口。



“模型表达式”、“语言元素”和“编辑器工具”窗口 - 这些窗口显示编写代码的工具。“模型表达式”窗口显示模型中可用的所有常数、参数、变量及函数。“语言元素”窗口用于为内置方法插入模板代码。“编辑器工具”窗口用于提取代码以编辑和运行模型树节点。

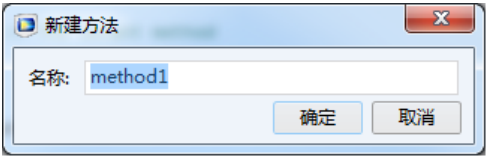
使用“方法编辑器”可以为模型树节点标准用法未涵盖的操作编写方法。“方法”是其他编程语言中所谓的子例程、函数或过程等名称的另一种说法。

“方法编辑器”的主要组成部分包括：

- “方法”功能区选项卡
- 含 App 树的“App 开发器”窗口
- “方法”窗口
- “模型表达式”、“语言元素”、“编辑器工具”和“设置”窗口（这些窗口在上图中堆叠在一起）

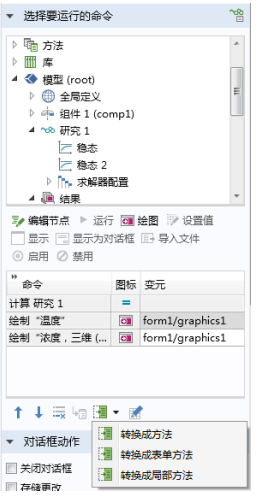
新建方法

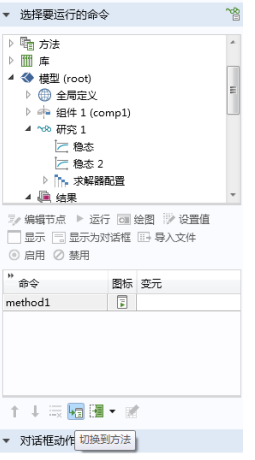
右键单击 App 树中的**方法**节点并选择**新建方法**，可以创建新方法。您也可以在功能区中单击**新建方法**。您可以在**新建方法**对话框中更改方法的名称。



在创建新方法时将自动打开 “方法编辑器”。通过这种方式创建的方法是全局方法，可以从所有方法、表单对象以及 “模型开发器” 功能区的 “开发工具” 选项卡进行访问。

与按钮或菜单项等关联的命令序列可以通过单击**转换成方法**自动转换成新方法。通过单击**切换到方法**可以打开这一新方法。此外，您也可以分别单击**转换成表单方法**或**转换成局部方法**来创建表单或表单对象的局部方法。这些选项在下图中显示。



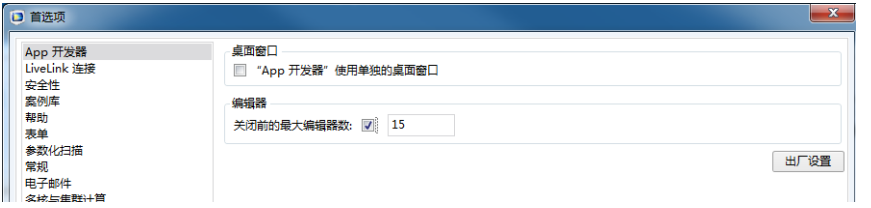


如果方法已存在，假设名为 `method1`，您可以通过以下任一方式来打开 “方法编辑器”：

- 在 App 树中双击 **method1** 节点。
- 在 App 树中，右键单击 **method1** 节点并选择**编辑**。
- 在表单对象或事件**设置**窗口的命令序列下方，单击**切换到方法**。

APP 开发器首选项

要访问 “App 开发器” 的**首选项**，可以从**文件菜单**中选择**首选项**，然后选择 **App 开发器**页面。



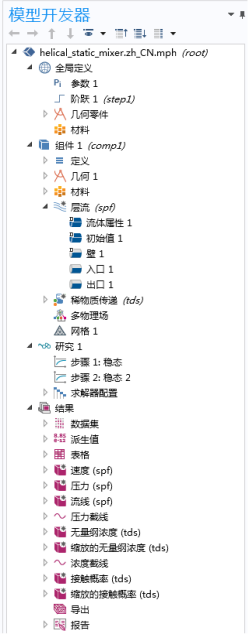
您可以配置 COMSOL Desktop 环境，使 “App 开发器” 显示在单独的桌面窗口中。选中 **“App 开发器” 使用单独的桌面窗口**复选框。

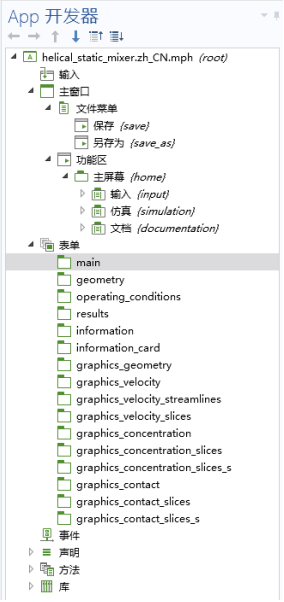
您可以分别使用键盘快捷键 Ctrl+Alt+M 和 Ctrl+Alt+A 在 “模型开发器” 与 “App 开发器” 之间切换。

您可以设置打开的 “表单编辑器” 或 “方法编辑器” 窗口选项卡的数量上限。选中**关闭前的最大编辑器数**复选框并编辑数量 （默认为 15）。保持较少的数量可加快包含大量表单的 App 的加载速度。

App 开发器和模型开发器

您可以使用 “App 开发器” 基于 “模型开发器” 构建的模型来创建 App。其中提供两个重要的 App 创建工具：“表单编辑器” 和 “方法编辑器”。此外，App 还可以包含菜单栏或功能区。“表单编辑器” 包含用户界面组件 （如输入框、图形对象及按钮）的拖放功能。“方法编辑器” 是一个编程环境，可供您修改表示不同模型部分的数据结构。下图显示 “模型开发器” 和 “App 开发器” 窗口。





在创建 App 时，您通常使用现有模型开始操作。然而，您也可以同时构建 App 用户界面和底层模型。您可以随时在 “模型开发器” 与 “App 开发器” 之间轻松进行切换。 App 的模型部分通过模型树来表示，有时称为嵌入模型。

“App 开发器” 中的工具可以通过多种方式来访问并操控嵌入模型中的设置；例如：

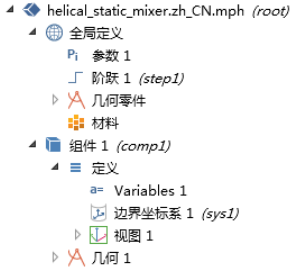
- 如果模型使用参数和变量，您可以使用 “新建表单” 向导或 “编辑器工具” 将其直接链接到输入框。通过这种方式， App 用户可以直接编辑影响模型的参数和变量的值。有关更多信息，请参见第 55 页和 85 页。
- 通过使用 “新建表单” 向导或 “编辑器工具”，您可以在 App 中包含可运行研究节点的按钮，从而启动求解器。此外，您还可以使用此向导来加入

图形、数值输出、复选框及组合框。有关更多信息，请参见第 41 页和 55 页。

- “数据访问”工具和 “编辑器工具”窗口可用于直接访问与表单对象结合使用的模型中或方法中的低级别设置。有关更多信息，请参见第 55、96 和 166 页。
- 通过使用 “录制代码”工具，您可以录制在模型树及其节点中执行操作时要执行的命令。这些命令将加入方法中供您进一步编辑。有关更多信息，请参见第 171 页。

参数、变量和范围

模型树可以包含用于控制模型设置的参数和变量。下图显示的 App 模型树包含**参数**和**变量**节点。



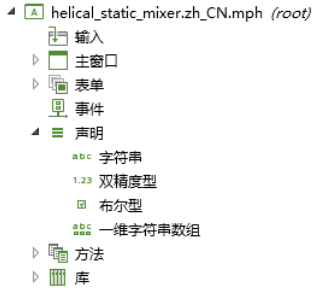
参数是用户定义的常数标量，在模型树的**全局定义**节点下进行定义，可用于整个 “模型开发器”。也就是说，这些参数具有 “全局”性质。其重要用途包括：

- 参数化几何尺寸
- 指定网格单元大小
- 定义参数化扫描

变量可以在**全局定义**节点中定义，也可在任意模型**组件**节点的**定义**子节点中定义。全局定义的变量可用于整个模型，而模型组件变量只能用于该组件。变量可用于空间表达式或时变表达式，包括正在求解的相关场变量。

在 “模型开发器”中，参数或变量是一个字符串，附加限制是它的值为有效的模型表达式。有关如何使用模型中的参数和变量的更多信息，请参阅*《COMSOL Multiphysics 简介》一书。*

App 可能需要在 “表单编辑器”和 “方法编辑器”中使用附加变量。此类变量在 “App 开发器”中 App 树的**声明**节点下声明。下图显示了包含多个声明的 App 树。



“App 开发器”中的声明变量包含以下几种类型：标量、数组、布尔变量、字符串、整数及双精度数。在使用变量之前，必须先声明其类型。

事实上，这些分类变量意味着它们可以直接在方法中使用，无需先使用内置方法进行转换。与表示模型表达式的参数和变量相比，这些分类变量能使代码编写更加容易。尽管如此，“App 开发器”仍提供了多种工具，可用于在不同类型的变量之间进行转换。有关更多信息，请参见第 136 页和第 313 页。有关分类变量的更多信息，请参阅 Application Programming Guide。

运行 App

如果您拥有 COMSOL Multiphysics 许可证，则可在 COMSOL Desktop 环境中运行 App。如果您拥有 COMSOL Server 许可证，便可在各种操作系统和硬件平台的主流 Web 浏览器中运行 App。此外，还可通过易于安装的 Windows® 版 COMSOL Client 连接到 COMSOL Server 来运行 App。

通过使用 COMSOL Compiler™，您可以将 App 编译成可在 Windows®、Linux® 和 macOS 操作系统上运行的可执行文件。

以下两节内容阐述如何使用这些不同的设置来运行 App。第三节 —— 第 39 页的 “发布 COMSOL App” —— 描述 App 的发布权限。

在 COMSOL Multiphysics 中运行 App

在 COMSOL Multiphysics 中，您可以使用以下任意方式来运行 App：

- 在功能区或 “快速访问工具栏” 中单击**测试 App**。
- 在**文件**菜单或 “快速访问工具栏” 中选择**运行 App**。
- 在 Windows® 桌面双击 MPH 文件图标。
- 在功能区中选择**在 Web 浏览器中测试**。

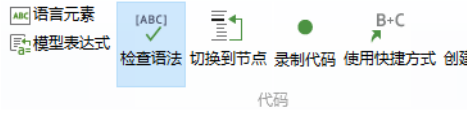
测试 App

测试 App 用于快速测试。此选项会打开一个单独的窗口，显示 App 用户界面，同时保持 “App 开发器” 桌面环境运行。

在测试 App 时，您可以通过单击**应用更改**按钮在运行时更改表单、方法及嵌入模型。但在运行时无法应用所有更改，在这种情况下，系统会提示您关闭 App 并再次单击**测试 App**。

要在不运行 App 的情况下预览表单布局，可以单击功能区中的**预览表单**。

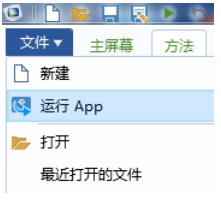
在使用**测试 App** 时，所有方法都可通过内置的 Java® 编译器自动进行编译。出现任何语法错误时都会生成错误消息，并会停止测试 App 进程。要在测试 App 之前检查语法错误，可以单击**方法**选项卡中的**检查语法**按钮。



检查语法可以使用内置的 Java® 编译器来编译方法，从而找到语法错误。此时检测到的任何语法错误都会显示在 “方法编辑器” 的**错误和警告**窗口中。有关更多信息，请参见第 161 页的 “方法编辑器”。

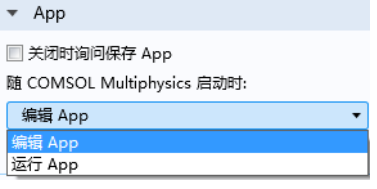
运行 App

运行 App 选项可以在 COMSOL Desktop 环境中启动 App。选择**运行 App** 可将 App 用于生产用途。例如，您可以运行其他人创建的 App，它可以受到编辑密码保护，但不限制您运行。



双击 MPH 文件

只要某个 MPH 文件扩展名与 COMSOL Multiphysics 关联，当您在 Windows® 桌面双击该 MPH 文件图标时，即可在 COMSOL Multiphysics 中打开 App。该 App 打开后，您可以编辑或运行它。您可以在 App 树的根节点控制这一行为。此节点的**设置**窗口包含一个名为 **App** 的栏，您可以在其中选择是**编辑 App** 还是**运行 App**。对此设置所做的更改将在保存 MPH 文件时应用。



编辑 App 选项将在 “App 开发器”中打开 App。

运行 App 选项将在运行时模式下打开 App，用于生产目的。此选项与在**文件**菜单中选择**运行 App** 类似，不同之处在于，双击 MPH 文件后会启动新的 COMSOL Multiphysics 会话。

如果您安装了 Windows® 版 COMSOL Client，则 MPH 文件扩展名会与 COMSOL Client 关联，当您双击 MPH 文件时，系统会提示您登录 COMSOL Server 安装页面。

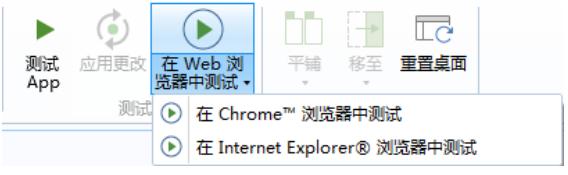
忽略许可证错误

在 App 树根节点的**设置**窗口中，**启动时忽略许可证错误**复选框用于控制在运行 App 时许可产品的行为。

选中该复选框后，即使所需的任何许可证都不可用也能启动 App。但仍无法使用缺少许可证的产品的功能。不过，您可以编写方法来创建 App，以使所用的功能动态适应可用的许可证类型。

在 WEB 浏览器中测试 APP

在 Web 浏览器中测试用于在 Web 浏览器中测试 App。此功能可以方便地测试从连接到 COMSOL Server 的 Web 浏览器访问 App 时的界面外观。



您可以选择在安装的哪个 Web 浏览器中启动 App。**在 Web 浏览器中测试**选项会打开一个单独的窗口，显示 App 用户界面，同时保持 “App 开发器”桌面环境运行。

测试 APP VS. 在 WEB 浏览器中测试

测试 App 启动的 App 用户界面基于 Microsoft®. NET Framework 组件，而**在 Web 浏览器中测试**启动的 App 用户界面则基于 HTML5 组件。**测试 App** 将显示使用 COMSOL Multiphysics 或 COMSOL Server 运行 App 时出现的用户界面，前提是 Windows® 版 COMSOL Client 用于连接安装的 COMSOL Server。**在 Web 浏览器中测试**将显示在使用 COMSOL Server 运行 App 时出现的用户界面，前提是 Web 浏览器用于连接安装的 COMSOL Server。

要在 macOS、iOS、Linux® 和 Android™ 系统的 Web 浏览器中测试 App 用户界面的外观和功能，需要安装 COMSOL Server。

下表汇总了用于运行 App 的不同选项。

服务器软件	客户端软件工具或组件
COMSOL Multiphysics	测试 App
COMSOL Multiphysics	在 Web 浏览器中测试
COMSOL Multiphysics	运行 App
COMSOL Server	Windows® 版 COMSOL Client
COMSOL Server	Web 浏览器
N/A	使用 COMSOL Compiler 编译的可执行文件

“服务器”列代表执行 CPU 占用较多的计算的软件组件。“客户端”列代表用于呈现 App 用户界面的软件工具或组件。对于可执行文件，所有计算都在本地完成。有关已编译 App 的更多信息，请参见第 35 页的 “编译和运行已编译的 App”。

保存正在运行的 APP

当您测试 App 时，系统会为它指派名称 `Untitled.mph`，这是原始 MPH 文件的副本。这与运行 App 的情况不同。

默认情况下，当用户退出 App 时，系统不会提示是否保存更改。您可以在 App 树的根节点控制这一行为。此节点的**设置**窗口包含一个名为 **App** 的栏，您可以在其中选择**关闭时询问保存 App** 复选框，如下图所示。

作为备选方案，您可以添加一个包含 App 保存命令的按钮或菜单项。有关更多信息，请参见第 127 页。

案例库

从**文件**菜单中选择**案例库**以运行和探索 COMSOL 安装中包含的示例 App。本书中的许多屏幕截图都来自这些示例。

您可以单击 “案例库” 树中对应的按钮来运行 App，或将其打开进行编辑。

只包含模型但没有额外表单或方法的 App 不能运行，只能打开用于编辑。包含表单和方法的 App 收集在名为 **Applications** 的文件夹中。

“案例库” 中的 App 会不断完善和更新。您可以通过单击 “更新 COMSOL 案例库” 来进行更新。

还有一些 App 不包含在 “案例库” 中，您可以在 COMSOL 网站的 “案例下载” 页面找到它们。要查找这些 App，单击 “案例下载” 按钮。此时将打开一个浏览器，显示 “案例下载” 网页。

每个 App 都有一个关联的缩略图，显示在 “案例库” 中。在 COMSOL Server Web 界面，该缩略图显示在 “案例库” 页面中。

要设置缩略图，单击 App 树的根节点。**设置**窗口包含两个选项用于选择图像：**从 “图形” 窗口设置**和**从文件加载**。您也可以**清除**图像。

从文件加载选项支持加载 PNG 或 JPG 文件格式的图像。请选择大小为 280*210 到 1024*768 像素的图像，以确保图像在 COMSOL Multiphysics 和 COMSOL Server 中显示为正确的缩略图。

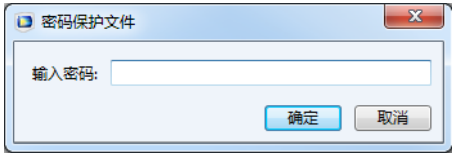
从“图形”窗口设置选项可以自动创建两个缩略图：

- 大小为 280*210 像素的图像，显示在 App 树根节点的**设置**窗口和“案例库”中。
- 大小为 1024*768 像素的图像，在 COMSOL Server 的报告和“案例库”中用作默认标题页面图像。

密码保护

App 可以通过密码保护来管理权限。您可以通过单击 “App 开发器” 窗口中 App 树的根节点，在**设置**窗口中分别为编辑和运行操作指派单独的密码。您必须拥有编辑 App 的权限才能创建运行它的密码。

当您打开受密码保护的 MPH 文件准备编辑或运行时，会出现一个对话框，提示您输入密码：



要移除密码保护，可以创建空密码。

密码保护用来加密所有模型和 App 设置，包括方法。不过，二进制数据不会加密，例如包括嵌入 CAD 文件在内的定型几何、网格数据以及解数据。

安全设置

在使用 “App 开发器” 创建 App 时，考虑托管 App 的计算机的安全性非常重要。COMSOL Multiphysics 和 COMSOL Server 都提供一组相近的安全设置，用于控制 App 是否可以执行外部函数调用、包含 C 库的链接、运行 MATLAB 函数、访问外部进程等等。

COMSOL Multiphysics 中的安全设置可通过**文件**菜单访问，位于**首选项**窗口的**安全性**页面。在 COMSOL Server 中，如果您以管理员身份登录，这些设置位于 COMSOL Server Web 界面的 “首选项” 页面。如果您不确定使用哪些安全设置，请与系统管理员联系。

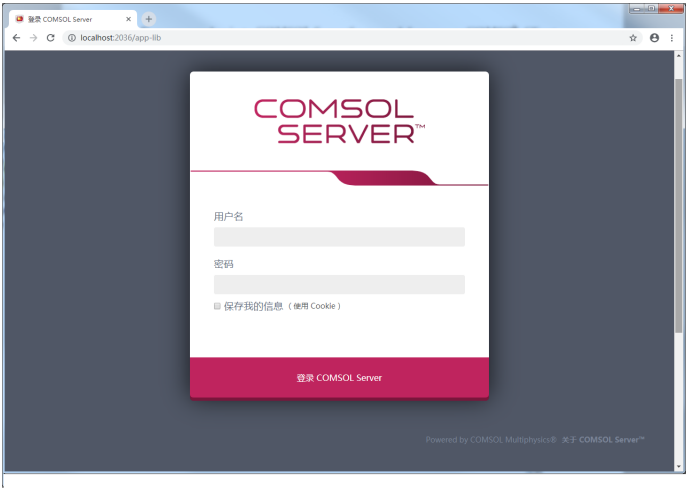
通过 COMSOL Server 运行 App

COMSOL App 在从 Web 浏览器或 Windows® 版 COMSOL Client 连接到 COMSOL Server 后，即可开始运行。Windows® 版 COMSOL Client 支持用户运行需要 CAD LiveLink™ 产品的 App，如在 COMSOL Client 中运行 App 所述。

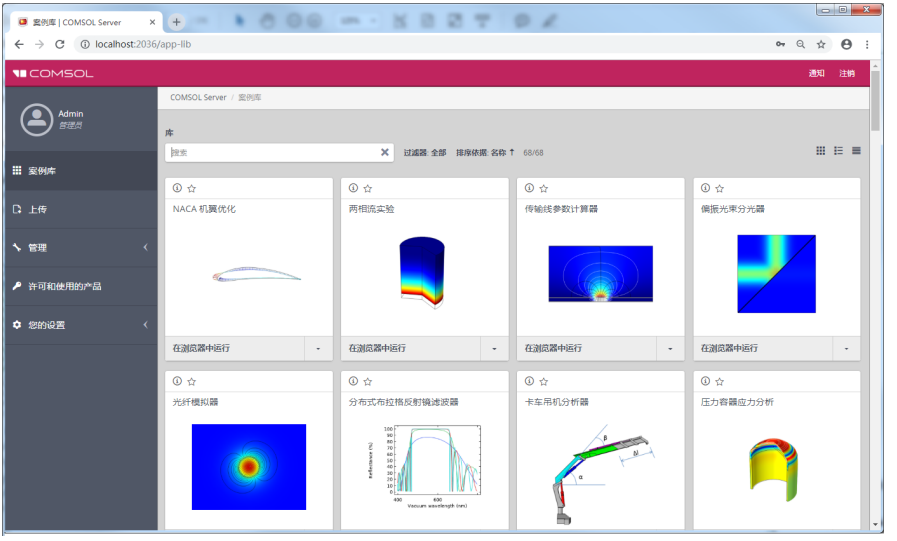
在 Web 浏览器中运行 App 时无需进行任何安装，也不需要任何 Web 浏览器插件。在 Web 浏览器中运行 App 时支持一维、二维和三维交互式图形。在 Web 浏览器中，三维图形渲染基于所有主流 Web 浏览器所采用的 WebGL™ 技术。

在 Web 浏览器中运行 App

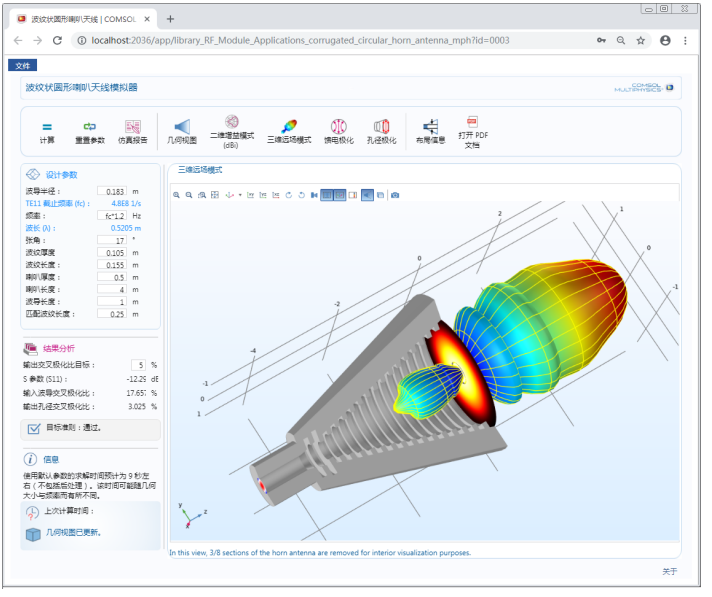
使用 Web 浏览器，您可以直接指向 COMSOL Server Web 界面的计算机名和端口号 - 例如 `http://comsol-server-machine-url.com:2036`，假设在安装 COMSOL Server 时使用了端口号 `2036`。您需要提供用户名和密码进行登录。如果您在本地运行 COMSOL Server，地址栏通常为 `localhost:2036`。



登录后，**案例库**页面会显示可运行的 App 列表。



单击**在浏览器中运行**以运行 App。 App 在浏览器的单独选项卡中运行。



在 Web 浏览器中运行 App 的限制

当您创建要在 Web 浏览器中运行的 App 时，请确保在 “App 开发器”中使用栅格布局模式；请参见第 102 页的 “草图和栅格布局”。这样可以确保用户界面布局能适应浏览器窗口的大小和长宽比。对于低分辨率的显示器，请确保在目标平台上测试用户界面布局，以确认所有表单对象均可见。包含可调图形表单的 App 可能不适合低分辨率显示器。在这种情况下，使用固定宽度和高度的图形，以确保所有表单对象都能适合目标浏览器窗口。根据 Web 浏览器和显卡的不同类型，一个 App 能使用多少图形对象可能会有所限制。您可以切换到图形对象的源来重用同一图形对象，而非使用多个图形对象，以此来应对这种局限。

在 Web 浏览器中运行时，不支持针对 CAD 的 LiveLink™ 产品软件包。

在智能手机和某些平板电脑的 Web 浏览器中运行 COMSOL App 时，并非支持所有功能。典型限制包括播放声音或打开文档的功能。此外，还可能不支持文件上传和下载。

如果 App 支持用户进行选择，例如单击边界以设置边界条件，则在 Web 浏览器中运行与在 COMSOL Multiphysics 或 Windows® 版 COMSOL Client 中运行有所不同。在 Web 浏览器中，当光标悬停在边界上方时，边界不会自动突出显示。而是需要单击一次才能突出显示边界。第二次单击时可将其选中。第三次单击时将突出显示，用于取消选择；第四次单击时可取消选择。这一过程同样适用于域、边和点。

请注意，文件浏览功能会略有不同，具体取决于 Web 浏览器及其版本。如果运行的 App 具有将文件保存到客户端计算机的功能，则可能影响用户体验。例如，可以在许多 Web 浏览器的设置中更改下载文件夹的位置。Web 浏览器还可以支持用户手动指定每个文件的下载位置。请参阅目标 Web 浏览器的文档，了解详细信息。

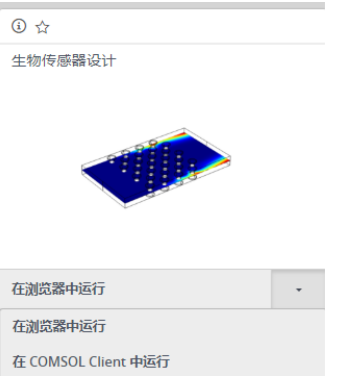
在 COMSOL CLIENT 中运行 APP

除了使用 Web 浏览器来运行 App，Windows® 版 COMSOL Client 还可用于连接到 COMSOL Server，从而在 Windows® 操作系统中实现本机运行 App。使用这种方式，通常能得到更佳的图形性能，并支持一维、二维和三维中更复杂的图形渲染。此外，Windows® 版 COMSOL Client 还支持运行需要 CAD LiveLink™

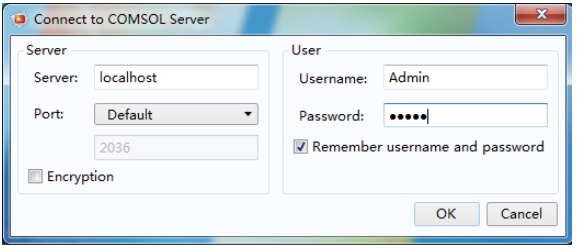
产品的 App，前提是连接的 COMSOL Server 需要具有必要的许可证。您可以通过两种不同的方式在 Windows® 版 COMSOL Client 中打开 App：

- COMSOL Server Web 界面支持您选择是在 Web 浏览器中运行 App，还是使用 Windows® 版 COMSOL Client 运行。

  如果您试图通过这种方式在 COMSOL Client 中运行 App，但尚未安装软件，则系统将提示您下载并安装。

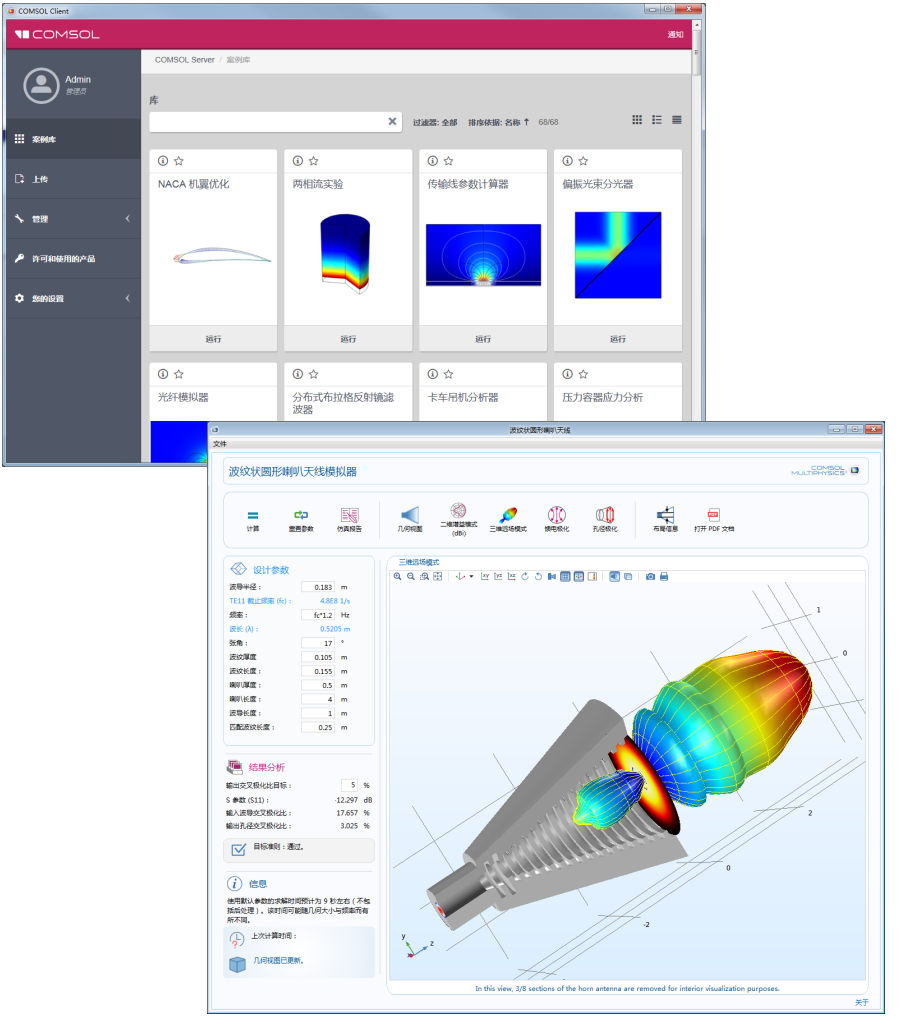


- 如果您已安装 Windows® 版 COMSOL Client，则会显示一个桌面快捷方式。您可以双击桌面图标，系统将提示您使用有效的用户名和密码登录 COMSOL Server，然后您便能使用 COMSOL Client 来运行 App。登录之后， COMSOL Client 会显示 COMSOL Server Web 界面，与您从 Web 浏览器登录后看到的界面相同。



使用 COMSOL Client， App 可以在单独的窗口中作为原生的 Windows® 应用程序运行。例如，在 COMSOL Client 中运行的 App 可具有带选项卡的 Windows® 功能区。在 Web 浏览器中运行时，功能区则显示为一个工具栏。

下图显示了 COMSOL Server Web 界面 （顶部），以及在 Windows® 版 COMSOL Client 中启动的 App （底部）。



在多台计算机或集群上运行 COMSOL SERVER

COMSOL App 可通过以下两种主要方式在多台计算机或集群上运行：

- 通过主实例和辅助实例安装 COMSOL Server。
- 在 “模型开发器” 中为特定集群配置一个研究节点。

主实例和辅助实例

与单个计算机实例 （或安装）相比，在多台计算机上使用主实例和辅助实例来运行 COMSOL Server 可以支持更多并发用户和 App。主要的 COMSOL Server 实例称为主实例，其他实例则称为辅助实例。主服务器用于所有传入的连接，例如，显示 Web 界面，或者在 Web 浏览器或 COMSOL Client 中运行 App。实际的计算会分配给辅助服务器计算机。此类安装的主要优势在于：无需为特定集群定制 App。负载平衡由主服务器自动管理，它会在辅助服务器之间分配工作负载。 COMSOL Server 安装可以包含多个主服务器和辅助服务器安装，而没有额外的许可证要求。您可以使用 COMSOL Server Web 界面执行管理任务，而无需检查运行 App 的用户的许可证密钥。仅当运行 App 时才检查许可证密钥。

为集群扫描或集群计算配置研究节点

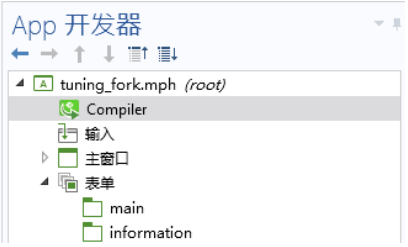
如果您希望将集群用于需要大量参数化扫描或高性能计算能力的 App，可以使用 “集群扫描” 和 “集群计算” 选项来配置该 App 的 “模型开发器” 研究节点。请注意，构建这类 App 时需要 “网络浮动许可证”。有关为集群配置研究节点的更多信息，请参阅 《COMSOL Multiphysics 简介》和 COMSOL Multiphysics Reference Manual。要运行这种启用了集群的 App，可以使用 COMSOL Server 或 “网络浮动许可证” 类型的 COMSOL Multiphysics。集群系统配置可以从 COMSOL Server Web 界面访问。

有关 COMSOL Server 的更多信息，请参阅 *《COMSOL Server 手册》*，该文档随 COMSOL Server 安装提供，您也可以访问以下链接获取：http://cn.comsol.com/documentation/COMSOL_ServerManual.pdf。

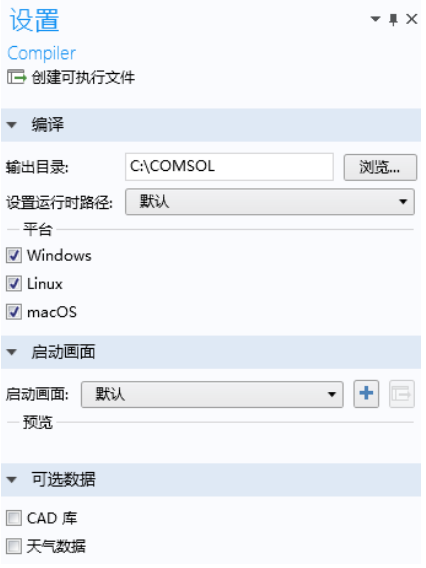
编译和运行已编译的 App

如果您拥有 COMSOL Compiler™ 许可证，则主功能区栏中会显示 **Compiler** 按钮，如下所示。

单击此按钮后， App 树中会添加一个 **Compiler** 节点，如下图所示。



相应的**设置**窗口如下所示。



编译 App

要编译 App，您需要在此窗口中进行一些选择。指定**输出目录**，用于保存编译后的可执行文件。

大多数情况下，**设置运行时路径**选项都可以保留**默认**设置。此选项用于指定在计算机上首次启动编译的 App 时， COMSOL Multiphysics 运行时库的存储位置。在 Windows® 操作系统中，假设用户名为 `paul`，则该用户的默认存储位置为 `C:\Users\paul\AppData\Local\COMSOL`。运行时库包含将 App 作为独立程序运行所需的所有 COMSOL Multiphysics 软件组件。其他选项还包括**询问** —— 要求用户在第一次运行 App 时指定位置，以及**指定**，用于在此窗口中为用户提供显式路径。

平台设置可以确定在编译时应生成哪些目标平台可执行文件。**启动画面**设置供您指定要在启动时显示的 BMP 图像文件。

在**可选数据**栏中，您可以选中所需的复选框以指定在编译的 App 中应包含哪些附加的软件组件。通常，系统会根据所用的功能自动选择这些组件，并能使用这些选项将可执行文件大小最小化。然而，如果您有动态包含此类功能的方法代码，则可能需要手动选择这些组件。只要 App 包含 “CAD 导入模块”、“设计模块” 或 CAD 软件包的任一 LiveLink™ 产品中的功能，都需要使用 **CAD 库**。当 App 使用传热接口时，需要使用**天气数据**，这是因为您在传热接口中使用了**组件** > **定义**节点下**环境热属性**中的**气象**选项。

在 Windows® 操作系统中，编译后的可执行文件将保存在输出目录中，如下图所示。

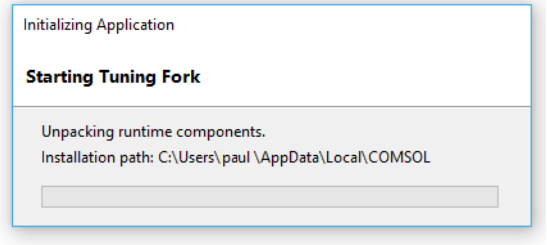
接下来，您可以右键单击这个 EXE 文件来创建快捷方式，然后将快捷方式放在 Windows® 桌面上。可执行 App 的桌面图标可以从 App 树中**主窗口**节点的**图标**设置中获取。

除此之外，您还可以从操作系统命令行编译 App。有关更多信息，请参阅 COMSOL Multiphysics Reference Manual。

运行编译的 App

举例来说，当您通过在 Windows® 操作系统中双击 `.exe` 文件来运行已编译的 App 时，会显示启动画面屏幕并启动该 App。如果 App 的**启动画面**选项已设为**默认**，则会显示带灰色的内置启动画面屏幕。建议您将其替换为自己的启动画面屏幕。

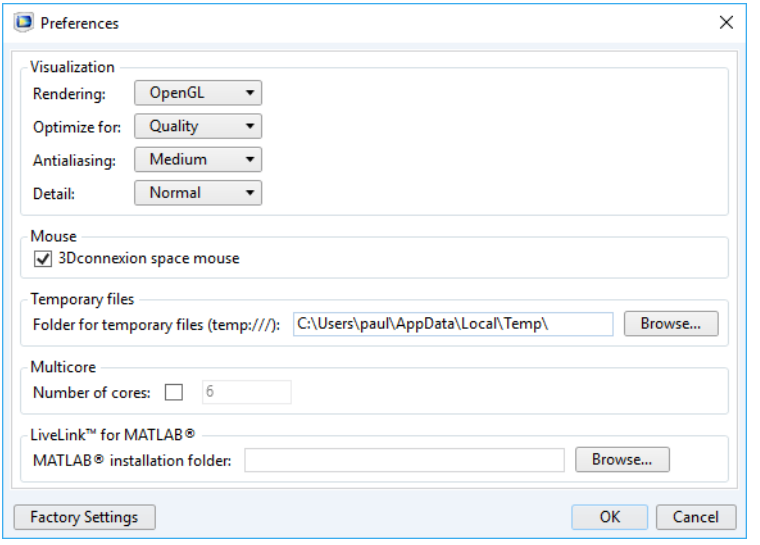
当您首次在特定的计算机上运行 App 时，还会显示单击认可协议和 **Initializing Application** 进度窗口。初始化进度窗口如下所示。



单击认可协议和 **Initializing Application** 进度窗口只显示一次，您下次启动同一 App 时将不再显示。如果您在同一台计算机上运行使用相同的 COMSOL Compiler 版本 （与运行时库的版本相同）生成的其他 App，也不会显示单击认可协议和初始化进度窗口。

编译的 App 中始终显示一个选项，用于设置 COMSOL **关于**对话框的显示情况。App 作者可以在**主窗口**的**设置**中控制此信息的显示；请参见第 124 页的 “关于对话框”。下图显示**关于**对话框。

编译 App 的用户可以在**关于**对话框中通过单击**首选项**按钮来访问该对话框。编译 App 的**首选项**对话框如下所示。



用户可以在此更改**可视化**、**鼠标**、**临时文件**、**多核**以及 **LiveLink™ for MATLAB®** 设置。这些设置是 COMSOL Multiphysics 中提供的**首选项**的子集，有关更多信息，请参阅 COMSOL Multiphysics Reference Manual。

如果编译的 App 检测到系统不支持 OpenGL® 图形硬件加速，则该 App 将自动切换为软件渲染并退出。该 App 在下次启动时，将使用软件渲染方式。

发布 COMSOL App

《COMSOL 软件许可协议》 (SLA) 允许您发布 COMSOL App 供他人使用 （包括商业用途），但有一定的限制，如 SLA 中所述：cn.comsol.com/sla。此权限使您能够将 App 与他人共享，并通过三种不同的机制向他们收取使用费用。

首先，您可以将 App 提供给他人，通过安装 COMSOL Multiphysics 来运行。如果通过 COMSOL Multiphysics 使用 App，则用户需要是购买 COMSOL Multiphysics 许可证的组织的成员。

其次，您可以将 App 提供给他人，通过安装 COMSOL Server 来运行。此方法具有更大的灵活性，支持您在安装 COMSOL Server 时进行设置，使世界各地的用户都可以访问您的 App。您只需向他们提供您的 COMSOL Server 安装的地址、用户名和密码。或者，用户也可以购买他们自己的 COMSOL Server 许可证。如果您使用 COMSOL Server 托管和运行 App， SLA 还允许您将 COMSOL

Server 许可证 (CSL) 分许可给组织外的人员，以托管和运行您发布给他人的 App，但受到一定的限制。

第三，您可以使用 COMSOL Compiler 将 App 编译成独立的程序，其中包含运行该程序所需的所有功能。这种方法为您带来最大的灵活性， App 的最终用户不需要 COMSOL Multiphysics 或 COMSOL Server 许可证即可运行 App。然后，该用户以及您允许该用户向其发布已编译 App 的任何人（您组织内外、全世界的任何人）都能够运行这个 App。

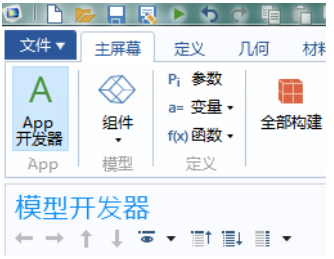
COMSOL Application License （也可通过 cn.comsol.com/sla 访问）还允许您修改 “案例库”中的 App，并发布这些修改后的 App 供他人使用 （包括商业用途），但有一定的限制，如 Application License 中所述。这样，您便可以使用 “案例库”中的一个 App 开始操作，通过添加或移除您自己的特征来开发自己的 App，等等。

如果您希望将 Application License 应用于您自己创建的 App，可以参见 Application License 中包含的相关操作说明。 Application License 中还陈述了您可以如何使用您对 “案例库”中的 App 的修改所选择的条款，而这些 App 的原始部分在 Application License 的许可下仍然可用。

从 COMSOL MULTIPHYSICS 模型开始

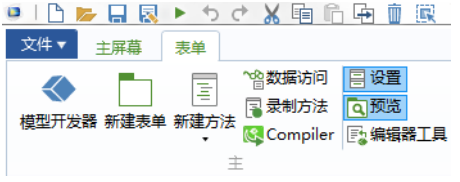
如果您尚未将模型加载到 COMSOL Desktop 环境，可以选择**文件 > 打开**从文件系统选择 MPH 文件，也可以从 “案例库” 选择一个文件。请注意，“案例库” 的 **App** 文件夹中的文件都是现成的 App。“案例库” 中的其他所有文件都包含模型和文档，但不包含 App 用户界面。

加载模型后，单击功能区 “主屏幕” 选项卡中的 **App 开发器**按钮。软件会切换到 “App 开发器” 桌面环境。



新建表单

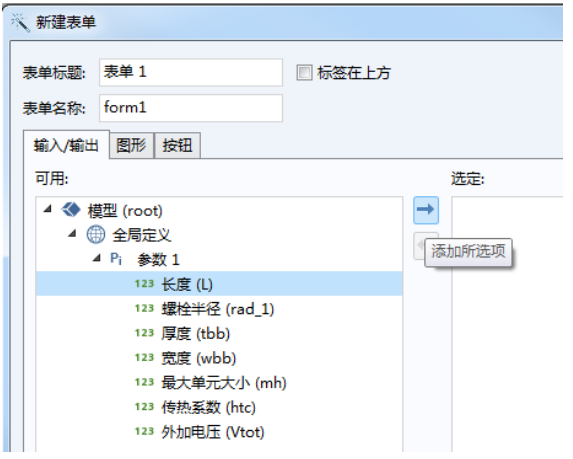
单击**主屏幕**选项卡中的**新建表单**按钮，开始构建用户界面布局。此时将启动**新建表单**向导。



新建表单向导可帮助您将最常用的用户界面组件 （即所谓的表单对象）添加到 App 的第一个表单中。其中包含三个选项卡：

- **输入 / 输出**

- 图形
- 按钮



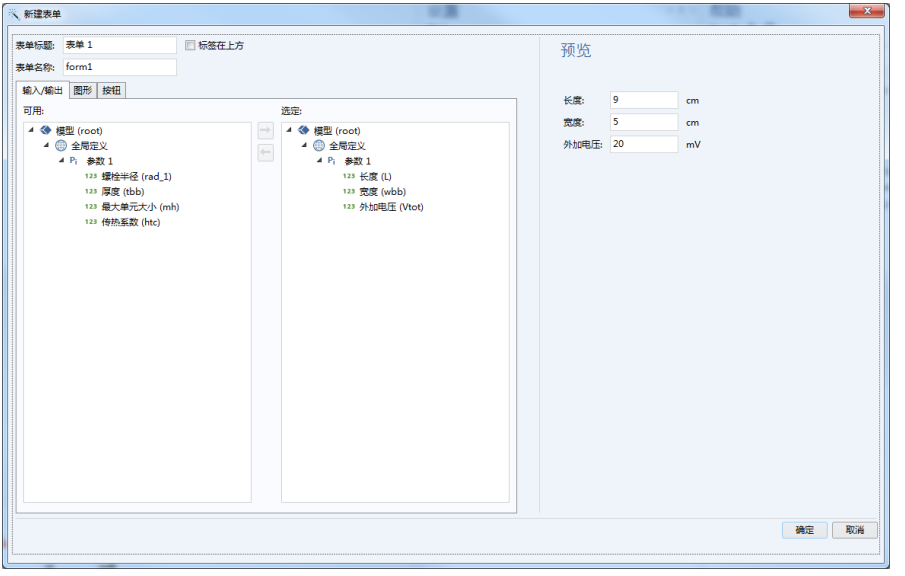
双击某个节点或单击**添加所选项** 按钮，可将该节点从**可用**区域移到**选定**区域。选定的节点将变成 App 中的表单对象，并在右侧的**预览**区域显示该表单的预览效果。表单对象的大小及其他设置可以在退出该向导后进行编辑。您可以在向导窗口顶部更改表单的名称和标题。有关详细信息，请参见第 47 页的“单独的表单设置窗口”。

您也可以通过单击**完成**在此阶段选择退出**新建表单**窗口，随后再手动添加表单对象。

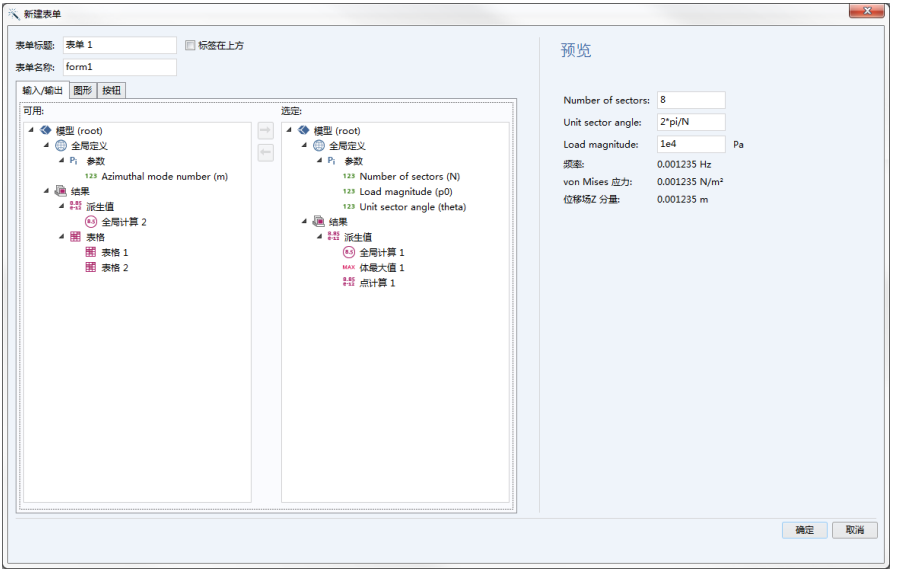
输入 / 输出选项卡

输入 / 输出选项卡显示可以作为输入框、数据显示对象、复选框或组合框的模型树节点。在该向导中添加的输入框将附带文本标签和单位 （如果适用）。您可以使用**数据访问** （请参见第 96 页）将模型的其他部分设为输入和输出。例如，复选框和组合框对象只能通过这种方式添加。例如，您可以通过**数据访问**来启用**网格**节点下**单元大小**的**预定义**组合框，使其显示在向导中。

在下图中，选定了三个参数作为输入框，包括“长度”、“宽度”和“外加电压”。



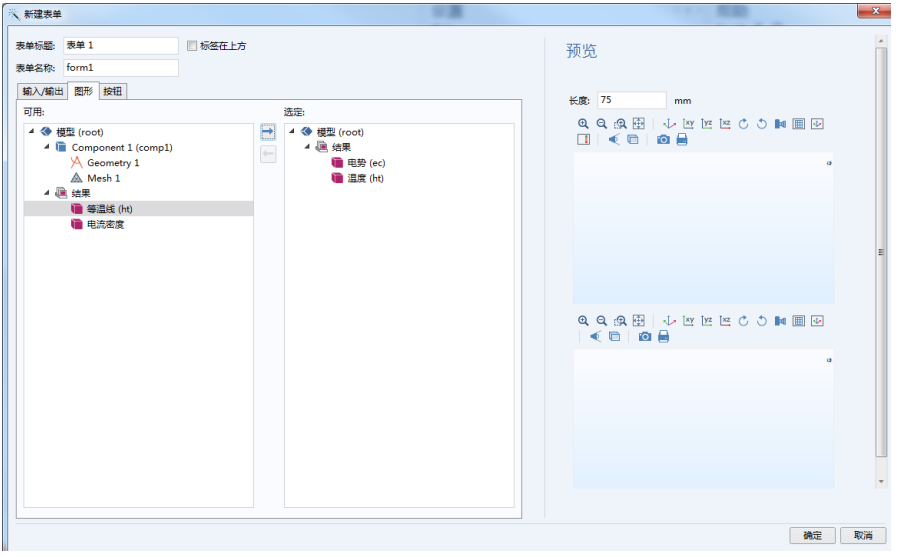
下图对应于另一个模型，其中选定了三个**派生值**节点作为数据显示对象。



退出向导后，您可以编辑输入框和数据显示对象的大小和字体颜色，以及其他设置。

图形选项卡

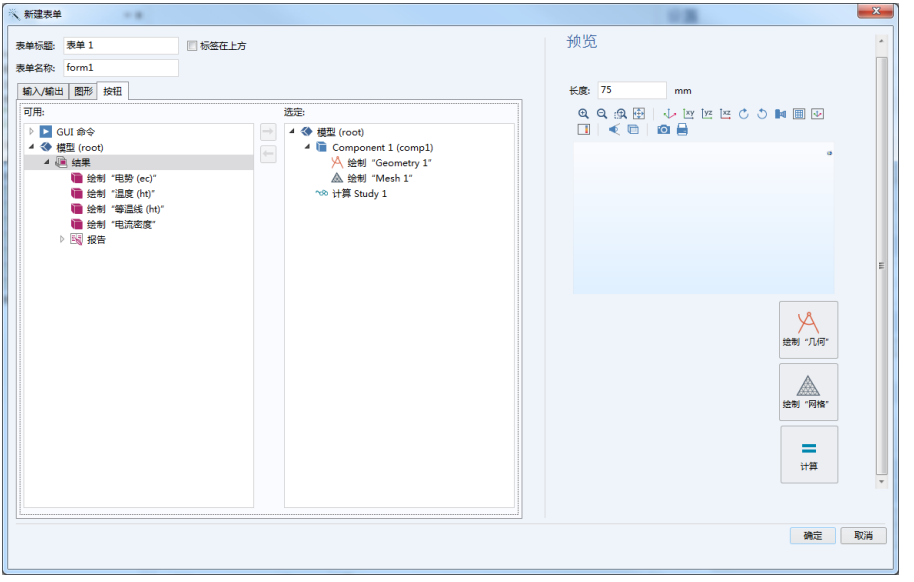
图形选项卡显示可以作为图形对象的模型树节点：**几何**、**选择**、**网格**和**结果**。在下图中，选定了两个这样的节点。



按钮选项卡

按钮选项卡显示可通过单击 App 用户界面中的按钮来运行的模型和 App 树节点。这种树节点示例包括**绘制几何**、**绘制网格**、**计算研究**，以及**结果**节点下每个不同的绘图组。此外，您还可以为 **GUI 命令**、**表单**和**方法**添加按钮。

在下图中，添加了三个按钮：**绘制几何**、**绘制网格**和**计算**。



通过使用 “表单编辑器”，您可以添加按钮来运行自己的定制命令序列或方法。

编辑向导

单击**确定**以退出向导。此时软件会自动切换到 “表单编辑器”。

保存 APP

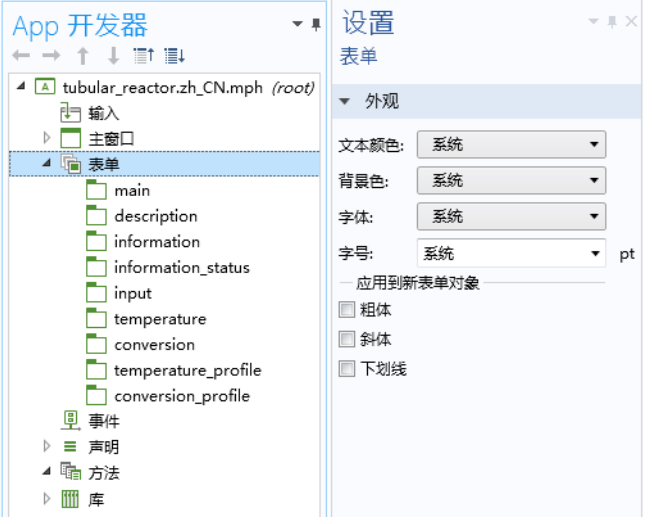
要保存 App，可以从**文件**菜单中选择**文件 > 另存为**。浏览至您拥有写权限的文件夹，将文件保存为 MPH 格式。该 MPH 文件包含有关 App 的所有信息，包括使用 “模型开发器” 创建的嵌入模型相关信息。

表单编辑器

“表单编辑器”用于设计用户界面布局，您可以在其中使用表单对象 （如输入框、图形、按钮等）来创建表单。

表单设置窗口

当您单击 App 树中的**表单**节点时，会显示表单的**设置**窗口。您可以通过**文本颜色**、**背景色**、**字体**、**字号**、**粗体**、**斜体**和**下划线**等设置来更改表单的整体外观。

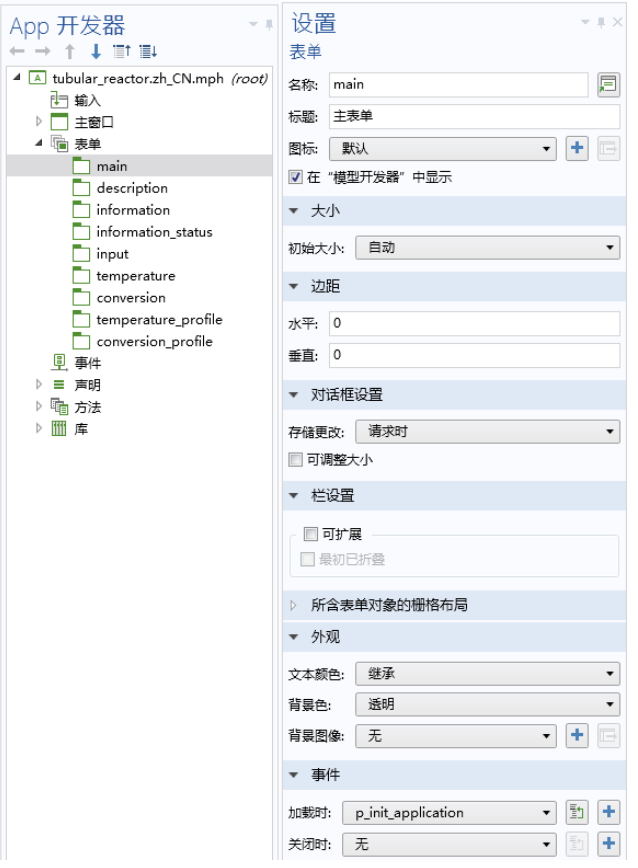


默认设置为所有新表单和新表单对象都继承适用的现有设置。

在上图和下面的一些图中，**设置**窗口停靠在 **App 开发器**窗口的右侧。默认情况下，**设置**窗口停靠在 “App 开发器”桌面环境的最右侧。

单独的表单设置窗口

下图显示表单的**设置**窗口。

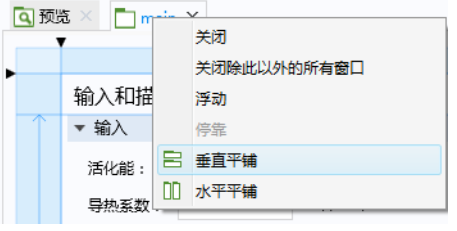


每个表单都有自己的**设置**窗口，并包含以下设置：

- **名称**，用于引用其他表单对象和方法中的表单。
- 表单**标题**，用于包含多个表单的 App 中。
- 当表单用作对话框或当**主窗口**设为由表单决定大小时的**初始大小**。
- 相对于左上角 （**水平**和**垂直**）的**边距**。
- 何时存储对话框更改的选项 （**存储更改**），另请参见第 66 页的 “将表单显示为对话框”。
- 对话框左上角显示的**图标**。
- 当表单用作对话框时，用于设置表单是否**可调整大小**的选项。

- 用于设置查看的栏是否**可扩展**以及是否**最初已折叠** （**栏设置**）的选项。
- 表单中包含的所有行和列的格式化表格 （**所含表单对象的栅格布局**）。
- **外观**，包含**文本颜色**、**背景色**和**背景图像**设置。
- 在加载或关闭表单时触发的**事件**。（**加载时**和**关闭时**。）

双击一个表单节点，在 “表单编辑器”中打开相应的窗口。或者，您也可以右键单击表单节点，然后选择**编辑**。右键单击表单窗口选项卡以查看其上下文菜单，其中包含以下选项：关闭、浮动及平铺表单窗口等。



草图和栅格布局模式

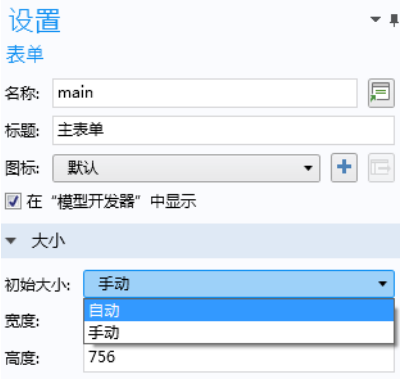
“App 开发器”默认设为草图布局模式，可供您使用固定的对象位置和大小。除非另行指定，否则 “表单编辑器”一节中的操作说明假定 “表单编辑器”为草图布局模式。有关栅格布局模式的信息，请参见第 102 页的 “草图和栅格布局”。

表单初始大小

表单的初始大小有两个选项：

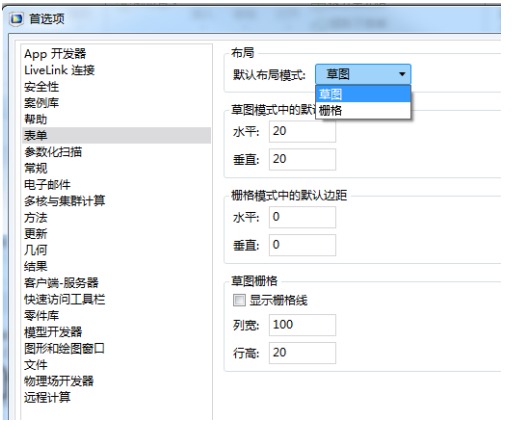
- **手动**，供您输入宽度和高度的像素大小。
- **自动**，可基于表单所含的表单对象来确定大小。如果您使用栅格布局模式，并且列或行设为**增长**，则大小不由表单对象进行定义。在这种情况下，使用 “表单编辑器”的栅格大小作为基点来估计大小。（通常会略大一些。）

您可以拖动栅格的右侧或底部边界来更改栅格大小。有关栅格布局模式的更多信息，请参见第 105 页的 “栅格布局”。



表单编辑器首选项

要访问 “表单编辑器” 的**首选项**，可以从**文件**菜单中选择**首选项**，然后选择**表单**页面。



表单栏包含的设置可用于更改**布局模式**、**边距**和**草图栅格**的默认设置。

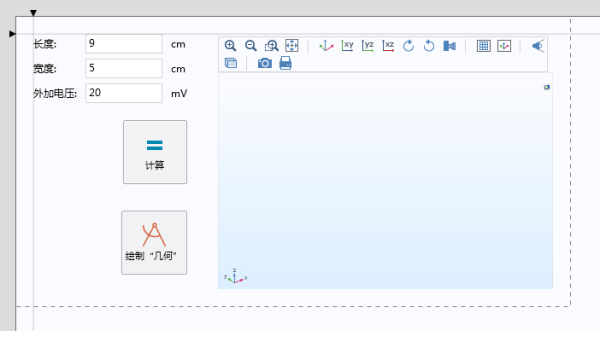
定位表单对象

您可以通过以下方式之一轻松更改表单对象 （如输入框、图形对象和按钮）的位置：

- 单击对象将其选中。选定的对象会以蓝色框架突出显示。
- 要选择多个对象，可以在按住 Ctrl 的同时单击。您还可以在表单窗口中单击并拖动以创建一个选择框，从而选择框内的所有对象。
- 通过按住并拖动操作来移到新位置。蓝色引导线可帮助您相对于其他对象进行定位。

- 在草图布局模式下，您还可以使用键盘上的箭头键来移动对象。使用 Ctrl+箭头键可以对位置进行微调。

在下图中，**绘制**按钮将从原始位置进行移动。蓝色引导线显示了该按钮相对于单位对象和**计算**按钮的对齐位置。



调整表单对象大小

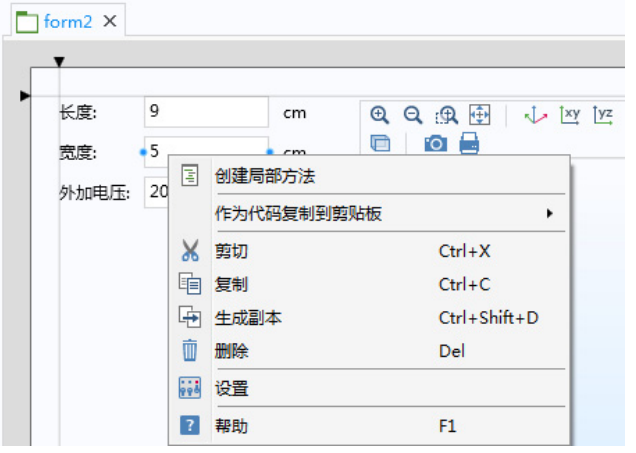
要调整对象大小：

- 单击对象将其选中。
- 按住并拖动突出显示的蓝色框架上的一个控制点 （显示为蓝色圆点）。如果没有控制点，说明这种表单对象的大小无法调整。

复制、粘贴、生成副本和删除对象

要删除对象，可以单击将其选中，然后按键盘上的 Delete 键。您也可以单击快速访问工具栏中的**删除**按钮。

您可以按 Ctrl+C 和 Ctrl+V 来复制粘贴对象。或者，也可以右键单击对象以获取以下菜单选项：**复制**、**生成副本**、**删除**等。

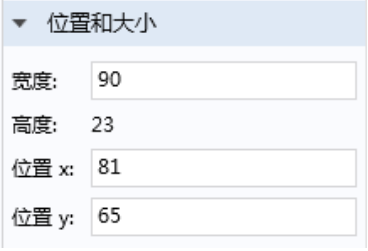


要粘贴一个已复制的对象，可右键单击表单中的空区域并再次右键单击。由于已复制对象，此时会显示**粘贴**菜单选项。在下图中，由于已复制**输入框**，因此会显示**粘贴 “输入框”** 选项。

通过像素数调整位置和大小

在草图布局模式下，您可以通过在对象**设置**窗口的**位置和大小**栏中键入像素数来调整其位置和大小：

- 单击对象将其选中。确保显示**设置**窗口。如果未显示，可以双击该对象，或在**表单**选项卡中单击**设置**按钮。
- 在**位置和大小**栏中编辑数字。

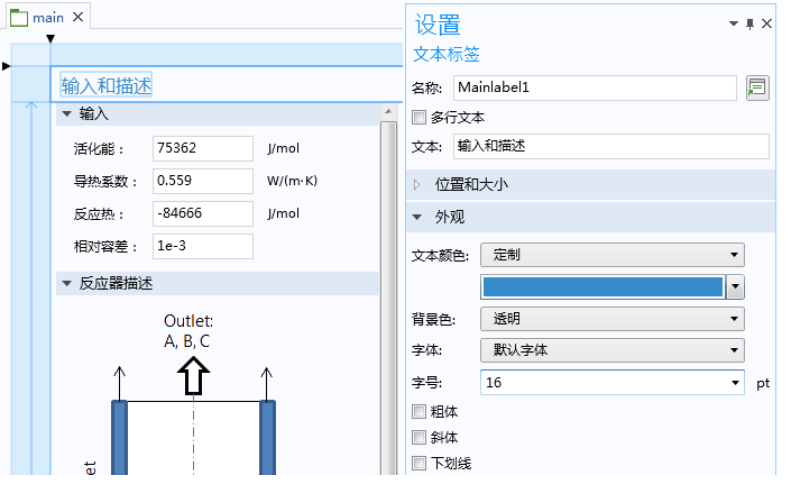


根据表单对象类型的不同，“位置和大小”栏会提供不同的选项。对于栅格布局模式，则提供了其他设置，对象的位置可通过行数和列数来进行调整。有关详细信息，请参见第 102 页的 “草图和栅格布局”。

更改文本显示外观

对于显示文本的表单对象，**设置**窗口中的**外观**栏可供您更改各种属性，例如，显示的文本、字体、字体颜色及字号。对于某些表单对象 （如按钮），对象的大小将适应文本字符串的长度。

下图显示一个文本标签对象的**设置**窗口，其中的字号和颜色已更改。



通过使用栅格布局模式（请参见第 102 页的“草图和栅格布局”），您可以进一步控制表单对象的大小，例如为按钮设置任意大小。

选择多个表单对象

例如，如果您使用 Ctrl+ 单击来选择多个表单对象，则**设置**窗口将包含一组可在所选对象之间共享的属性。共享属性总是存在于**外观**栏、**位置和大小**栏或**事件**栏。

表单对象的名称

表单对象都具有**名称**，这是一个不含空格的文本字符串。其中可以包含字母、数字和下划线。此外，不允许使用保留的名称 `root` 和 `parent`。**名称**字符串在其他表单对象和方法中用来引用对象。当您将光标悬停在**设置**窗口的**名称**框上方时，对象的路径会显示为工具提示。

插入表单对象

除了使用**新建表单**向导创建的表单对象，您还可以插入各种表单对象。在**表单**功能区选项卡中，选择**插入对象**菜单以查看所有可用的对象列表。

本表单编辑器章节的其余部分仅描述通过**新建表单**向导添加的表单对象类型。使用向导添加的表单对象可以包括：

- **按钮**
- **图形**
- **输入框**
- **文本标签** （与 “输入框” 关联）
- **单位** （与 “输入框” 关联）
- **数据显示**

不过，在使用**数据访问** （参见第 96 页）时，还可以添加其他表单对象，包括：

- **复选框**
- **组合框**

有关复选框、组合框及其他表单对象的更多信息，请参见第 207 页的 “附录 A - 表单对象”。

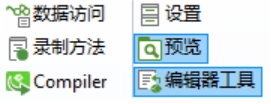
与表单和表单对象关联的事件和操作

您可以将对象与事件触发的操作关联起来，这些对象包括按钮、菜单项、功能区按钮、表单、表单对象等。操作可以是一个命令序列，包括全局方法或局部方法。在定义这些局部方法的表单或对象外部无法访问或查看这些方法。可与对象关联的事件取决于对象的类型，包括：单击按钮、键盘快捷键、加载表单（**加载时**）、关闭表单 （**关闭时**）以及更改变量的值 （**数据更改时**）。

在按住 Ctrl+Alt 的同时单击表单对象，可以在**方法**编辑器中打开任何关联的方法。如果没有与表单对象关联的方法，则会创建一个新的局部方法与该表单对象关联，并在**方法**编辑器中打开。如果表单对象具有关联的命令序列，则该序列会转换为代码并插入局部方法中。

表单编辑器中的编辑器工具

编辑器工具窗口是对**新建表单**向导和**插入对象**菜单的重要补充，可用于快速创建复合表单对象。要显示**编辑器工具**窗口，可以在**表单**选项卡的**主**组中单击相应的按钮。



您可以右键单击编辑器树中的节点，添加一组与**新建表单**向导中相同的表单对象。

在选定某个节点时，编辑器树下方的工具栏中会显示可用于插入对象的选项。您也可以右键单击节点来查看这些选项的列表。

根据节点的不同，可能出现以下选项：

- **输入**

  通过以下方式插入**输入框**、**复选框**、**组合框**或**文件导入**对象：

  - 使用选定的节点作为**源**来插入**输入框**。该对象带有**文本标签**和**单位**对象（如果适用）。
  - 使用选定的节点作为**源**来插入**复选框**。
  - 使用选定的节点作为**源**来插入**组合框**。此时会自动创建一个选择列表，对应于节点中的列表。此选项仅当与**数据访问**（请参见第 96 页）结合使用时才可用，从而使对应的节点显示在编辑器树中。
  - 使用选定的节点作为**文件目标**来插入**文件导入**对象。

- **输出**
  - 插入**数据显示**对象，并在适用的情况下附带**文本标签**。
  - 当选定的节点为**表格**时插入**结果表**对象。
- **按钮**
  - 插入**按钮**对象，其中包含可运行选定节点的的命令序列。
- **图形**
  - 使用选定的节点作为**初始图形内容的源**来插入**图形**对象。
- **编辑节点**
  - 软件将自动打开对应模型树节点的**设置**窗口。

当您使用 “方法编辑器”时，“编辑器工具”窗口也是一个重要的工具。在“方法编辑器”中，它用于生成与编辑器树的节点关联的代码。有关更多信息，请参见第 166 页的 “方法编辑器中的编辑器工具”。

按钮

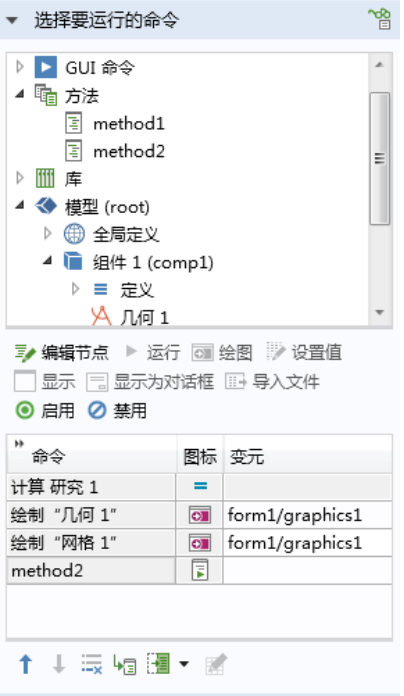
单击**按钮**的操作相当于一个事件，可以触发由对应的命令序列定义的动作。按钮**设置**窗口的主栏可供您：

- 编辑按钮的表单对象**名称**。
- 编辑按钮上显示的**文本**。
- 使用**图标**来代替默认的按钮渲染方式。
- 将按钮**样式**设为**大**、**小**或**扁平**。
- 添加**工具提示**，其中包含当光标悬停在按钮上方时显示的文本。

- 通过单击输入框并在键盘上使用修改键 Shift、Ctrl 和 Alt 与其他键的组合，可以添加**键盘快捷键**。其中， Alt 必须与至少一个其他修改键结合使用。

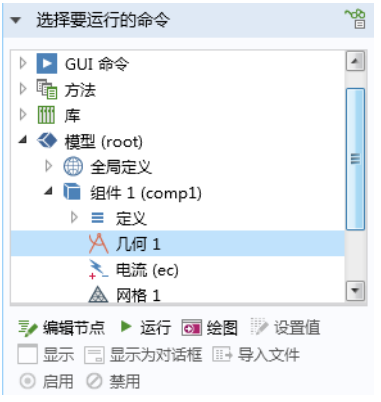
选择要运行的命令

选择要运行的命令栏可供您控制与单击按钮事件关联的操作。在下图显示的按钮**设置**窗口中，该按钮可以触发包含四个命令的序列。



在菜单、功能区或工具栏项的**设置**窗口中，也包含**选择要运行的命令**栏，都具有本节描述的功能。有关使用菜单、功能区和工具栏项的更多信息，请参见第 75 页的“图形工具栏”、第 122 页的“主窗口”、第 277 页的“表格”以及第 286 页的“工具栏”。

要定义命令序列，可以在**选择要运行的命令**栏中选择编辑器树中的节点。然后单击编辑器树下方某个突出显示的按钮，或右键单击该节点并选择相应的命令。在下图中，**几何**节点处于选中状态，可用的命令**运行**和**绘制**被突出显示。单击**运行**可将几何构建命令添加到命令序列中。单击**绘制**可以添加一个命令来依次构建并绘制几何。单击**编辑节点**选项后，软件会自动跳转到模型树或 App 树中对应的节点。



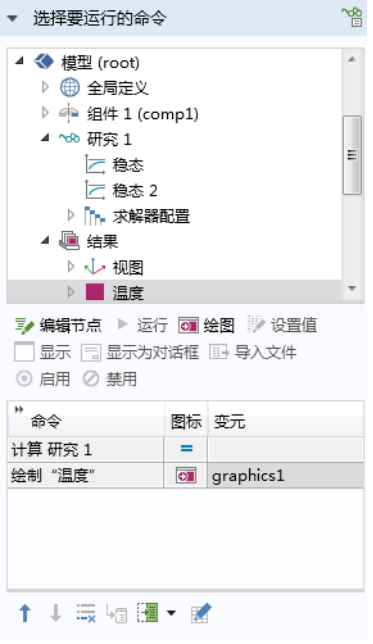
您无需在**绘制几何**命令之前执行**构建几何**命令（可通过单击**运行**触发该命令）。**绘制几何**命令将依次构建并绘制几何。**绘制网格**命令也会以同样的方式先构建网格，然后再绘制网格。

突出显示用于选择的命令图标是指适用于特定树节点的命令。根据节点的不同，可能显示以下命令图标：

- **运行**
- **绘制**
- **设置值**
- **显示**
- **显示为对话框**
- **导入文件**
- **启用**
- **禁用**

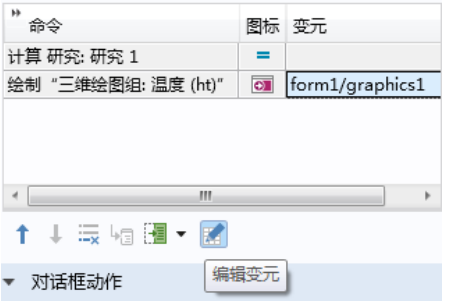
某些命令（如各种绘图命令）需要变元。例如，绘图命令的变元可定义对于各种不同的图形对象，具体要绘制哪个。

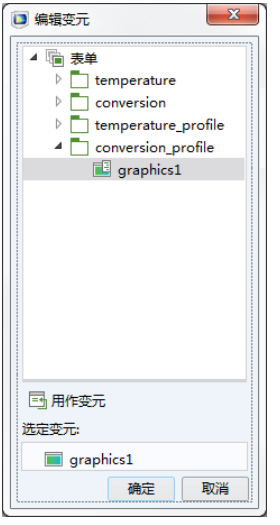
以下示例显示由**新建表单**向导创建的**计算**按钮的**设置**窗口和命令序列。此按钮有一个包含两个命令的序列：**计算研究 1** 和**绘制温度**。



其中**绘制温度**命令有一个变元 graphics1。

要添加或编辑输入变元，可以单击命令序列下方的**编辑变元**按钮，如下图所示。





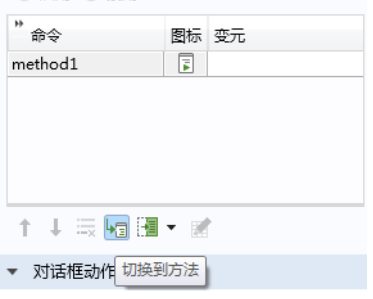
要在特定表单中引用图形对象，需要使用以下语法：`/form1/graphics2`、`/form3/graphics1` 等。如果未指定特定的表单，例如 `graphics1`，则使用按钮所在的表单。

要控制命令序列的顺序和内容，可以使用命令序列表下方的**上移**、**下移**和**删除**按钮。

将命令序列转换成方法

通过单击**转换成方法**，命令序列可以自动转换为新的方法，并可在 “方法编辑器” 中进一步编辑。

通过单击**切换到方法**可以打开这一新方法。



此外，您也可以分别单击**转换成表单方法**或**转换成局部方法**来创建表单或表单对象的局部方法。

方法可以调用对应于命令序列中各个命令的内置方法，如下图所示。

```
method1 ×
1   model.study("std1").run();
2   useGraphics(model.result("pg2"), "form1/graphics1");
3   useGraphics(model.result("pg1"), "form1/graphics2");
4
```

在本例中，第一行：

```
model.study(“std1”).run()
```

运行对应于第一个研究 `std1` （第一个研究节点称为**研究 1**，除非用户手动更改）的模型树节点。第二行和第三行：

```
useGraphics(model.result(“pg2”), “form1/graphics1”);
useGraphics(model.result(“pg1”), “form1/graphics2”);
```

分别使用内置方法 `useGraphics` 来显示对应于绘图组 `pg1` 和 `pg2` 的绘图。在本例中，这些绘图分别显示在两个不同的图形对象中，即 `graphics1` 和 `graphics2`。

有关方法的更多信息，请参见第 161 页的 “方法编辑器”。

设置参数和变量的值

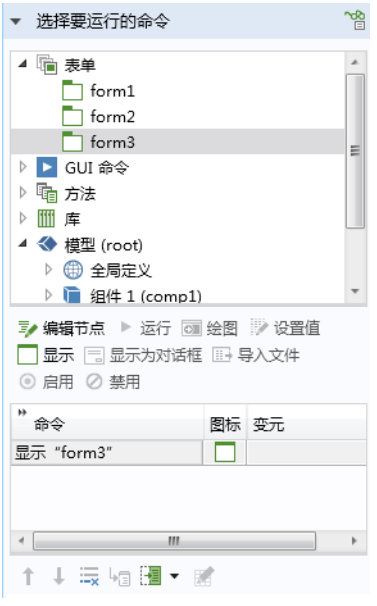
设置值命令可供您设置**参数**、**变量**和**声明**节点中显示的参数和变量的值。此外，**设置值**还可用于设置**数据访问**功能支持的属性的值 （请参见第 96 页）。下图显示用于初始化一组参数和字符串变量的命令序列。

如需了解如何通过方法执行相同的操作序列，可以选择命令表下方的**转换成方法**。

更改表单的可见性

表单上的按钮也可用于显示新表单。可通过两种方式实现。第一种是使用**显示**命令，可将原始表单替换为新表单。第二种是使用**显示为对话框**命令。在这种情况下，新表单将以对话框的形式弹出，位于当前表单上方，并通常需要用户输入。

您可以在**选择要运行的命令**栏中选择**显示**命令。下图显示带有**显示 “form3”**命令的按钮的命令序列。



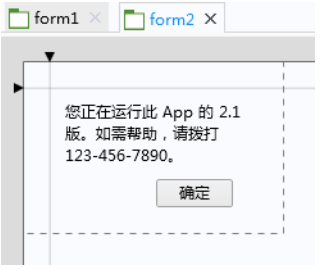
此命令将保留表单与按钮的关联性，并将指定的表单设为对用户可见。

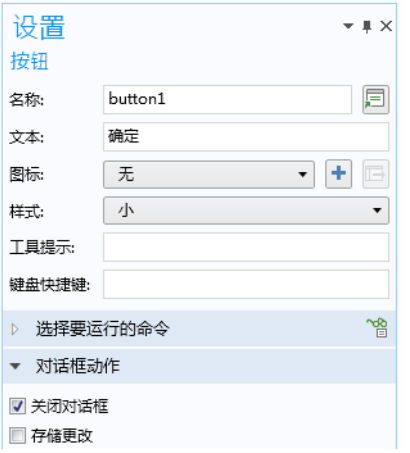
将表单显示为对话框

要使用**显示为对话框**命令，可以在**选择要运行的命令**栏中选择要显示的表单。下图显示带有**将 “form2”显示为对话框**命令的按钮的设置示例。

通过这些设置，在 App 中单击按钮时将启动对应于 **form2** 的以下对话框：

本例中的 **form2** 窗口包含一个文本标签对象和一个**确定**按钮，如下图所示。





在**设置**窗口中，**对话框动作**栏包含两个复选框：

- **关闭对话框**
- **存储更改**

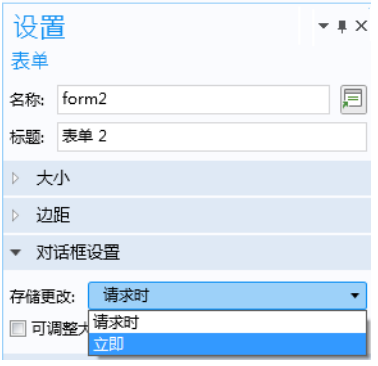
在上例中，**关闭对话框**复选框处于选中状态。这样可以确保在单击**确定**按钮时关闭 **form2** 窗口。由于 **form2** 不包含任何用户输入，因此无需选中**存储更改**复选框。

常用的对话框按钮及关联的对话框动作如下：

按钮	对话框动作
确定	关闭对话框并存储更改
取消	关闭对话框
应用	存储更改

对话框在关闭之前会阻止任何其他用户与 App 进行交互。

要控制何时存储对话框中输入的数据，可以在表单**设置**窗口的**对话框设置**栏的列表中，选择在发生更改时，是在**请求时**存储数据还是**立即**存储，如下图所示。



如果选择**存储更改**选项**请求时**，则在对话框中单击 “确定” 按钮 （或类似按钮）之前，用户在该对话框中更改的变量不会更新。这需要在 “确定” 按钮的**设置**窗口中选中**存储更改**复选框。如果选择**立即**选项，则用户在对话框中更改的变量会立即更新，即使在对话框仍然打开时也是如此。

外观

在按钮的**设置**窗口中，**外观**栏包含字体设置以及用于控制按钮对象状态的设置。

更改表单对象的启用和可见状态

按钮对象是**可见**还是**已启用**，可以通过**状态**子栏中的复选框来控制。大多数表单对象的**外观**栏都具有相似的设置，但部分表单对象包含其他选项；例如，输入框对象。

已清除**可见**复选框的按钮或其他表单对象不会显示在正在运行的 App 用户界面中。已清除**启用**复选框的表单对象将被禁用或 “灰显”，但仍然可见。表单对象的状态还可通过内置方法进行控制。例如，假设一个布尔变量 `enabled_or_disabled` 用于确定一个**名称**为 `button3` 的按钮的 “已启用” / “已禁用”状态。在这种情况下，您可以使用以下方式控制按钮的状态：

```
setFormObjectEnabled(“button3”, enabled_or_disabled);
```

同样，调用

```
setFormObjectVisible(“button3”, visible_or_not);
```

可以使布尔变量 `visible_or_not` 控制是否向用户显示按钮。

有关更多信息，请参见第 319 页的 “GUI 相关方法”和 Application Programming Guide。

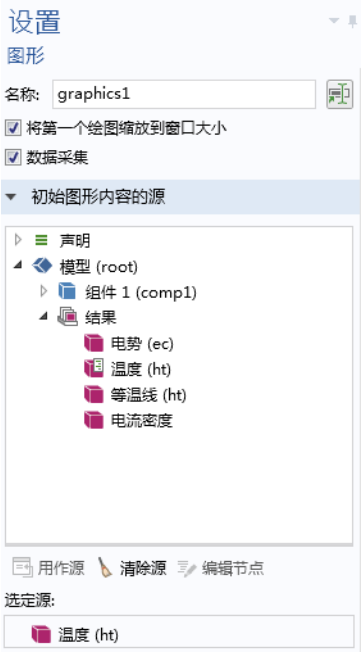
图形

每个**图形**对象在创建后都会生成默认的名称，如 `graphics1`、 `graphics2` 等。这些名称用于引用按钮、菜单项和方法命令序列中的图形对象。要在特定表单中引用图形对象，可以使用以下语法：`/form1/graphics2`、`/form3/graphics1` 等。

选择初始图形内容的源

在图形对象的**设置**窗口中，可以使用**初始图形内容的源**栏来设置要在默认情况下显示的绘图组或动画。可以单击**用作源**或双击树中的节点进行选择。如果显

示的绘图组已存在解，则在启动 App 时会显示相应解的可视化效果。下图显示了将**温度**图选作源的图形对象的**设置**窗口。



除**结果**绘图节点外，您还可以使用**动画**、**选择**、**几何**以及**网格**节点作为**选定源**。

选中**将第一个绘图缩放到窗口大小**复选框后，可以确保出现在图形窗口中的第一个绘图显示整个模型 （缩放到窗口大小）。在图形内容首次发送到图形对象时会触发该操作。

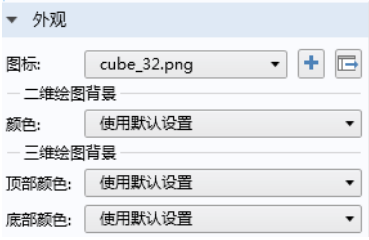
选中**数据采集**复选框可以使图形对象进行交互，以便您执行相关操作，例如，在特定点单击绘图并调用该点所在坐标处温度的数值。有关更多信息，请参见第 83 页的 “数据采集”。

外观

图形对象**设置**窗口的**外观**栏包含以下选项：

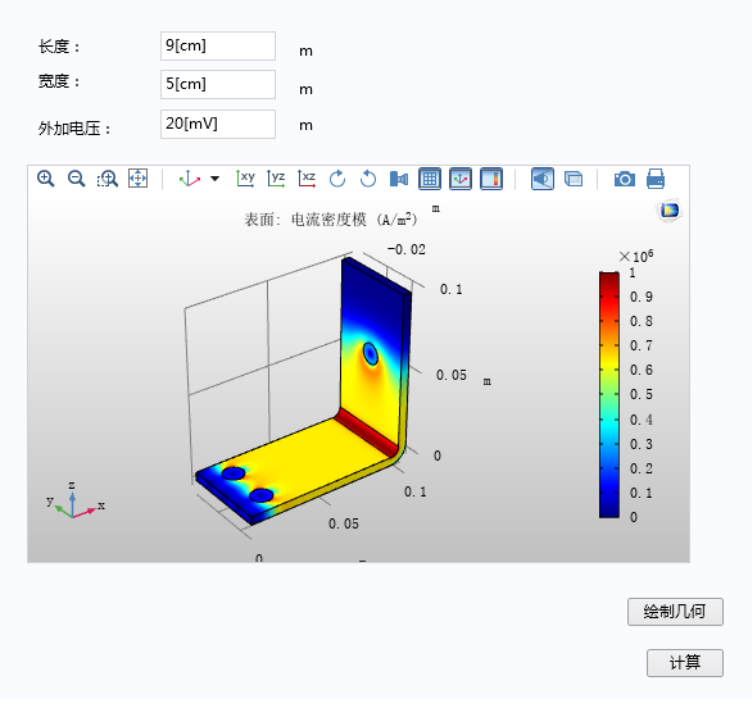
- 右上角包含**图标**，如徽标图像。

- 设置二维绘图的背景**颜色**。
- 通过选择**顶部颜色**和**底部颜色**来设置三维绘图的单色或渐变背景色。



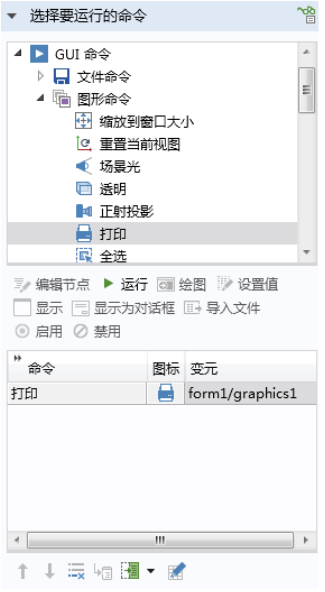
此外，**状态**子栏（上图未显示）还包含用于设置图形对象的可见和启用状态的选项。有关更多信息，请参见第 68 页的“更改表单对象的启用和可见状态”。

在下图显示的 App 中，背景的**顶部颜色**设为白色，**底部颜色**设为灰色。此外，其中没有包含标准绘图工具栏。



图形命令

在按钮等的命令序列使用的编辑器树中，**图形命令**文件夹包含用于处理或修改图形对象的命令。下图显示的命令序列包含一个用于打印图形对象内容的命令。



可用的**图形命令**包括：

- **缩放到窗口大小**
  - 将整个模型设为可见。
- **重置当前视图**
  - 将当前活动的视图重置为启动 App 时的状态；另请参见第 77 页的 “视图”。
- **场景光**
  - 切换场景光 （打开或关闭）。
- **透明**
  - 切换透明度设置 （打开或关闭）。
- **正射投影**
  - 启用正射投影 （与透视投影相反）。
- **打印**
  - 打印图形对象的内容。

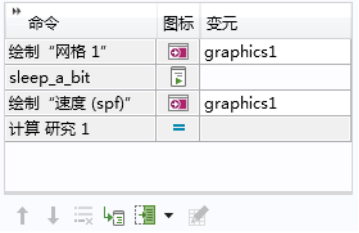
- **全选**
  - 选择所有对象。
- **清除选择**
  - 清除所有对象的选择。
- **显示选择颜色**
  - 启用选择颜色可视化功能。
- **显示材料颜色和纹理**
  - 启用材料颜色和纹理可视化功能。

请注意，**缩放到窗口大小**、**重置当前视图**、**场景光**、**透明**、**显示选择颜色**、**显示材料颜色和纹理**以及**打印**命令在标准图形工具栏中有相应的工具栏按钮。请参见后续的 “图形工具栏”一节。

求解时绘图

您可以在求解过程中绘制结果，从而让用户能够监视模型的收敛情况。在本例中，假设在**求解时显示结果**栏中启用了**绘制**选项。此选项位于模型树中**研究**节点的**设置**窗口中，如下图所示。

您可以加入一种调用内置 sleep 法的方法，以便在切换显示其他类型的图形之前先简要显示图形信息。将其插入命令序列中的绘图命令后，如下图所示。



命令	图标	变元
绘制 “网格 1”		graphics1
sleep_a_bit		
绘制 “速度 (spf)”		graphics1
计算 研究 1		

在本例中， sleep_a_bit 方法包含一行代码：

```
sleep(1000); // sleep for 1000 ms
```

有关 sleep 方法的更多信息，请参见第 324 页的 “sleep”。

在上述命令序列中，**绘制速度**命令在**计算研究**命令之前。从而确保在求解过程中，图形对象可以显示速度图。

使用多个图形对象

由于在运行 App 的平台上可能存在图形硬件限制，因此，应尽量减少所用的图形对象。这样可以确保 App 具有最大的可移植性。此外，如果您试图在 Web 浏览器中运行 App，可能对所用的图形对象数量还有一些额外的限制。不同的硬件、操作系统及 Web 浏览器组合有不同的限制。

鉴于此，不同表单中的两个名称相同的图形对象被视为两个不同的图形对象。例如， form1/graphics1 与 form2/graphics2 表示两个不同的图形对象。此外，如果一个图形对象用在子表单 （请参见第 247 页的 “表单”）中，则使用该子表单的每个图形对象都被视为不同的图形对象。

要在 App 中显示许多不同的绘图，您可以创建按钮、开关按钮或单选按钮等，它们可以绘制到未使用子表单的表单中的同一图形对象中。

如果您需要使用方法来更改绘图，可以使用 useGraphics 方法。有关编写方法的更多信息，请参见第 161 页的 “方法编辑器”。以下示例代码基于布尔变量的值重用同一图形对象，从而实现切换绘图组。

```
if (my_boolean) {
  useGraphics(model.result(“pg1”), “form1/graphics1”);
  my_boolean=!my_boolean; // logical NOT to change between true and false
} else {
  useGraphics(model.result(“pg2”), “form1/graphics1”);
  my_boolean=!my_boolean;
}
```

清除图形对象内容

您可以通过调用 useGraphics 方法来清除图形对象的内容，例如：

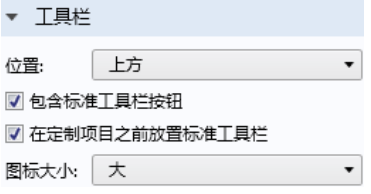
```
useGraphics(null, “/form1/graphics1”)
```

该方法可清除表单 form1 中的图形对象 graphics1 的内容。

图形工具栏

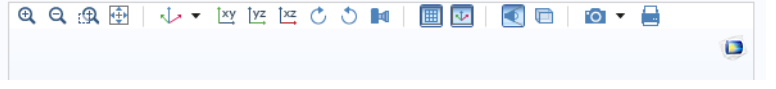
初始图形内容的源中使用的树节点类型可以确定显示的工具栏类型。根据空间维度的不同，以及引用的源是**几何**、**网格**、**选择**还是**绘图组**节点，工具栏会有所不同。例如，**绘图组**节点可以显示额外的**显示图例**按钮。

在图形对象**设置**窗口的**工具栏**部分，您可以控制是否包含图形工具栏并设置它的位置（**下方**、**上方**、**左**、**右**）。此外，您还可以为**图标大小**选择**小**或**大**选项。



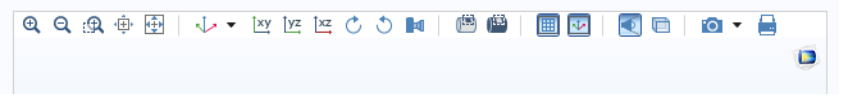
几何与网格的图形工具栏

下图显示当三维模型的**几何**或**网格**节点用作**初始图形内容的源**时，标准图形工具栏的外观。



选择的图形工具栏

当**初始图形内容的源**设为**显式**选择时，图形工具栏将包含额外三个项：**缩放到选择**、**选择框**和**取消选择框**。如下图所示。



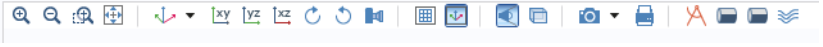
有关选择的更多信息，请参见第 80 页的 “选择”。

绘图组的图形工具栏

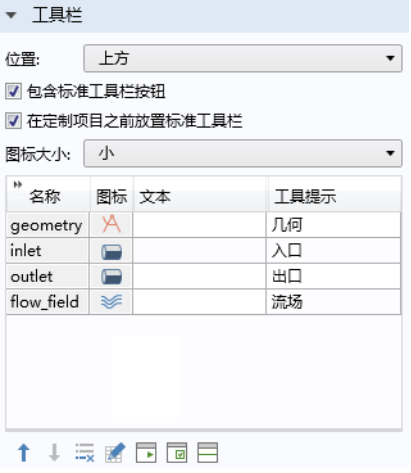
下图显示当**三维绘图组**节点用作**初始图形内容的源**时，标准图形工具栏的外观。

定制图形工具栏按钮

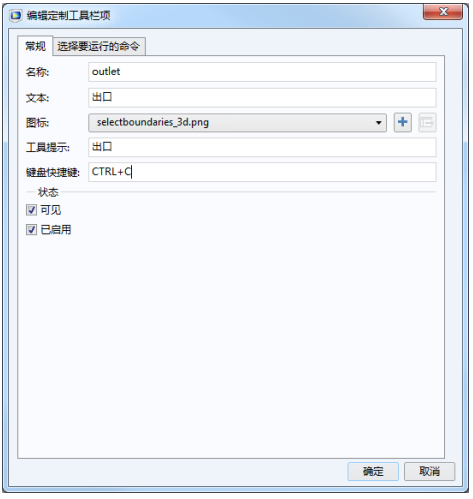
在 “工具栏” 部分，您还可以在图形工具栏中添加定制按钮。您可以使用表格下方的按钮来添加或移除定制的工具栏按钮 （项）。还可以将工具栏按钮上移或下移，并添加**分隔线**和**编辑**按钮。下图显示了结果包含四个额外按钮的标准图形工具栏，这些额外按钮显示在右侧。

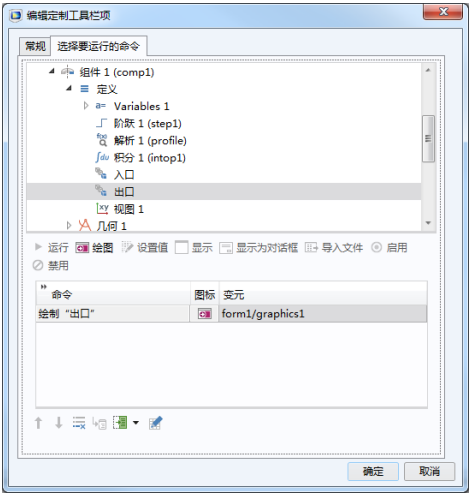


下图显示这些图形工具栏项对应的表格。



要编辑工具栏项的命令序列，可以单击**编辑**按钮，以打开**编辑定制工具栏项**对话框。

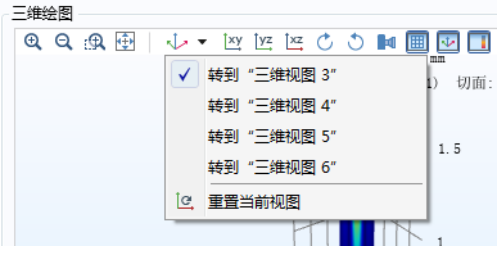




此对话框包含的设置与按钮或工具栏项的类似，并根据该项是否为切换项，将内容分隔为两个或三个选项卡。有关详细信息，请参见第 57 页的 “按钮” 和第 286 页的 “工具栏”。

视图

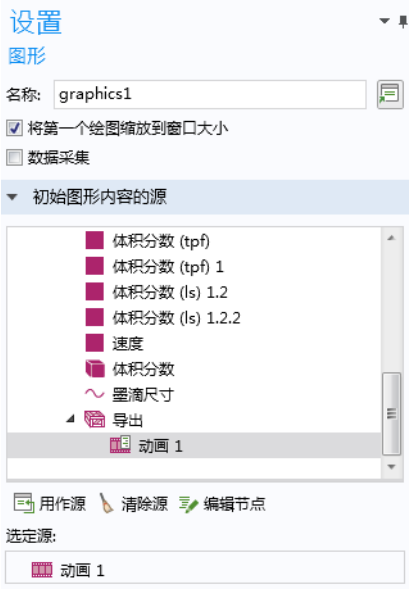
在图形工具栏中，**切换到默认视图**按钮 （仅适用于三维图形）将显示一个菜单，其中列出所有可用的视图。当前处于活动状态的视图前有一个复选标记。



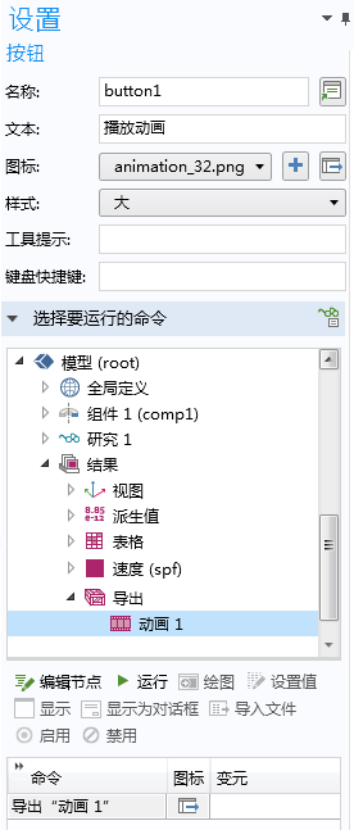
除了包含所有视图的列表，还有一个**重置当前视图**选项，用于将当前活动的视图重置为 App 启动时的状态。

动画

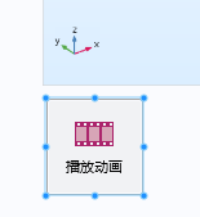
通过将**结果 > 动画**节点用作**初始图形内容的源**，可以在 App 中显示动画。



要运行动画，可以使用 “新建表单” 向导或 “编辑器工具” 窗口，通过运行**结果 > 动画**节点的按钮等来创建命令。



在使用 “新建表单” 向导或 “编辑器工具” 时，动画按钮的默认外观如下所示：

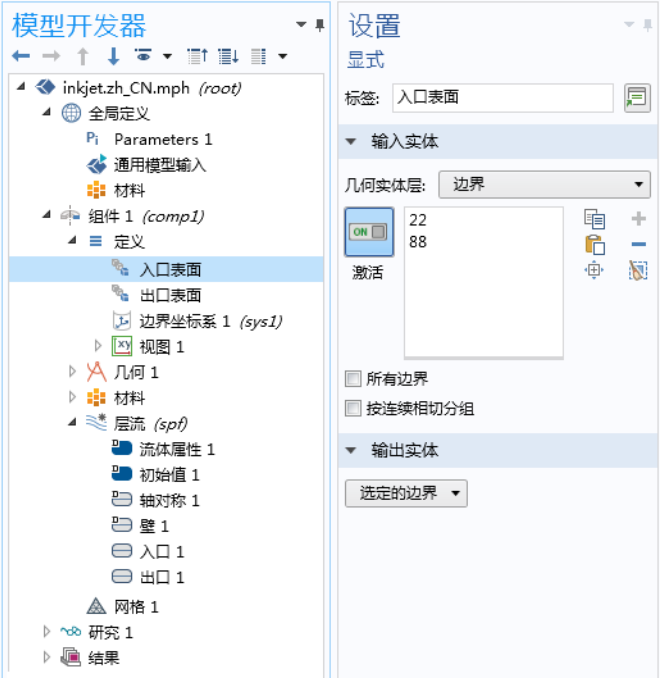


选择

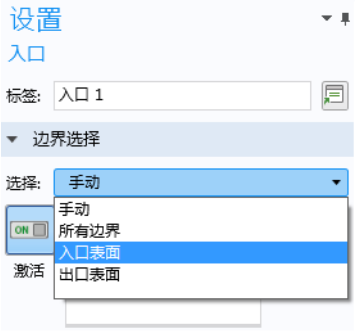
模型开发器中的选择

在 “模型开发器”中，已命名的选择可供您在指派材料属性、边界条件及其他模型设置时，对域、边界、边或点进行分组。您可以在**组件 > 定义**节点下添加子节点，来创建不同类型的选择。这些选择可以在整个模型组件中被重用。

我们以边界条件的选择为例，来阐述选择的使用方式。当您选择将哪些边界与特定的边界条件关联时，可以直接在 COMSOL Desktop 环境的图形窗口中单击这些边界。这是选择的默认选项，称为**手动**选择 （如下所示）。随后，这些边界会添加到局限于该边界条件的选择中。命名的选择支持您定义可在多种不同的边界条件中重用的全局选择，只需从下拉列表中选择即可。下图显示一个名为**入口表面**的**显式**选择，其中包含两个关联边界 （22 和 88）。



下图显示**入口**边界条件的**设置**窗口，其中使用了**入口表面**选择。在本例中，还包含一个 “出口表面” 选择。

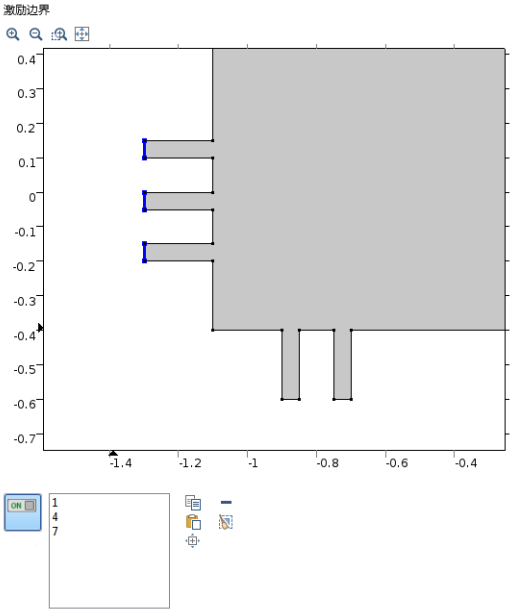


为方便起见，除了**手动**选项，还提供**所有边界**的快捷方式。

App 开发器中的选择

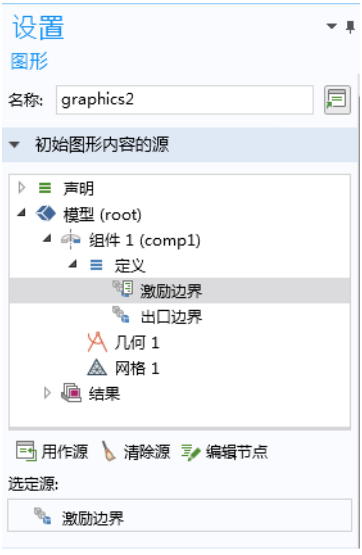
显式选择类型可供您基于实体编号对域、边界、边或点进行分组，这也是最适用于 “App 开发器” 的选择类型。您可以通过**选择输入**对象或**图形**对象来允许 App 用户交互更改属于**显式**选择的实体。在下面的示例中，嵌入模型包含通过

显式选择定义的边界条件。同时使用**选择输入**对象和**图形**对象，使用户能够选择要由入射波激励的边界。



用户可以通过以下方式来选择边界：直接在对应于**图形**对象的图形窗口中单击，或者在对应于**选择输入**对象的边界编号列表中添加几何实体编号。

为了能通过单击边界直接将其选中，您可以将图形对象链接到**显式**选择，如下图所示。选中**显式**选择并单击**用作源**。在下图中，有两个**显式**选择：**激励边界**和**出口边界**，并且图形对象 graphics2 已链接到**激励边界**选择。



当图形对象以这种方式直接链接到选择时，该图形对象会显示几何，用户可以通过单击边界与之进行交互。这些边界随后会添加到相应的选择中，或从中移除。

为了能按编号进行选择，您可以将**选择输入**对象链接到显式选择。有关更多信息，请参见第 268 页的 “选择输入”。

通过**编辑器工具**窗口，您可以快速添加已链接至**显式**选择的**图形**对象或**选择输入**对象。右键单击编辑器树中的**显式**选择节点可以获取这些选项。

您可以将全局**事件**设为通过**显式**选择触发。从而可以在用户单击几何对象、域、面、边或点时运行命令序列或方法。有关使用全局事件的更多信息，请参见第 129 页的 “事件”和第 132 页的 “数据更改事件的源”。

数据采集

在图形对象的**设置**窗口中，选中**数据采集**复选框可以使图形对象进行交互，以便您执行相关操作，例如，在特定点单击绘图并调用该点所在坐标处温度的数值。在下图显示的**数据采集的目标**栏中，选中了标量双精度变量 Tvalue。此变

量在**声明**节点下声明。在运行的 App 中，指针位置所在的温度值将存储在变量 `Tvalue` 中。

如果**数据采集的目标**是一维双精度数组，则存储的值将对应于单击位置的 x,y（二维）或 x,y,z 坐标。

数据采集的目标可以是以下任意一种：

- 标量双精度变量
- 一维双精度数组
- 域点探针

- 边界点探针
- 图形数据声明

有关 “图形数据”声明的更多信息，请参见第 157 页的 “图形数据”。

输入框

输入框可供用户更改参数或变量的值。在 “新建表单”向导中选择参数或变量时，会创建三个表单对象：

- **文本标签**对象，用于参数或变量描述。
- **输入框**对象，用于输入值。
- **单位**对象 （如果适用），含度量单位。

使用**编辑器工具**窗口选择参数或变量时，会创建相同的三个表单对象。

假设您不使用**编辑器工具**窗口：要插入额外的输入框，可以使用功能区中的**插入对象**菜单并选择**输入框**。在 “表单编辑器”中，可以从**源**栏的树中选择输入

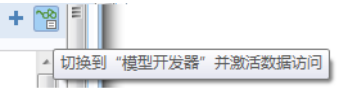
框来将其链接至特定的参数或变量，然后单击**用作源**。您还可以在**设置**窗口的**源**栏中设置**初始值**。下图显示输入框的**设置**窗口。

除参数和变量以外，输入框也可以使用**信息**节点作为**源**。

初始值的默认设置为**来自数据源**。这意味着如果源是参数，则输入框中显示的初始值与“模型开发器”的**参数**节点中指定的参数值相同。另一个**初始值**选项是**定制值**，支持将初始值设为与源不同的值。如果清除**可编辑**复选框，则 App 将显示**初始值**，且该值无法更改。

您可以添加带文本的**工具提示**，当您将鼠标指针悬停在输入框上方时可以显示文本内容。

源栏的标题包含多个按钮，可供您轻松访问相关工具，这些工具可用于将更多属性和变量设为输入框的源。

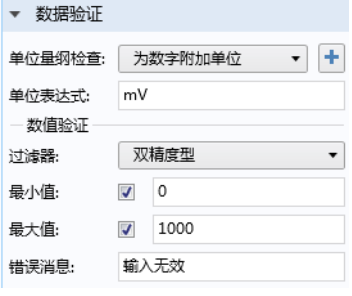


新建声明并将其用作源按钮可用于在**声明**节点下添加新变量。有关更多信息，请参见第 136 页的 “声明”。**新建表单声明并将其用作源**按钮可用于在表单的**声明**节点下添加新的局部变量，如下所示。

切换到 “模型开发器” 并激活数据访问按钮可用于访问下层模型属性，如下一节中所述。有关**数据访问**的更多信息，请参见第 96 页的 “表单编辑器中的数据访问”。

数据验证

输入框**设置**窗口的**数据验证**栏可供您对用户输入的单位和值进行验证。

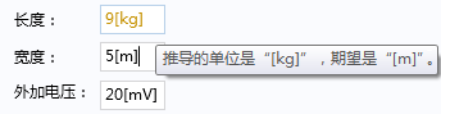


在 “新建表单”向导中创建输入框时，可以在适当情况下使用**为数字附加单位**设置。此设置假设用户在输入框中输入一个数字后，系统便可以处理通过 COMSOL 方括号 [] 单位语法在数字后添加单位的情况。如果**单位表达式**为 mm，则可以为 1[mm]，也可以为任意长度单位，例如 0.1[cm]。不兼容的单位类型将显示**错误消息**。表达式为 1.23[mm] 的参数在用作源的情况下，其附加单位是 mm，且编辑框中显示的初始值为 1.23。

单位量纲检查列表包含以下选项：

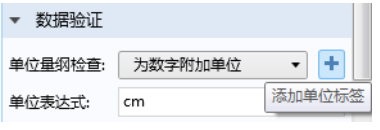
- **无**
- **与物理量兼容**
- **与单位表达式兼容**
- **为数字附加单位** （默认设置）
- **附加来自单位集的单位**

当 App 用户输入不兼容的单位 （即无法转换为**数据验证**设置中的指定单位的度量单位）时，值或表达式会以橙色突出显示并发出警告。通过选择**与物理量兼容**或**与单位表达式兼容**选项可以启用此特征。此外，用户还会看到一个工具提示，指出单位不匹配，如下图所示。



如果出现单位不匹配，并且 App 不执行进一步的错误控制，则输入的表达式的数值将转换为默认单位。在上图中， 9[kg] 将转换为 9[m]。

在**单位量纲检查**列表右侧有一个**添加单位标签**按钮。如果输入框右侧尚未放置单

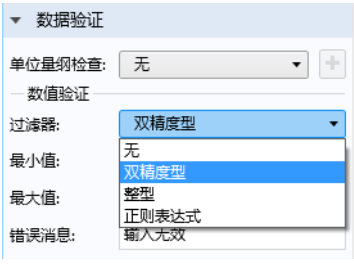


位标签，则单击此按钮可以在此添加一个单位标签。

无选项不提供单位验证。

数值验证

通过**为数字附加单位**、**附加来自单位集的单位**和**无**选项，您可以使用过滤器对输入数字进行数值验证。



无选项的**过滤器**列表包含以下选项：

- **无**
- **双精度型**
- **整型**
- **正则表达式**

为数字附加单位和**附加来自单位集的单位**选项的**过滤器**列表仅包含**双精度型**和**整型**选项。

双精度型和**整型**选项可基于**最小值**和**最大值**来过滤输入。如果输入超出这些值的范围，则会显示**错误消息**。您可以在这些编辑框中使用全局参数。如果使用全局参数，则在定义此类参数时，可以包含单位也可以不含单位。如果使用无单位的全局参数，当它们用作**最小值**和**最大值**时，系统只需获取这些参数的数值。例如，考虑对长度参数 L （单位为 cm）的输入框进行数据验证。并进一步假设将全局参数 Lmax 用作**最大值**。如果您希望 L 的最大值为 15 cm，则参数 Lmax 可以使用以下值：15 （无单位）、 15[cm]、 0.15[m]、 150[mm] 等。

对于**附加来自单位集的单位**选项，**最小值**和**最大值**始终与单位集所设单位的**初始值**相关。有关单位集的更多信息，请参见第 150 页的 “单位集”。

正则表达式选项在**单位量纲检查**设为**无**时显示，可供您使用正则表达式来匹配输入字符串。有关正则表达式的更多信息，请参见动态帮助。单击窗口右上角的帮助图标并搜索 “regular expression”。

如有更高级的需求，请注意，对输入框内容的所有验证几乎都可以通过调用方法来实现，该操作可以使用输入框**设置**窗口中的**事件**栏执行。

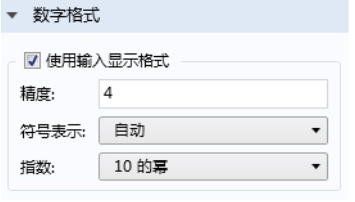
错误消息

您可以定制**错误消息**显示的文本。在开发和调试 App 的过程中，有时很难判断此类错误的来源。因此，可以使用**测试 App** 来显示附加调试信息，如下图所示。

调试信息通常包含表单对象的类型和路径，以及失败的原因；例如 5<=x<=10。在使用“运行 App”或 COMSOL Server 启动 App 时，不会显示任何额外信息。

数字格式

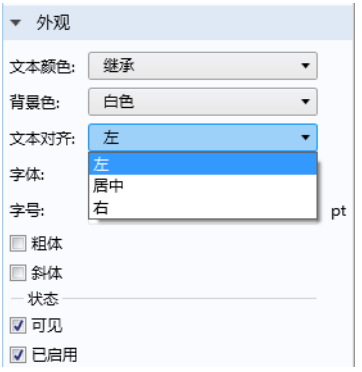
数字格式栏包含**使用输入显示格式**复选框。选中此复选框时，将启用与**数据显示**对象相同类型的显示格式。



有关更多信息，请参见第 93 页的 “数据显示”。

外观

除颜色和字体设置外，输入框的**外观**栏还包含**文本对齐**设置，可将文本设为**左侧**、**中心**或**右侧**对齐。



输入框是**可见**还是**已启用**，可以通过**状态**子栏中的复选框来控制。有关更多信息，请参见第 68 页的 “更改表单对象的启用和可见状态”。

单位

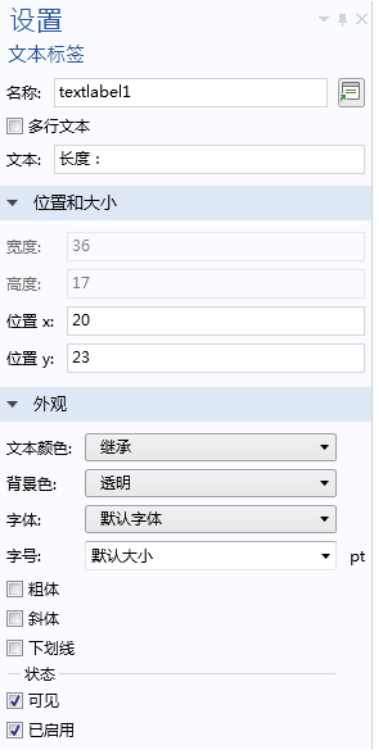
在**单位**对象的**设置**窗口中，您可以将单位设为固定字符串，也可以将其链接到源，如输入框。下图显示单位对象的**设置**窗口。

在使用 “新建表单” 向导添加输入框时，会在适用的情况下自动添加单位对象。默认情况下，单位通过 Unicode 渲染来显示。此外，您也可以选中 **LaTeX 标记**复选框来使用 LaTeX 渲染。此时，单位的显示将与选定的字体无关。

文本标签

文本标签对象仅用于在表单中显示文本。在使用 “新建表单” 向导添加输入框时，会自动添加**文本标签**对象，其中显示关联参数或变量的描述文本。窗口中

有一个**多行文本**复选框。当它处于选中状态时，会启用**自动换行**复选框。下图显示**文本标签**对象的 “设置” 窗口。



要插入附加的**文本标签**，可以使用功能区中的**插入对象**菜单并选择**文本标签**。**位置和大小**栏的内容将发生更改，具体取决于您是在草图布局模式还是栅格布局模式下工作。

数据显示

数据显示对象用于显示标量和数组的数值。如果有关联单位，它将作为**数据显示**对象的一部分显示出来。

源

在数据显示对象的**设置**窗口中，从**源**栏的模型树中选择一个节点。然后单击**用作源**按钮，如下所示。有效的参数、变量和属性包括：

- **派生值**节点的输出，如**全局计算**或**体最大值**节点
- 在**声明** > **标量**、**一维数组**和**二维数组**节点下声明的变量
- 使用**数据访问**工具启用的属性；请参见第 96 页的 “表单编辑器中的数据访问”
- 以下**信息**节点变量之一，它们位于根节点和每个 “研究”节点下：
  - **预计计算时间**
    这是您在根节点**设置**窗口的**期望**框中输入的值。
  - **上次计算时间** （根节点下）
    这是上次针对上一个计算研究所测量的计算时间。
  - **上次计算时间** （研究节点下）
    这是上次针对该研究所测量的计算时间。

  当您第一次启动 App 时，系统会重置上次测量时间，显示**尚不可用**。

使用新建表单向导生成数据显示对象

在 “新建表单”向导的**输入 / 输出**选项卡中，仅**派生值**节点会生成**数据显示**对象。**声明**下的变量和**数据访问**启用的常数将生成**输入框**对象。

选中**派生值**节点时，系统会基于对应的**派生值**节点变量创建两个表单对象：

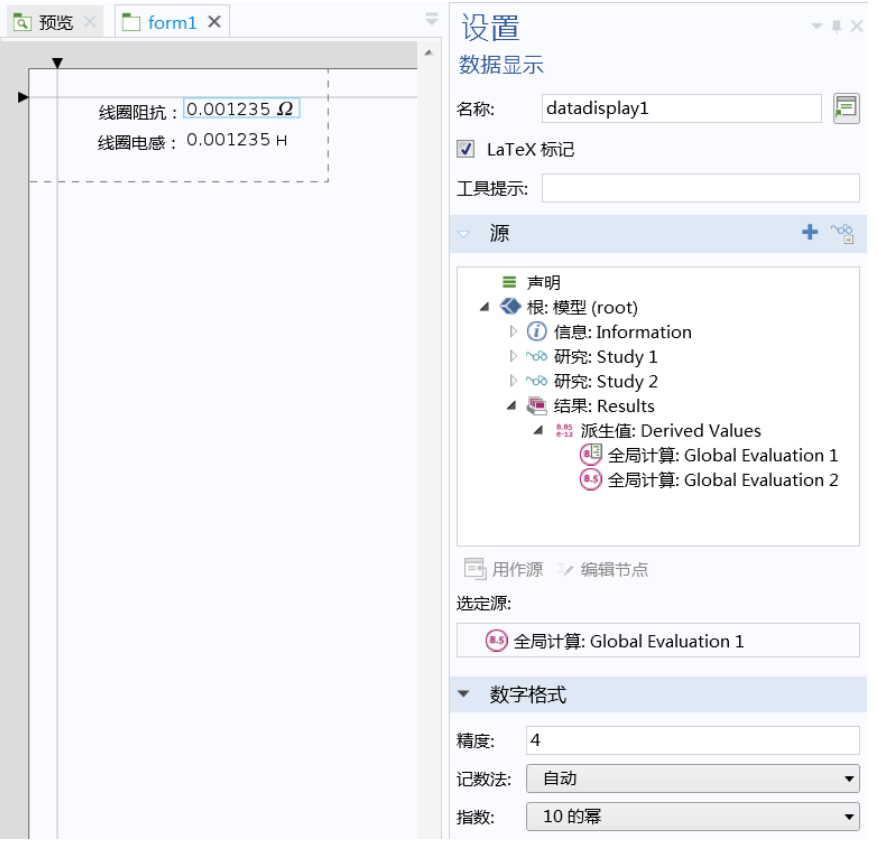
- 用于变量**描述**的**文本标签**对象
- 用于变量值的**数据显示**对象

您随后可以编辑这些表单对象的设置。要插入附加的数据显示对象，可以使用功能区中的**插入对象**菜单并选择**数据显示**。

数字格式

数字格式栏可供您设置**精度**、**符号表示**和**指数**。

下图显示的示例包含变量线圈电阻和线圈电感的数据显示对象。在适用的情况下，格式化的单位标签会自动显示为该对象的一部分。



渲染方法

默认情况下，数据显示对象的单位通过 Unicode 渲染进行显示。此外，您也可以选中 **LaTeX 标记**复选框来使用 LaTeX 渲染。此时，数据显示与选定的字体无关。

仅 LaTeX 渲染支持数组和矩阵的格式化显示。下图中的二维双精度数组 （请参见第 145 页）使用选中了 **LaTeX 标记**的**数据显示**对象进行显示。

$$\begin{bmatrix} 0 & 0 & 0.9 & 0.8 & 0.7 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0.5 & 0.7 & 1.5 & 0.8 & 0.6 & 0.3 & 0.2 & 0.1 & 0.1 & 0.1 \\ 0 & 0 & 0.9 & 0.8 & 0.7 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

您可以添加带文本的**工具提示**，当您将光标悬停在数据显示对象上方时可以显示文本内容。

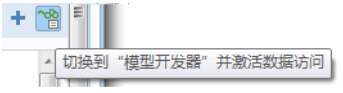
表单编辑器中的数据访问

许多类型的表单对象的**设置**窗口都有一个包含树结构的栏，您可以从中选择节点，这些树结构包括模型树或其中的一部分，以及 App 树的一部分。例如，输入框的**源**栏或按钮的**选择要运行的命令**栏。默认情况下，模型树和 App 树中有许多属性都未启用，这是因为模型中可能包含成百上千个属性，显示这些属性的完整列表不利于操作，也没有必要。不过，您可以通过**数据访问**技术将这些“隐藏的”属性用于 App。

本节将以输入框和按钮为例，介绍**数据访问**的用法。

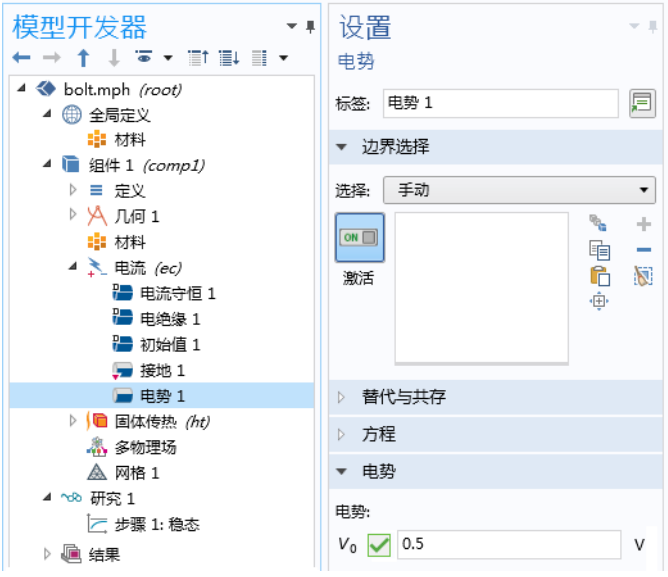
输入框的数据访问

默认情况下，您可以将输入框链接到模型树的**参数**或**变量**节点下定义的参数和变量，以及链接到 App 树的**声明**节点下声明的变量。要访问其他模型树节点属性，可以在输入框**设置**窗口中单击**源**栏的标题中的**切换到“模型开发器”并激活数据访问**按钮，如下图所示。

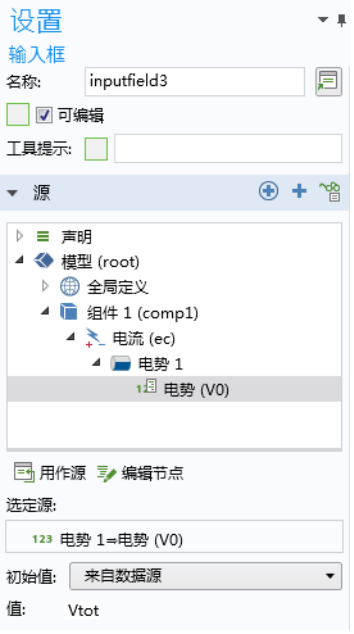


您也可以从“模型开发器”的**开发工具**选项卡中的 **App** 组进行访问。

此时，当您单击模型树节点时，会在各个设置旁显示复选框。在下图中，**电势**边界条件的复选框处于选中状态：



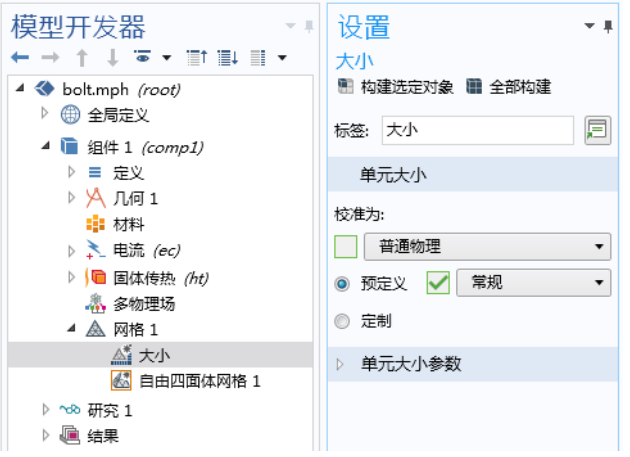
下图显示输入框的**设置**窗口，其中的源列表现在包含**电势**。



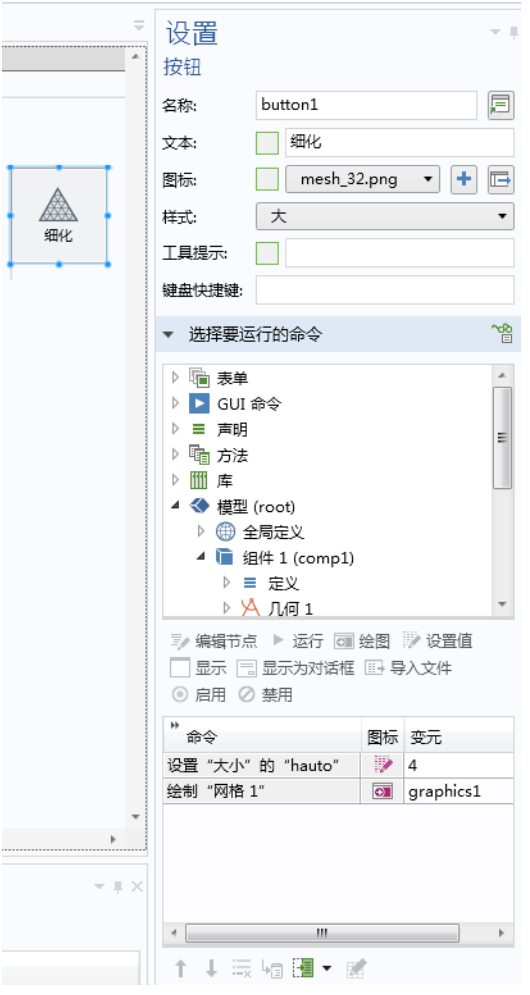
此外，如上图所示，**数据访问**支持访问输入框表单对象的**可编辑**复选框和**工具提示**文本。除“模型开发器”设置外，**数据访问**还支持访问“App 开发器”的某些属性。

数据访问可用于为按钮设置参数、变量或模型属性的值。例如，您可以为预定义的网格单元大小创建按钮。当**网格**节点**设置**窗口中的**序列类型**设为**用户控制网**

格时，可以使用下图显示的设置：在本例中，**单元大小**的**预定义**属性已经可用并处于选中状态。



下图显示用于创建网格的按钮的**设置**窗口，其中**单元大小 > 预定义**已设为**细化**。



在上述示例中，**设置值**命令用于设置**预定义网格大小 (hauto)** 属性的值。**预定义网格大小 (hauto)** 属性对应于之前显示的**大小**节点中的以下设置：

预定义网格大小	值
极细化	1
超细化 - 超粗化	2 - 8
极粗化	9

hauto 属性的值为双精度型，可以取任何正值。对于非整数值，线性插值用于定制的网格参数。例如，您可以将滑块对象设为调整预定义网格大小。有关滑块对象的更多信息，请参见第 282 页的 “滑块”。

通常，对于单独的模型树属性，您可以通过在更改值时录制代码，然后检查自动生成的代码，从而快速确定这些属性的许用值。有关更多信息，请参见第 171 页的 “录制代码”。

您还可以使用组合框对象来直接访问从**极细化**到**极粗化**的所有选项。有关更多信息，请参见第 215 页的 “组合框”。

数据访问汇总

下表汇总了表单对象和事件，以及菜单、工具栏和功能区项中支持**数据访问**的位置。

表单对象、事件或项	设置窗口中的栏
输入框	源
按钮	选择要运行的命令
切换按钮、菜单切换项和功能区切换项	“源”和 “选择要运行的命令”
复选框	源
组合框	源
数据显示	源
图形 （图形工具栏项）	选择要运行的命令
表单集合	活动窗格选择器 平铺或分页
卡片堆	活动卡片选择器
信息卡片堆	活动信息卡选择器
单选按钮	源
文本	源
列表框	源
滑块	源
工具栏 （工具栏项）	选择要运行的命令
菜单项	选择要运行的命令
功能区项	选择要运行的命令
事件 （全局）	选择要运行的命令 数据更改事件的源

全局事件、菜单、功能区或工具栏项均在**设置**窗口中提供**选择要运行的命令**栏，具备上述按钮相关栏的功能。全局事件和许多表单对象在**设置**窗口中提供了**源**栏，具备上述输入框相关栏的功能。有关全局事件、菜单、功能区和工具栏项的信息，请参见第 75 页的 “图形工具栏”、第 122 页的 “主窗口”、第 129 页的 “事件”、第 277 页的 “表格” 以及第 286 页的 “工具栏”。

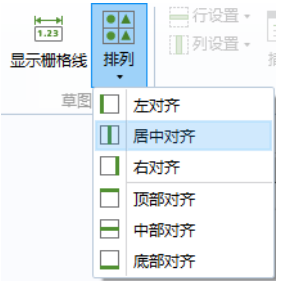
草图和栅格布局

“表单编辑器” 提供两种布局模式用于定位表单对象：草图布局模式和栅格布局模式。默认设置为草图布局模式，可供您为对象输入固定位置和大小 （以像素计）。使用栅格布局模式可基于含单元格的背景栅格来确定对象的位置和大小。在栅格布局模式下，表单被分为许多交叉的行和列，每个交叉区域最多只有一个表单对象。您可以使用这种布局模式来设计可调整大小的用户界面，例如设计一个可在多个平台的 Web 浏览器中运行的 App。

草图布局

通过单击功能区的**布局**组中的**草图**或**栅格**按钮，可以在草图与栅格布局模式之间进行切换。

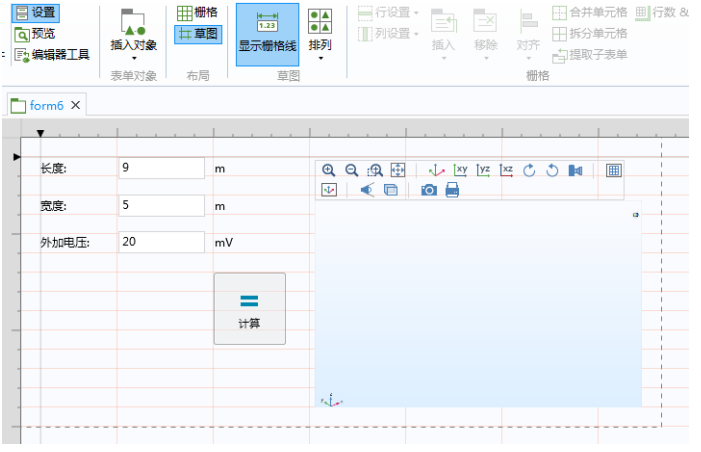
表单选项卡中的**草图**组包含两个选项：**显示栅格线**和**排列**。**排列**菜单可供您将表单对象组相互对齐。



草图栅格

显示栅格线选项可以显示草图栅格，用于对齐各个对象。请注意，草图布局模式中使用的栅格与栅格布局模式中的栅格不同。草图布局模式的默认设置不显示栅格线。如果在不显示栅格线的情况下拖动表单对象，该对象会相对于其他表单对象的位置进行对齐。

如果选中**显示栅格线**选项，则在拖动表单对象时，其左上角会与栅格线的交点对齐。



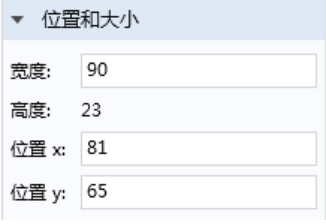
在表单的**设置**窗口中，您可以更改草图栅格的设置：

- **列宽**
- **行高**
- **将栅格与边距对齐**

- **对齐区域**
  - 您可以使用滑块在**小**到**大范围内更改对齐区域的大小**
- **仅与栅格对齐**
  - 清除此复选框后，可同时与栅格和其他表单对象的位置对齐

位置和大小

草图布局模式以像素为基础，表单对象的定位通过该表单对象左上角相对于屏幕左上角的坐标来表示。 *x* 坐标会随着对象在屏幕中右移而增加， *y* 坐标则随着对象下移而增加。您可以在表单对象**设置**窗口的**位置和大小**栏设置它的绝对位置。



系统会根据需要为表单对象分配空间，或根据**宽度**和**高度**值进行指定。表单对象可以重叠。

按钮和切换按钮表单对象包含**自动**和**手动**选项，用于设置**宽度**和**高度**值。**手动**选项支持基于像素的输入，**自动**选项可以使按钮大小适合**文本**字符串的大小。

栅格布局

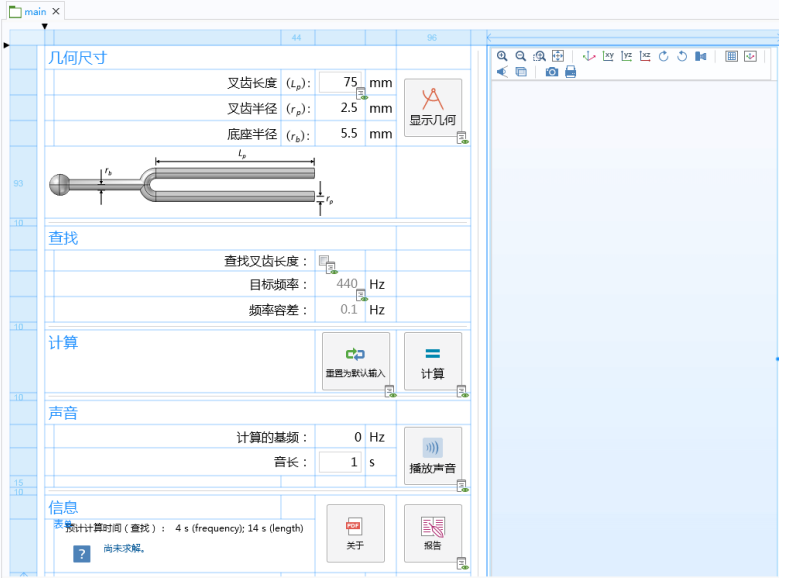
通过单击功能区的**布局**组中的**栅格**按钮，可以切换到栅格布局模式。

功能区的**栅格**组中的按钮和菜单可供您轻松访问多个命令来执行以下操作：

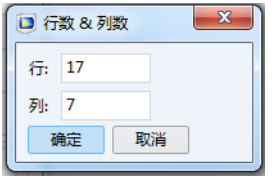
- 通过切换**适应**、**增长**和**固定**来更改行与列的增长规则，这些规则用于确定当用户界面发生调整时的布局 （**行设置**和**列设置**）。
- 插入或移除行和列 （**插入**和**移除**）。
- 在栅格单元格中对齐表单对象 （**对齐**）。
- 合并和拆分单元格 （**合并单元格**和**拆分单元格**）。
- 将矩形阵列的单元格提取为子表单，并将其插入新表单 （**提取子表单**）。
- 定义行数和列数 （**行数 & 列数**）。

表单设置窗口和栅格

在切换到栅格布局模式后，表单窗口会显示蓝色栅格线。



您可以单击功能区中的**行数 & 列数**按钮来定义行数和列数。



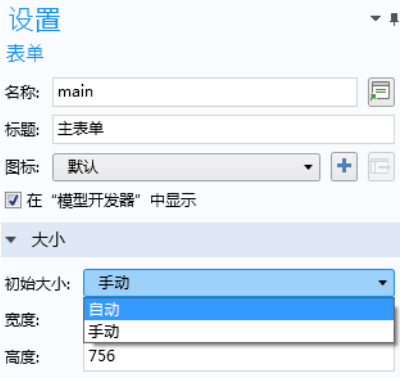
设置窗口中的**所含表单对象的栅格布局**栏显示了列宽和行高。

要交互式选择“表单编辑器”中显示的表单，可以单击该表单的左上角。

此时会显示一个蓝色框架。要交互式更改表单的整体大小，可以拖动其右侧和底部边框。在执行此操作时，您无需选定表单。

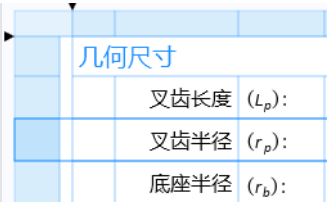
请注意，如果您从草图切换为栅格布局模式，所有的行和列都将使用**适应**设置，并且不会显示框架的控制点。如果任意行和列的**高度**或**宽度**设为**增长**，则框架会显示控制点，用于分别在垂直或水平方向调整大小。

仅当表单的**初始大小**设为**自动**时，交互式调整的框架的大小才会影响其初始大小。如果**主窗口**的**初始大小**设为**使用主表单大小**，则框架的大小还会影响主窗口的初始大小。

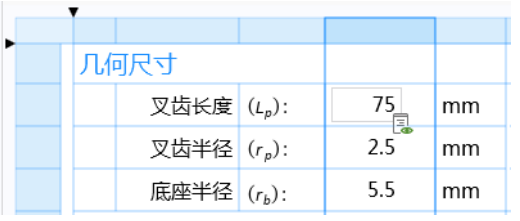


行和列

单击某个行最左侧的单元格将其选中。最左侧的单元格只用于选择行，无法在此插入表单对象。在选定行时，会在功能区选项卡中启用**行设置**菜单以及**插入**和**移除**命令。下图表明第四行已突出显示。



同样，要选择列，可以单击顶部单元格。此单元格无法包含任何表单对象。下图表明第三列已突出显示。在这种情况下，会启用功能区选项卡中的**列设置**菜单。



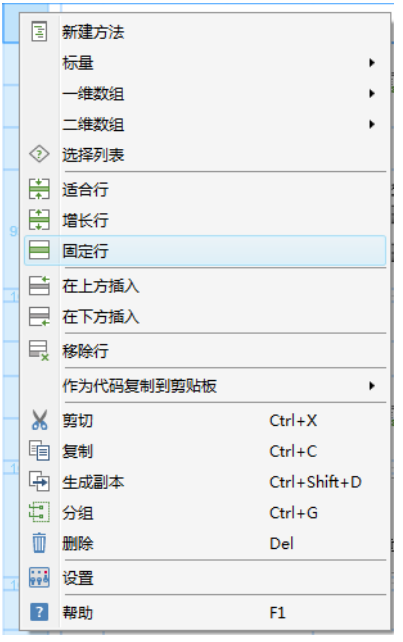
行设置与**列设置**具有三个相同的选项：

- **适应**，可将行高或列宽设为适合该行或列的表单对象大小的最小值。
- **增长**，可将行高或列宽设为根据表单的总大小按比例增长。
- **固定**，为行高或列宽的像素数设置固定值。

您可以通过拖动栅格线来交互式更改行高和列宽。

这时会显示像素的数量，并且**行设置**或**列设置**增长策略将自动改为**固定**。

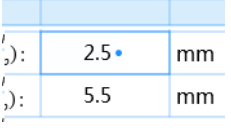
除了从功能区更改**行设置**或**列设置**，您也可以右键单击行或列，然后从关联菜单中进行选择。



右键单击行或列时显示的菜单也包含用于执行以下操作的选项：插入、移除、复制、粘贴以及为行或列生成副本。

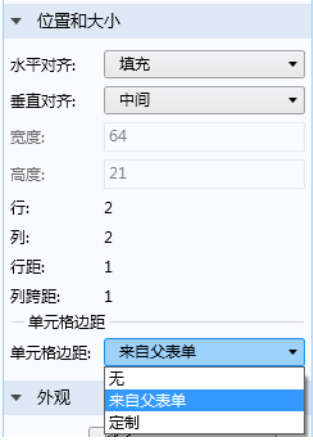
单元格

单击一个单元格将其选中。选中的单元格会以深蓝色的栅格线突出显示。



您可以选择**合并单元格**和**拆分单元格**来调整表单对象的单元格大小和布局。

在栅格布局模式下，您可以指定表单对象与所含单元格的边框之间添加的边距。



在表单对象的**设置**窗口中，**位置和大小**栏的**单元格边距**下拉列表包含以下选项：

- **无**
  - 无单元格边距
- **来自父表单** （默认设置）
  - 即在表单的**设置**窗口中指定的边距；请参见第 115 页的 “继承列和单元格边距”
- **定制**
  - 定制仅用于此表单对象的边距

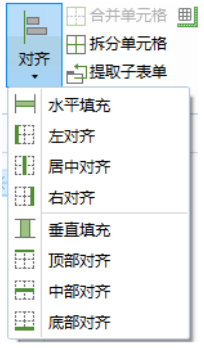
如果**水平对齐**或**垂直对齐**设为**填充**，并且列或行的增长策略支持调整表单对象的大小，则可以分别指定最小宽度或高度。最小大小可以设为**手动**或**自动**。**手动**选项可供您为最小大小指定像素值。除非表单对象内容需要较大的值，否则**自动**选项可将最小大小设为零像素。最小大小设置用于在运行时确保在表单对象缩小到最小大小之前显示滚动条。

根据单元格中包含的表单对象的类型，**宽度**和**高度**值可以设为**自动**或**手动**，如第 104 页的 “位置和大小”一节所述。

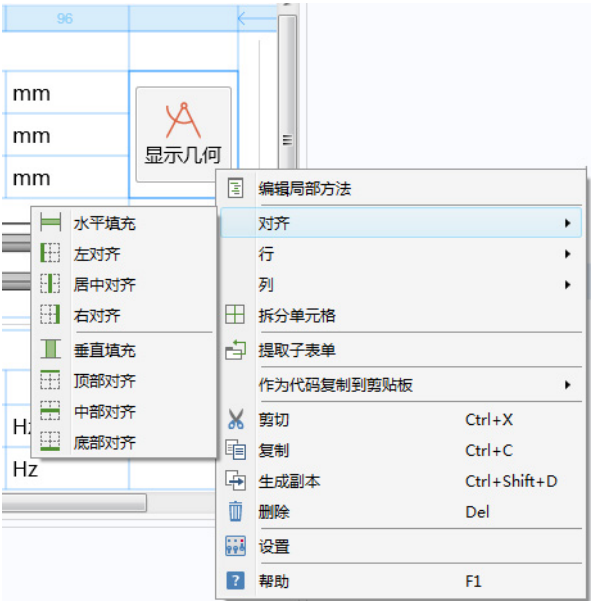
您可以单击并拖动一个橡皮框来选择多个单元格。

对齐表单对象

对齐菜单包含多个选项，用于在单元格内对齐表单对象。您也可以让表单对象在水平或垂直方向动态填充单元格。

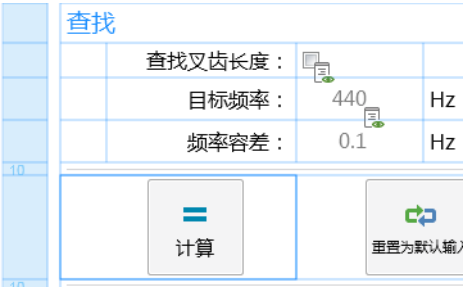


此外，您还可以右键单击表单对象，从上下文菜单中选择这些选项。



拖放表单对象

您可以通过拖放表单对象来移动它们。单击一个表单对象将其选中，然后将它拖到另一个不包含其他表单对象的单元格中。

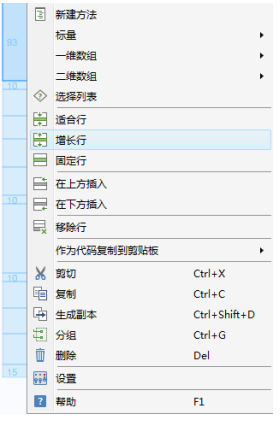


如果您将该对象拖到已被占用的单元格，则两个对象会交换位置。

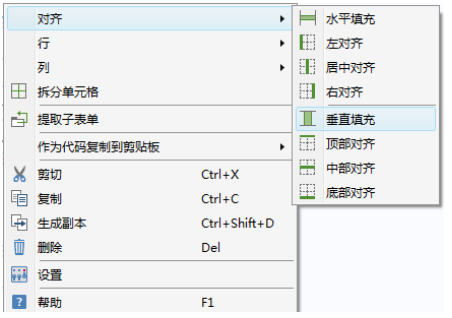
自动调整图形对象的大小

要将 App 的图形对象设为可调整大小，请遵循以下步骤：

- 将包含图形对象的表单的布局模式从草图改为栅格布局模式。
- 将包含图形对象的任何行的**高度**设置改为**增长**。为此，可以单击目标行最左侧的列。然后，在表单**设置**窗口中更改**高度**设置。或者，也可以右键单击目标对象并选择**增长行**。



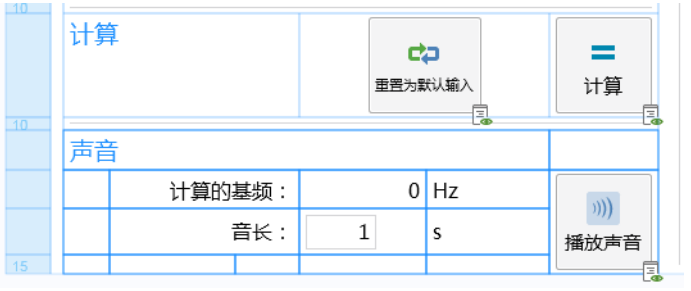
- 将包含图形对象的任何列的**宽度**改为**增长**。为此，可以右键单击目标列最上方的行，然后选择**增长列**。
- 选择图形对象，然后将**水平对齐**和**垂直对齐**都改为**填充**。您可以在**设置**窗口中执行此操作，也可以右键单击图形对象，然后选择**对齐 > 水平填充**和**对齐 > 垂直填充**。



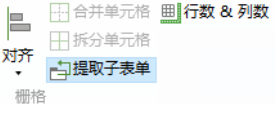
按照以上步骤，您可以通过执行栅格布局模式操作轻松将图形对象设为可调整大小，例如通过添加空行和空列，以及合并单元格等。如果您已处于栅格布局模式，则图形对象在两个方向均默认设为**填充**。

提取子表单

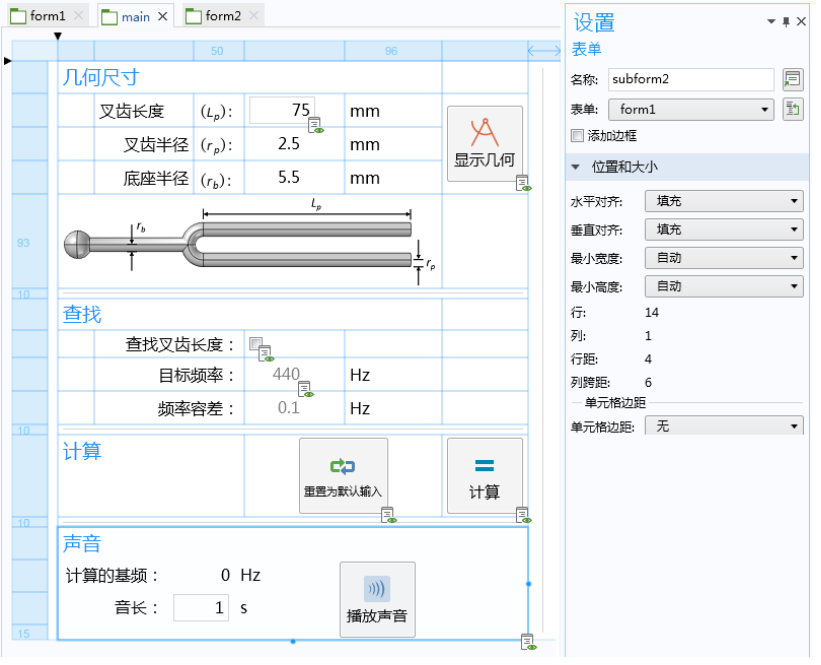
您可以在表单中选择一个矩形单元格阵列，并将它移到新表单。首先，使用 Ctrl+ 单击或 Shift+ 单击来选择单元格。



然后，单击功能区中的 “提取子表单”按钮。



此操作可以创建一个包含所选单元格的新表单，并将原始单元格替换为**表单**类型的表单对象。在子表单的**设置**窗口中，**表单**引用指向包含原始单元格的新表单。



继承列和单元格边距

您可以通过使用子表单来组织用户界面，例如将多个输入表单进行分组。下图显示了正在运行的 App 的一部分，其中包含两个子表单：**梁尺寸**和**钢筋条**。

有关将子表单添加到表单的更多信息，请参见上一节和第 247 页的 “表单”。

在垂直对齐子表单时 （如上例），建议您确保所有列的宽度相等。为此，可以使用子表单**设置**窗口中的**继承列**选项。下图显示两个子表单的一部分**设置**窗口：一个是**梁尺寸**子表单 （左图），**名称**为 `geometry_beam`，另一个是**钢筋条**

子表单（右图），**名称**为 geometry_rebars。 geometry_rebars 子表单的**继承列**已设为 geometry_beam。

在**单元格边距**子栏中，您可以指定在表单对象与所含单元格边框之间添加的**水平**和**垂直**边距。如果表单中包含的所有表单对象的**单元格边距**已设为**来自父表单**，则这些设置将对其产生影响；请参见第 109 页的 “单元格”。

在 App 之间进行复制

您可以在同时运行的多个 COMSOL Multiphysics 会话之间复制粘贴表单和表单对象。您也可以在一个会话中从当前 App 复制粘贴到新加载的 App。

在栅格布局模式下，可以在会话之间复制一个单元格、多个单元格、整行及整列。

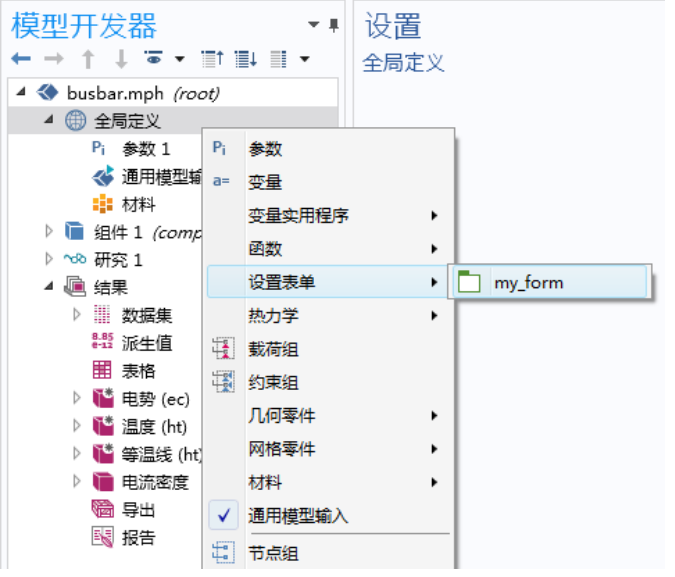
当您在 App 之间复制粘贴表单和表单对象时，复制的对象可能包含对其他表单和表单对象的引用。这类引用在复制到 App 后未必有意义。有关对象粘贴适用规则的更多信息，请参见第 289 页的 “附录 B - 在 App 之间进行复制”。

在 App 之间复制粘贴对象时，如果系统检测到潜在的兼容性问题，则会显示一个消息对话框。在这种情况下，您可以选择取消粘贴操作。

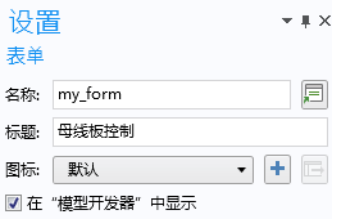
在 “模型开发器” 中使用表单

您可以在 “模型开发器” 中使用不含图形表单对象的表单。举例来说，您可以使用此功能为常见任务或重复性任务创建定制的**设置**窗口。

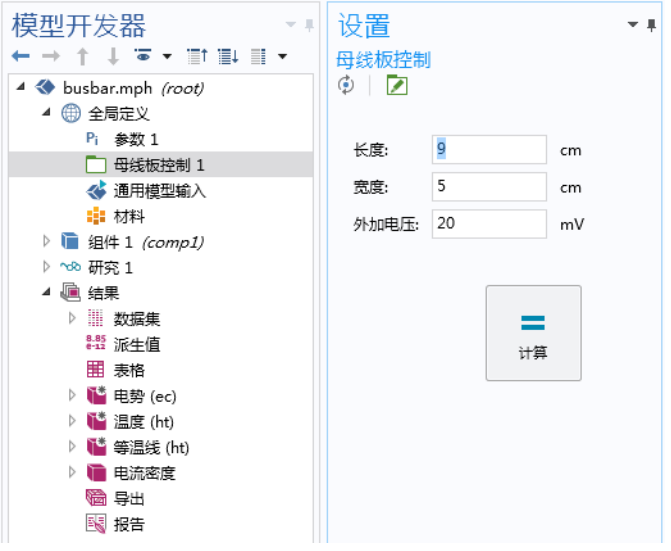
通过右键单击**全局定义**并从**设置表单**中选择表单，即可使用该表单。



您可以通过**在 “模型开发器”中显示**复选框控制表单是否在 “模型开发器”中显示为**设置表单**。此复选框位于 “App 开发器”中相应表单的**设置**窗口中。



表单被添加到模型树后，会显示为**设置**窗口，如下图所示。

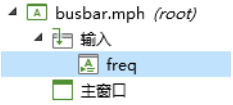


要显示**设置表单**，您可以：

- 单击相应的模型树节点。
- 从功能区的 “开发工具”选项卡的**设置表单**菜单按钮中将其选中。
- 通过从功能区的**开发工具**选项卡的**显示对话框**菜单按钮中选择它，将其显示为对话框。

输入

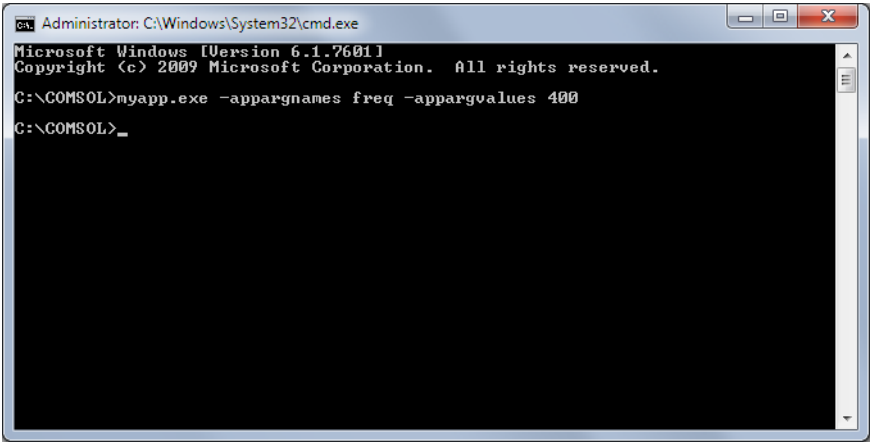
从操作系统命令行启动 App 时，可以提供输入变元。您可以在 App 树的**输入**节点下指定此类输入变元。



在 **App 变元**的相应**设置**窗口中，命令行变元会自动写入您定义为**选定源**的声明中。这些变元可用于提供输入数据或配置设置等。

在使用 COMSOL Multiphysics、 COMSOL Server 启动 App 时，以及启动通过 COMSOL Compiler 编译的 App 时，都可以使用命令行变元。下面是一个

Windows® 中编译的 App 示例，其中给出一个输入变元 `freq`，该变元取（双精度）值 `400`。



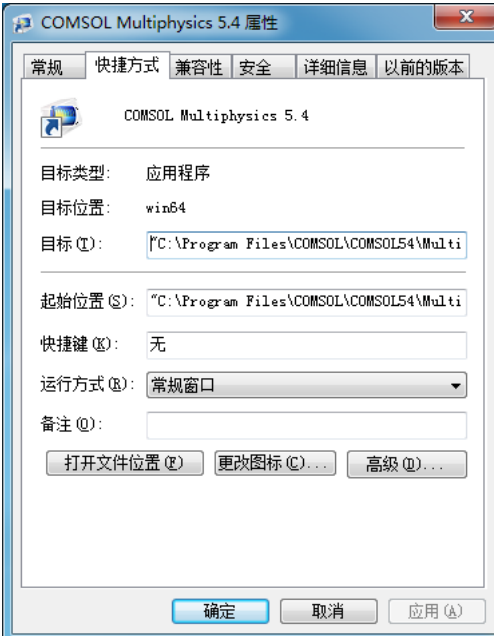
对于 COMSOL Multiphysics，相应的命令为

comsol.exe -run myapp.mph -appargnames freq -appargvalues 400

运行该命令时，您需要定位到可执行文件 `comsol.exe` 所在的 COMSOL Multiphysics 安装目录，例如

```
C:\Program Files\COMSOL\COMSOL54\Multiphysics\bin\win64
```

此外，您也可以复制并粘贴 **COMSOL Multiphysics 5.4** Windows® 桌面快捷方式图标 （为了保留原始快捷方式），右键单击该图标，然后选择**属性**；如下图所示。



例如，您可以将**目标**文本框修改为：

```
"C:\Program Files\COMSOL\COMSOL54\Multiphysics\bin\win64\comsol.exe" -run
myapp.mph -appargnames freq -appargvalues 400
```

要提供包含特殊字符的输入变元，您需要使用单引号。以下编译的 App 示例演示如何提供文件路径 （例如配置文件的路径）作为输入变元：

```
myapp.exe -appargnames configfile -appargvalues 'C:\\COMSOL\\my_conf.dat'
```

对于 COMSOL Server，您可以直接在浏览器地址栏 (URL) 中提供变元；例如：

```
http://<host:port>/app/myapp_mph?appargnames=freq&appargvalues=400
```

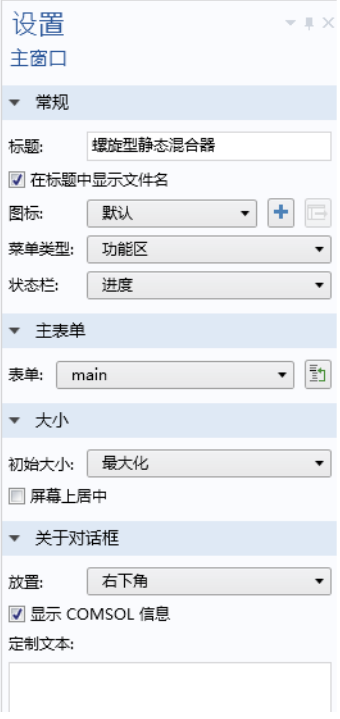
主窗口

在 App 树中，**主窗口**节点表示 App 的主窗口，也是用户界面的顶级节点。它包含窗口布局、主菜单明细以及可选的功能区明细。

常规

设置窗口包含**常规**栏，用于设置：

- **标题**
- **在标题中显示文件名**
- **图标**
- **菜单类型**
- **状态栏**



标题是 App 主窗口顶部的文本，其最左侧显示相关**图标**。默认情况下，**标题**与创建该 App 所用的模型的标题相同。如果您希望在**标题**左侧显示 App 的文件名，则保持选中**在标题中显示文件名**复选框。

在**图标**列表中，从库中选择一个图像，或从本地文件系统将图像 (`*.png`) 添加到库中，然后将其用作图标。如果您添加一个新图像，它将放入图像库，并由此嵌入 App。您还可以单击**将图像添加到库并在此使用**按钮右侧的**导出**按钮来导出图标。

App 树的**主窗口**节点包含一个子节点，名为**菜单栏**。您可以使用**菜单**类型设置，将该子节点从**菜单栏**改为**功能区**。

状态栏列表可以控制状态栏中显示的内容。选择**进度**可以显示进度条 （默认设置），也可以选择**无**。请注意，您也可以使用方法来创建定制的进度条。

主表单

主表单栏包含对主窗口显示的表单的引用。在使用表单集合时此设置非常重要，因为它可以确定在首次打开 App 时，将哪个表单显示为主窗口。

大小

在**大小**栏中，**初始大小**设置可以确定在首次启动 App 时主窗口的大小。其中包含三个选项：

- **最大化**，可以在运行 App 时将窗口最大化。
- **使用主表单大小**，可以使用主表单的大小；请参见第 47 页的 “单独的表单设置窗口”。主表单通过**主表单**栏进行定义。此选项可进一步添加主窗口本身所需的大小，包括：窗口框架和标题栏、主菜单、主工具栏以及功能区。窗口大小可以根据菜单类型为**菜单栏**还是**功能区**，来自动进行计算。
- **手动**，供您输入宽度和高度的像素大小。在这种情况下，宽度和高度框中不会添加任何值。在使用此选项时，您需要确保有足够的空间来显示窗口标题、功能区和菜单栏。

此外，您还可以使用**屏幕上居中**复选框，它适用于与最大化窗口不对应的任意**大小**设置。

有关**使用主表单大小**选项的更多信息，请参见第 105 页的 “表单设置窗口和栅格”。

关于对话框

关于对话框栏用于设置**关于此 App** 对话框 （包含法律信息）的定制部分。**放置**选项可供您在**自动**、**文件菜单**、**功能区**或**右下角**之间进行选择。使用**右下角**选项，可以在 App 用户界面的右下角放置一个指向**关于此 App** 对话框的超链接。如果选中**显示 COMSOL 信息**复选框，则会显示 COMSOL 软件版本和产品信息。在**定制文本**框中输入的任何文本都会显示在该对话框的法律文本上方。在**定制文本**框中，包含 `http` 或 `www` 的文字会在可能的情况下被系统识别为超链接。例如， `http://cn.comsol.com` 或 `cn.comsol.com` 将替换为超链接。

菜单栏和工具栏

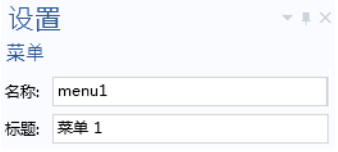
菜单栏节点可以包含**菜单**子节点，用来表示主窗口顶部的菜单。

对于**菜单栏**选项，您还可以添加**工具栏**。**工具栏**节点和**菜单**节点的子节点类型相同。

菜单、项和分隔线

菜单和**工具栏**节点的子节点类型包括**菜单**、**项**、**切换项**或**分隔线**，如下图所示：

“菜单”节点包含**名称**和**标题**设置。

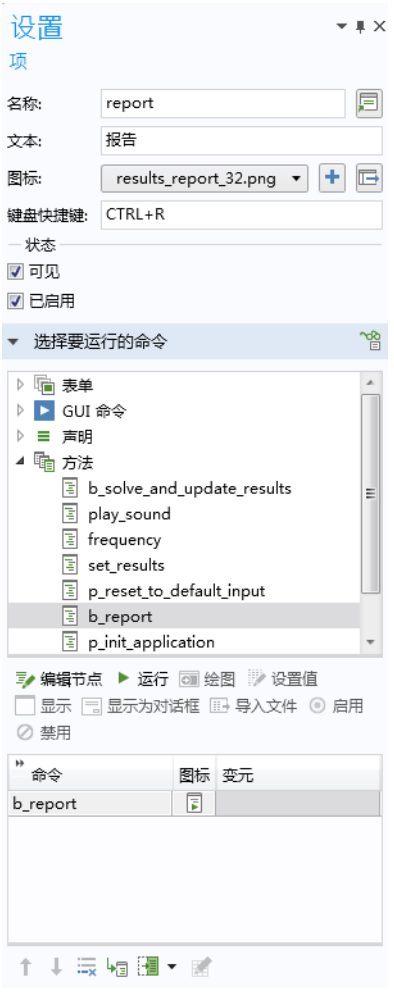


菜单节点可以包含子**菜单**节点，用来表示子菜单。

分隔线在菜单组和项组之间显示一条水平线，无设置项。

项节点的**设置**窗口与按钮的类似，也包含一个命令序列。和按钮一样，项也可以具有关联文本、图标和键盘快捷键。有关更多信息，请参见第 57 页的 “按钮”。同样，**切换项**节点的**设置**窗口与切换按钮的相似。

下图显示与报告创建方法关联的**项**的**设置**窗口。



下图显示一个包含**文件**菜单的 App 示例。

在 COMSOL Desktop 环境下运行 App 时，将始终显示**关闭 App** 菜单项。

另存为项的**设置**窗口如下图所示。

您可以通过方法来启用和禁用功能区、菜单及主工具栏项。有关更多信息，请参见第 313 页的 “附录 E - 内置方法库”。

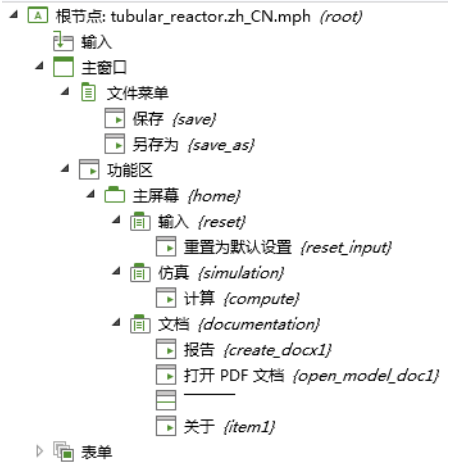
功能区

您可以选择将**功能区**添加到**主窗口**，而非**菜单栏**。**功能区**节点包含功能区明细，其中工具栏放置在一个或多个选项卡中。对于**功能区**选项，可以直接在**主窗口**节点下添加**文件**菜单。

功能区选项卡和功能区栏

功能区节点的子节点类型为**功能区选项卡**。**功能区选项卡**的子节点类型为**功能区栏**。**功能区栏**的子节点类型可以为**项**、**切换项**、**菜单**或**分隔线**。

项和**菜单**的功能与上述**菜单栏**和**工具栏**的功能相同。作为**功能区栏**子节点添加的**分隔线**是一条垂直线，用于在运行的 App 中分隔**项**组和**菜单**组。在 App 树中，**分隔线**则显示为一条水平线。下图显示了一个示例。



事件

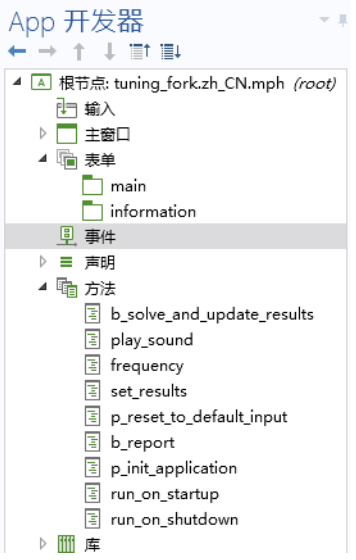
事件是指任意一项活动 （例如，单击按钮、键入键盘快捷键、加载表单或更改变量的值），它表明 App 需要执行一个或多个操作。每个操作都可以是前面描述的一个命令序列，也可以是方法的执行。方法本身可以是与特定表单或表单对象关联的局部方法，或是可以在 App 的任意位置发起的全局方法。全局方法在 App 树的**方法**节点中列出。表单方法在相应表单的节点下列出。局部方法在表单或其关联的表单对象的**设置**窗口中定义。当表单对象具有关联方法时，可通过在对象上执行 Ctrl+Alt+ 单击操作，将其打开进行编辑。如果对不含方法的表单对象执行 Ctrl+Alt+ 单击操作，则会创建与该对象关联的新局部方法，并将其打开进行编辑。

发起这些操作的事件也可以是全局或局部的。全局事件在 App 树的**事件**节点中列出，包含可通过各种数据实体的更改来触发的所有事件，如全局参数或字符串变量。全局事件还可以同 App 的启动和关闭操作关联起来。局部事件 （如局部对象）在表单或其关联的表单对象的**设置**窗口中定义。

只要源数据发生更改，便会触发**事件**节点，不论这种更改是由方法、表单对象，还是任何其他方式引起的。不过，仅当用户更改表单对象的值时才触发与该表单对象关联的事件。

启动和停止运行时的事件

全局或局部方法可以同 App 启动 （**启动时**）和停止运行 （**要停止运行时**）的事件相关联。要访问这些事件，可以单击 App 树中的**事件**节点。



以下情况会触发停止运行事件：

- App 的用户通过单击 App 窗口右上角的**关闭 App** 图标来关闭 App 窗口
- **退出 App** 命令由表单对象发出
- 使用 `exit()` 命令运行方法

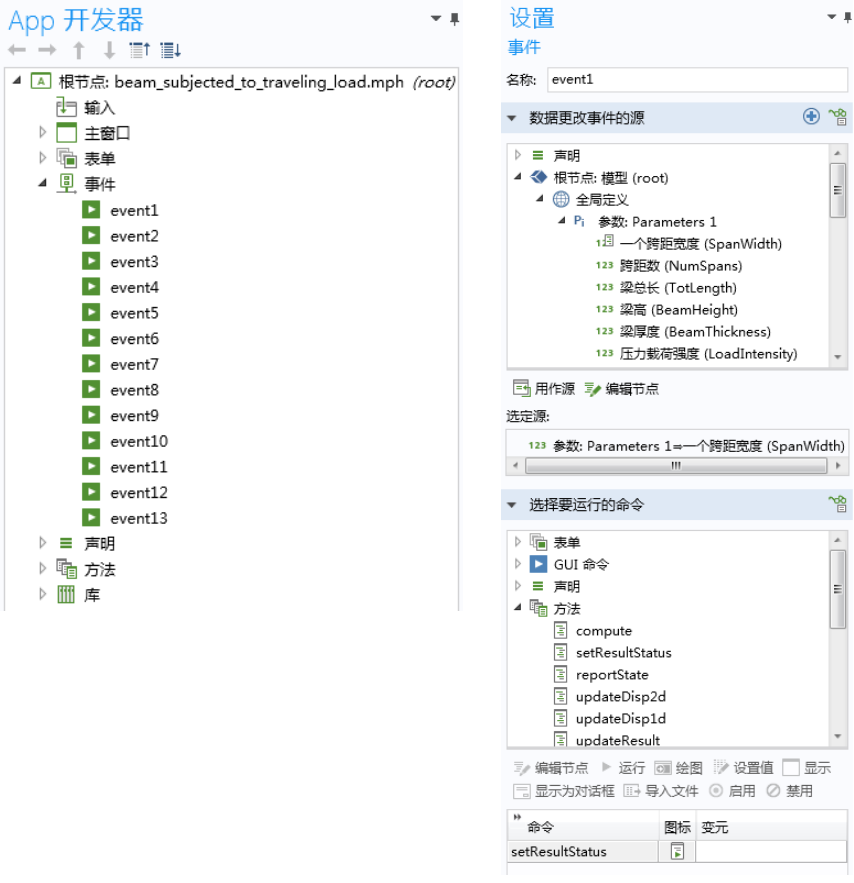
例如，在触发停止运行事件时运行的方法可以自动保存关键数据，或提示用户保存数据。此外，在触发停止运行事件时运行的方法还可以返回布尔值 `true`，从而取消停止运行。

启动时事件的限制

用于**启动时**事件的方法无法使用与图形或用户界面相关的 “App 开发器”功能。这是因为**启动时**事件在加载完整的 App 用户界面之前运行。例如，用于初始化图形的方法 （如**缩放到窗口大小**）需要作为表单的**加载时**事件运行，而非作为全局的**启动时**事件运行。另一个示例是使用内置方法 （如 `confirm`）显示对话框。在这种情况下，不会显示任何对话框，操作将被忽略。

全局事件

右键单击**事件**节点并选择**事件**，以将事件添加到 App。事件会侦听正在运行的 App 中发生的更改。一旦发生更改，它便会运行一个命令序列。在下图中，当字符串变量 `SpanWidth` 的值发生更改时，会运行 `setResultsStatus` 方法。



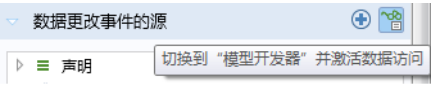
请注意，由于这种事件类型具有全局范围，并且不与特定的表单关联，因此在引用图形对象时，必须使用完整路径 `/form1/graphics1`。

下面两节描述事件**设置**窗口中提供的选项。

数据更改事件的源

本节描述 “App 开发器”窗口中树结构的部分视图。节点表示某种数据或具有类似子节点。

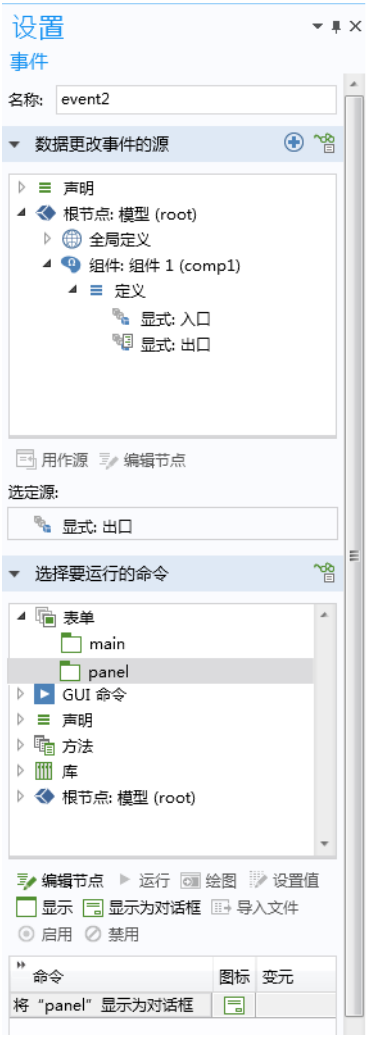
您可以单击**数据更改事件的源**栏标题中的**切换到 “模型开发器”并激活数据访问**按钮，来展开可用的数据节点列表。



有关更多信息，请参见第 169 页的 “方法编辑器中的数据访问”。

请注意，**显式**选择也可以作为**数据更改事件的源**。从而支持在用户单击几何对象、域、面、边或点时运行命令序列或方法。下图显示一个全局事件对话框，

该事件在用户更改名为**出口**的**显式**选择的内容时，打开一个表单面板作为对话框。



选择要运行的命令

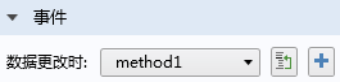
在**事件**的**设置**窗口中，**选择要运行的命令**栏与按钮的类似，可供您定义命令序列。有关更多信息，请参见第 57 页的 “按钮”。

表单和表单对象事件

表单和表单对象事件与全局事件类似，但针对表单或单个表单对象进行定义。这些事件没有关联的命令列表，而是直接引用一个全局、表单或局部方法。

由数据更改触发的事件

对于某些类型的表单对象，您可以指定当数据更改时要运行的方法。此设置位于表单对象**设置**窗口的**事件**栏，如下图所示。



数据更改时下拉列表包含**无** （默认设置）、 App 树的**方法**节点或相应表单的**方法**节点下的任何可用方法，以及局部方法 （可选）。

支持此类事件的表单对象包括：

- **输入框**
- **复选框**
- **组合框**
- **图形**
- **文件导入**
- **数组输入**
- **单选按钮**
- **文本**
- **列表框**
- **表格**
- **滑块**

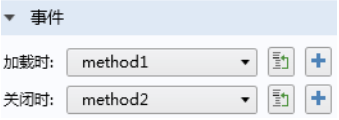
按钮具有通过单击触发的关联事件。菜单、功能区和工具栏项具有通过选择它们来触发的关联事件。相应的操作是在按钮对象或项的**设置**窗口中定义的命令序列。有关命令序列的更多信息，请参见第 57 页的 “按钮”。

选择多个表单对象

您可以使用 Ctrl+ 单击操作并选择要运行的方法，同时为多个表单对象指定**数据更改时**事件。例如，通过这种方式，您可以快速指定由任意选定表单对象发起的数据更改事件应运行一个方法，以通知用户绘图和输出无效。此功能并非适用于所有表单对象组合。

通过加载或关闭表单触发的事件

在加载 （**加载时**）或关闭 （**关闭时**）表单时可以运行方法。



这种类型的事件在表单的**设置**窗口中提供，通常在表单显示为对话框时使用，或用于激活在表单集合中用作窗格的表单。请注意，用于初始化图形的方法（如**缩放到窗口大小**）需要作为表单的**加载时**事件运行，而非作为全局的**启动时**事件运行。

使用局部方法

事件可以调用未显示在 App 树中的局部方法。有关局部方法的更多信息，请参见第 184 页的 “局部方法”。

声明

App 树中的**声明**节点用于声明全局变量和对象，除模型中已定义的全局参数和变量外，还可以使用这些变量和对象。**声明**节点下定义的变量用于表单对象和方法。在表单对象中，这些变量可以存储要在其他表单对象或方法中使用的值。对于不在表单对象与方法之间传递，而只是用于方法内部的变量，无需在**声明**节点中进行声明。在方法中，**声明**节点下定义的变量具有全局范围，可以直接通过名称使用。有关如何访问模型树中定义的全局参数的信息，请参见第 193 页的 “访问全局参数”。

您可以为表单创建局部**声明**节点。表单的此类**声明**只能在该特定表单 （包括该表单的局部表单对象和方法）中使用。

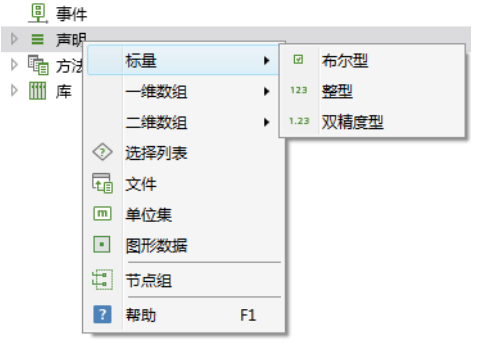
下面列出不同类型的全局**声明**：

- **标量**
- **一维数组**
- **二维数组**
- **选择列表**
- **文件**
- **单位集**
- **快捷方式**
- **图形数据**

声明表单仅支持以下类型：

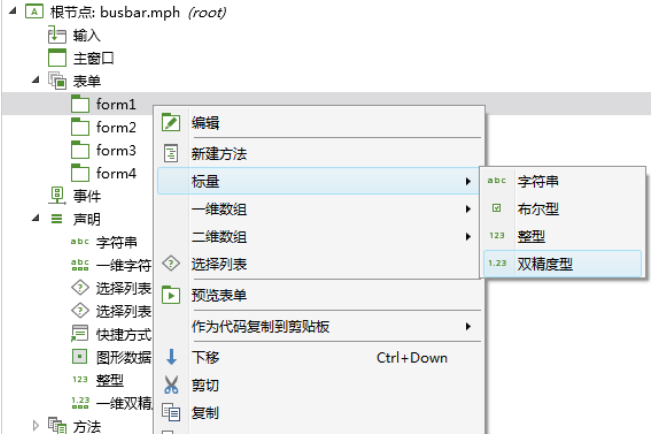
- **标量**
- **一维数组**
- **二维数组**
- **选择列表**

您可以右键单击**声明**节点或使用功能区来访问这些声明类型。

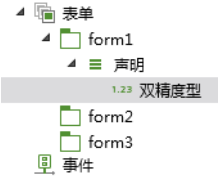


请注意，**快捷方式**不是通过此菜单创建的，您可以单击表单对象**设置**窗口中**名称**旁的**创建快捷方式**按钮，也可以对选定的表单对象使用 Ctrl+K 进行创建。

要为表单创建局部**声明**，可以右键单击相应的表单并选择变量类型，如下所示。



表单的局部变量位于**声明**节点（该表单的子节点）下，如下所示。

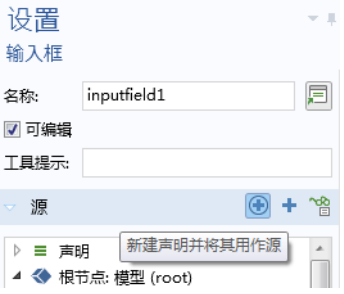


前三种类型的声明，即**标量**、**一维数组**和**二维数组**，可以为以下数据类型：

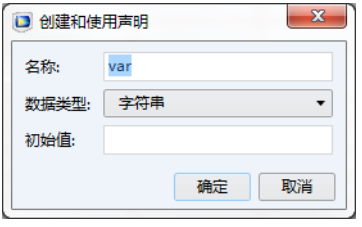
- **字符串**

- 布尔型
- 整型
- 双精度型

除了右键单击**声明**节点，您还可以在许多类型的表单对象的 “设置” 窗口的**源**栏中单击**新建声明并将其用作源**按钮。



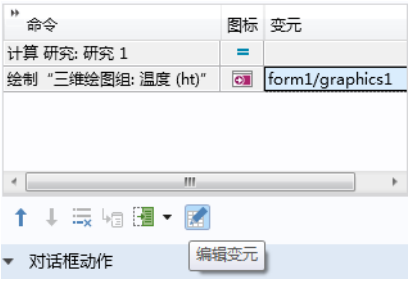
此时会打开一个对话框，可供您快速声明标量变量。

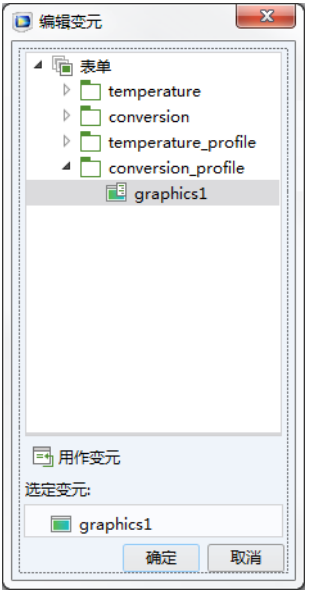


将声明用作命令的输入变元

例如，按钮的命令序列中使用的某些命令可以具有输入变元。有关更多信息，请参见第 57 页的 “按钮”。

下图显示的命令序列包含一个**绘制温度**命令，其输入变元为 form1/graphics。





您可以使用声明作为命令的输入变元。

如果要使用标量变量、一维数组或二维数组作为输入变元，则可以使用相应的变量名称。也可以通过索引来访问数组的单个元素，或二维数组的行或列。例如，您可以使用 my_variable(1) 来访问一维数组 my_variable 的第一个分量。二维数组元素可以作为标量通过两个索引进行检索，例如 my_matrix(2,3)。索引本身可以是其他声明的变量，如 my_variable(n)。

对于需要图形对象作为输入变元的命令，仅支持字符串类型的声明，并能根据需要加上适当的索引。如果有一个名为 graphics1 的图形对象，并有一个名为 graphics1 的字符串声明，则会使用该字符串声明的内容。使用单引号的情况属于例外，如 'graphics1'，此时会使用图形对象 graphics1。此规则同样适用于命令和输入变元的其他组合。

变量名称

变量的**名称**是不含空格的文本字符串。其中可以包含字母、数字和下划线。不支持保留名称 root 和 parent，并且不能使用 Java® 编程语言关键字。

标量

标量声明用于定义要用作字符串、布尔型、整型或双精度型数据的变量。

字符串

标量字符串变量类似于模型中的全局参数或变量，但有所区别。模型中的参数或变量有一定的限制，它们的值必须为有效的模型表达式，而标量字符串变量则没有这一限制。您可以通过方法中的转换函数，使用字符串变量来表示双精度型、整型或布尔型数据。有关更多信息，请参见第 325 页的 “转换方法”。您也可以在许多表单对象中使用字符串变量作为源，例如输入框、组合框、卡片堆和列表框。

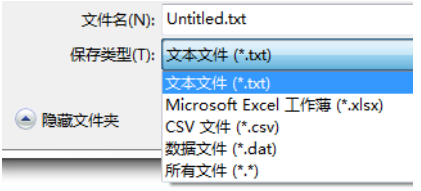
下图显示字符串变量 `graphics_pane`、 `email_to` 及 `solution_state` 的**设置**窗口。

字符串声明及其他声明可通过**变量列表**表格下方的**从文件加载**和**保存到文件**按钮从文件进行加载或保存到文件中。

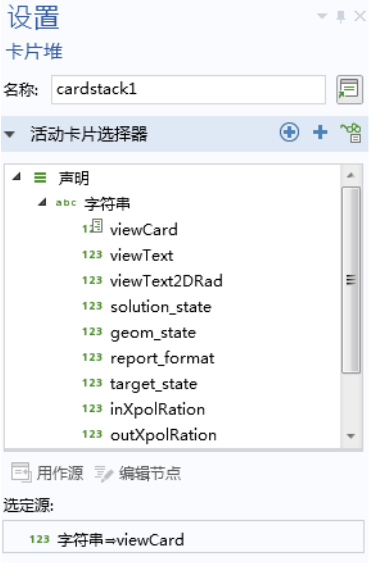
从文件加载和**保存到文件**按钮所执行的加载和保存操作适用于以下文件格式：

- 文本文件 (`.txt`)
- Microsoft® Excel® 工作簿 (`.xlsx`)
  - 需要 LiveLink™ *for* Excel®
- CSV 文件 (`.csv`)
- 数据文件 (`.dat`)

下图显示了可供选择这些文件格式的下拉列表。



下图显示卡片堆对象的**设置**窗口，说明如何使用声明的字符串，其中使用字符串变量 `viewCard` 作为源 （**活动卡片选择器**）。



有关使用卡片堆的更多信息，请参见第 251 页的 “卡片堆”。

布尔型

您可以使用布尔变量作为复选框、其他表单对象及方法的源。布尔变量可具有两种状态：`true` 或 `false`。默认值为 `false`。下图显示两个布尔变量的声明。

示例代码

在以下示例代码中，布尔变量 `bvar` 的值通过复选框进行控制。如果 `bvar` 为 `true`，则在 `graphics1` 中绘制**绘图组 4 (pg4)**。否则，绘制**绘图组 1 (pg1)**。

```
if (bvar) {
  useGraphics(model.result("pg4"),"graphics1");
} else {
  useGraphics(model.result("pg1"),"graphics1");
}
```

整型和双精度型

整型和双精度型变量与字符串相似，只是它们的值需要分别为整型或双精度型。

一维数组

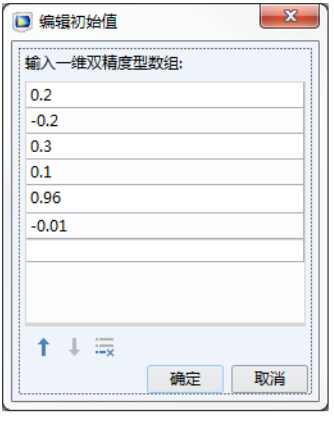
一维数组节点可以声明一个或多个命名的字符串、布尔型、整型或双精度数组，您可以从表单对象和方法进行访问。一维数组中的元素数量不受任何限制，例如，您可以使用一维数组在行数可变的表格中存储列。**设置**窗口包含一

个表格，您可以在每行指定一个变量数组。在下图中，声明了两个双精度数组：xcoords 和 ycoords。

当表格表单对象中添加行时，**新元素值**列中的值会赋给数组的新元素。字符串、布尔型及整型数组的功能与双精度数组类似。

初始值

初始值可以是任意长度的一维数组。要编辑初始值，可单击**变量列表**下方的**编辑初始值**按钮。此时会打开一个对话框，供您输入每个分量的值。下图显示了一维双精度数组的示例。



数组语法

数组定义必须以大括号开始并以大括号结束 （{ 和 }），并且每个元素都必须用逗号分隔。如果需要在数组元素中使用特殊字符 （如空格和逗号），则需要为该元素添加单引号 (‘)。下表显示了几个一维数组示例：

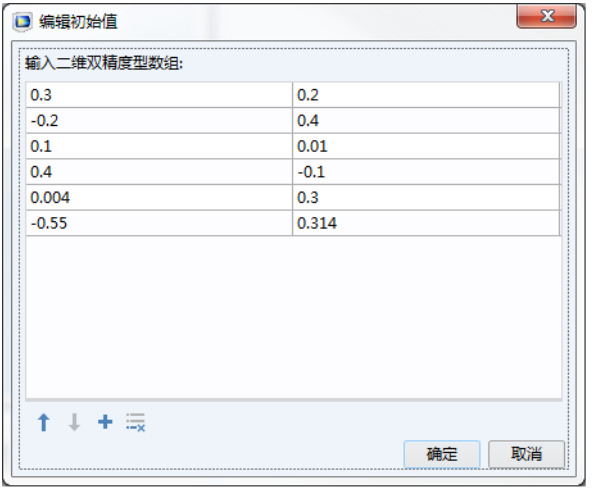
数组语法	得到的数组
`{1, 2, 3}`	包含元素 1、 2 和 3 的 3- 元素数组
`{}`	空数组
`{‘one, two’, ‘three by four’}`	元素中包含特殊字符的 2- 元素数组
`{{1, 2, 3},{‘one, two’, ‘three by four’}}`	包含一个 3- 元素数组和一个 2- 元素数组的 2- 元素数组

二维数组

二维数组节点可以声明一个或多个二维数组，您可以使用表单对象和方法进行访问。在下图中，声明了二维双精度数组 `xycoords`。

初始值

默认（或初始）值可以是任意大小的二维数组。要编辑初始值，可单击**变量列表**下方的**编辑初始值**按钮。此时会打开一个对话框，供您输入每个分量的值。下图显示了二维双精度数组的示例。



数组语法

下表显示了几个二维数组示例：

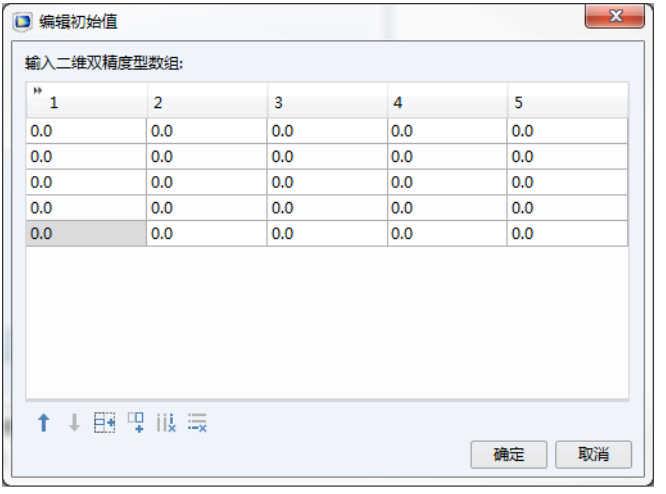
数组语法	得到的数组
{{}}	空三维数组
{{‘5‘,‘6‘},{‘7‘,‘8‘}}	2x2 的字符串矩阵
{{1, 2, 3}, {4, 5, 6}}	2x3 的双精度矩阵

对于二维数组，行对应于第一个索引，以使 {{1,2,3},{4,5,6}} 等同于以下矩阵：

```
1 2 3
4 5 6
```

假设上述 2x3 矩阵存储在二维数组变量 arr 中，则元素 arr[1][0] 等于 4。

要交互式定义二维数组的**初始值**，为**列数**选择**未定义**选项。单击**编辑初始值**按钮可以打开一个对话框，您可以在其中交互更改行数和列数，如下图所示。

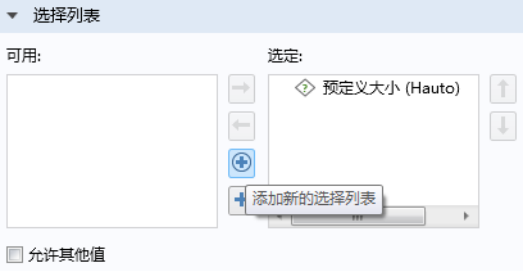


选择列表

选择列表节点包含的列表可用于组合框、单选按钮或列表框。选择列表的**设置**窗口包含**标签**、**名称**和一个表格，其中分为**值**和**显示名称**两列。您可以在第一列输入属性值 （**值**），并在第二列 （**显示名称**）输入对用户显示的相应文本

（例如，在组合框列表中）。**值**始终被解释为字符串。在下例中， mat1 从组合框返回时，会变成字符串 “mat1”。

除了通过右键单击**声明**节点来创建选择列表，您还可以在使用此类列表的表单对象的 “设置” 窗口中单击**添加新的选择列表**按钮，如下图所示。



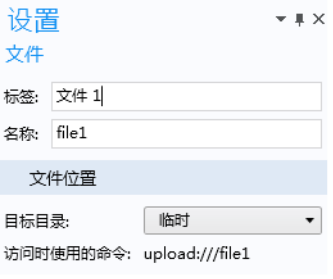
此外，您还可以单击**添加新的表单选择列表**来创建表单的局部选择列表。

激活条件

您可以右键单击**选择列表**节点，来添加**激活条件**子节点。并使用激活条件在两个或多个选择列表之间进行切换，具体取决于变量的值。如需有关使用带激活条件的选择列表的示例，请参见第 223 页的 “使用组合框更改材料”。

文件

当您使用内置方法 `importFile` 时，文件声明主要用于在方法代码中导入文件。有关使用 `importFile` 方法及其他方法来处理文件的更多信息，请参见第 314 页的 “文件方法”。不过，**文件**声明节点下的条目也可用于**文件导入**对象。下图显示文件声明的**设置**窗口。



用户选定的文件可通过 `upload:///file1`, `upload:///file2` 等语法在表单对象或方法中被引用。文件名句柄 （`file1`、 `file2` 等）稍后可用于引用由用户在运行时选取的实际文件名。

有关文件声明和文件处理的更多信息，请参见第 290 页的 “附录 C - 文件处理和文件方案语法”。

单位集

单位集节点包含的列表可在组合框、单选按钮或列表框中用于更改单位。单位集的**设置**窗口包含两栏：**单位组**和**单位列表**。

单位组表格中的每一行都是一个单位组，表示在 App 用户界面的上下文中具有特殊含义的单位集合。每一列都表示带有**值**和**显示名称**标签的一组单位。

单位列表表格中的每一行都是一个单位列表，其对应的列包含相同尺寸的单位；例如 mm、cm、dm、m 和 km。**单位列表**表格的标题为**名称**以及在**单位组**栏中定义的**显示名称**。单位列表用于指定在运行 App 时，引用**单位集**的表单对象可切换显示的单位。

上图演示了允许切换公制和英制单位的 App 的**单位集**的使用。在本例中，定义了两个单位组：`SI` 和英制。**单位集**的**标签**已更改为单位制。

值列包含字符串值，表示当前选择的单位组。这些字符串值可以通过方法进行操作。**显示名称**列是用户界面中显示的字符串。**初始值**列表包含默认单位组（上例中的 `SI`）。

在上例中，**单位列表**表格包含三列：**名称**、 **SI** 和**英制**。 **SI** 与**英制**列是根据**单位组**栏中的组动态创建的。表格中的每一行分别对应于一个物理量，如本例中的长度与电势。表格中的每一列分别对应于允许的长度单位和电势单位。

下图显示一个示例 App，其中的组合框表单对象用于在 **SI** 和**英制**单位组之间进行选择。

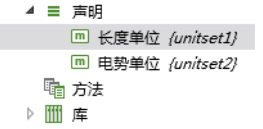
下图显示组合框的**设置**窗口，其中使用上例的**单位集**作为**源**。

通过这种方式，可以用**单位集**代替**选择列表**，来创建用于单位选择的组合框。同样，您也可以使用列表框或单选按钮对象来代替组合框。

以下两个图分别显示**长度**和**外加电压**这两个输入框对应的**设置**窗口。

单位量纲检查设为**附加来自单位集的单位**。**单位集**设为**单位制 {unitset1}**（本例中使用的用户定义的**单位集**声明标签）。**单位列表**分别设为**长度**和**电势**。在使用**附加来自单位集的单位**时，**数值验证**栏（位于**数据验证**下）引用**单位集**的**初始值**；在本例中分别为 **cm** 和 **mV**。**最小值**和**最大值**在 App 运行时自动缩放，并且单位由 App 用户进行更改。有关输入框对象设置的更多信息，请参见第 85 页的“输入框”。

以下图形分别说明如何使用两个**单位集**声明来设置长度和电势单位。



以下图形显示相应的**单位集**声明**设置**窗口。

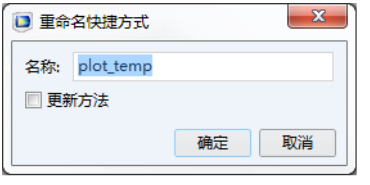
请注意，在本例中，通过使用三个**单位集**声明，您可以为**长度**和**宽度**输入框单独设置长度单位。下图显示了这样一个示例，其中使用三个组合框来代替单位标签，每个组合框都使用单独的**单位集**声明作为源。

当您需要更大的灵活性时，可以结合使用**选择列表**和**单位集**。例如，对于组合框，您可以使用**单位集**作为**选定源** （字符串），并选择非**单位集**的**选择列表**。

快捷方式

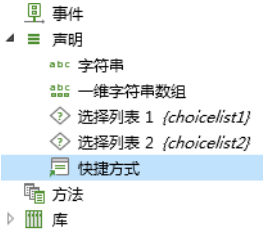
您可以使用特定的语法在方法中引用表单对象及其他用户界面注释。例如，使用默认的命名方案 `form3/button5` 可以引用 `form3` 中名为 `button5` 的按钮；而 `form2/graphics3` 则可以引用 `form2` 中名为 `graphics3` 的图形对象。您还可以更改表单和表单对象的默认名称。例如，如果 `form1` 是主表单，则可以将其名称改为主表单。

要按名称简化引用表单对象以及菜单、功能区和工具栏项，可以使用定制名称创建快捷方式。在对象或项的**设置**窗口中，单击**名称**框右侧的按钮并键入您选择的名称。

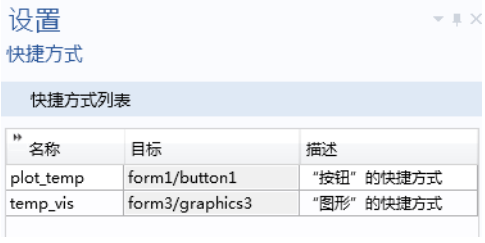


要创建或编辑快捷方式，还可以使用键盘快捷键 Ctrl+K。

您创建的所有快捷方式均显示在 App 树的**声明**下的**快捷方式**节点中。



在下方的**快捷方式设置**窗口中，创建了两个快捷方式 `plot_temp` 和 `temp_vis`，分别用于按钮和图形对象。

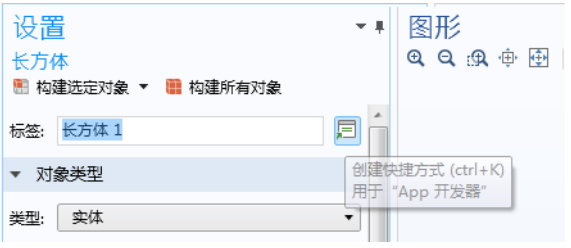


快捷方式可在其他表单对象或 “方法编辑器” 的代码中引用。下例显示的快捷方式 `temp_vis` 用作温度图的输入变元。

当对象被重命名、移动、复制以及生成副本时，快捷方式会自动更新。它们在方法中作为只读的 Java® 变量，就像字符串、整型、双精度型和布尔型声明。

建议您使用快捷方式，这样就能在 App 用户界面结构发生更改时，不必再调整方法编辑器代码。

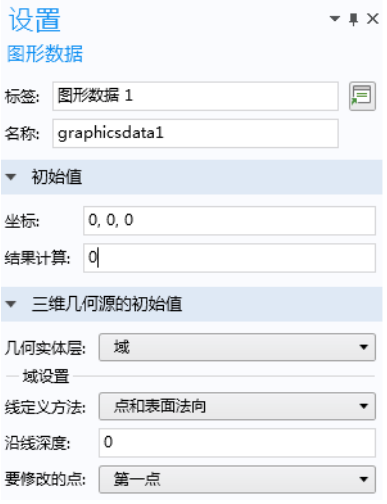
快捷方式也在 “模型开发器”中提供，并与 “App 开发器”结合使用。在模型树节点的**设置**窗口中，单击**标签**框右侧的按钮并键入您选择的名称。



快捷方式的定制名称可在方法中用作全局变量，例如，还能在录制代码或新方法时使用。

图形数据

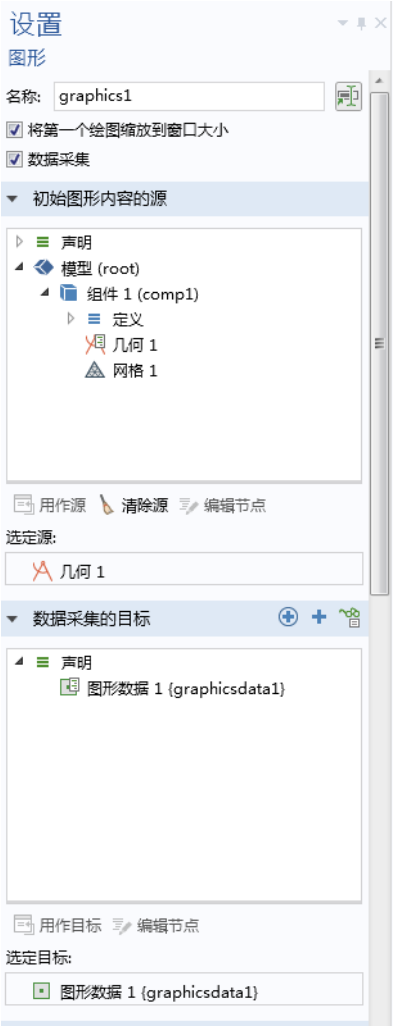
图形数据声明节点用于根据用户的单击操作从图形对象的特定坐标选取数据。下图显示相应的**设置**窗口。



初始值栏包含提取的数据属性**坐标**和**结果计算**的默认值。**三维几何源的初始值**栏包含选择方法的设置，可在图形对象的**初始图形内容的源**设为几何节点时使用。

图形数据声明的各种属性显示在“编辑器工具”窗口中，如下图所示。

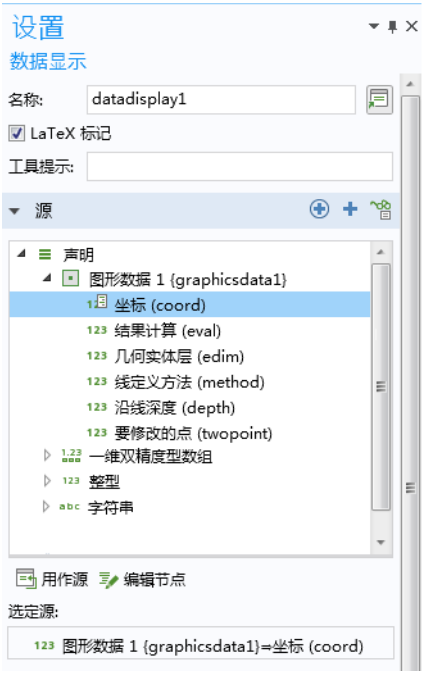
要使用**图形数据**声明节点进行数据采集，可在图形对象的**设置**窗口中选中**数据采集**复选框，并选择**图形数据**节点作为**数据采集的目标**，如下图所示。



结果中的图形数据

当图形对象的**初始图形内容的源**设为绘图组节点时，**结果计算**值对应于鼠标指针所在位置的字段值。**坐标**值对应于该位置的坐标。请注意，在“模型开发器”中，则对应于**计算二维**或**计算三维**表格中显示的数据。

下图显示将**坐标**属性用作**源**的数据显示对象。



也可以使用**坐标**属性作为数组输入对象的**源**。**结果计算**属性可以用作包括数据显示和输入框对象在内的多个表单对象的**源**。

几何中的图形数据

几何实体层、**线定义方法**、**沿线深度**和**要修改的点**设置仅当图形对象的**初始图形内容的源**设为三维几何节点时适用。当**几何实体层**设为**域**时，这些设置提供的点选择方法与**域点探针**相同；当**几何实体层**设为**边界**时，则与**边界点探针**相同。**线定义方法**、**沿线深度**和**要修改的点**设置仅当**几何实体层**设为**域**时适用。

方法编辑器

您可以使用 “方法编辑器” 来编写代码，以执行 “模型开发器” 中模型树节点的标准运行命令以外的操作。例如，方法可以执行循环、处理输入和输出以及向 App 用户发送消息和通知等。

Java® 编程语言用于编写 COMSOL 方法，这意味着您可以使用所有 Java® 语法和 Java® 库。除了 Java® 库，“App 开发器” 还有自己的内置库，用于构建 App 和修改模型对象。模型对象是一种数据结构，用于存储 App 中嵌入的底层 COMSOL Multiphysics 模型的状态。有关这些内置方法的更多信息，请参见第 313 页的 “附录 E - 内置方法库”和 Application Programming Guide。

“App 开发器” 中 App 树的内容通过 App 对象进行访问，这是模型对象的一个重要部分。您可以使用 “方法编辑器” 来录制和编写代码，“方法编辑器” 可以直接访问并更改正在运行的 App 的用户界面外观，例如按钮文本、图标、颜色及字体。

方法分为全局方法、表单方法和局部方法。全局方法显示在 App 树中，可通过所有方法和表单对象进行访问。表单方法在 App 树中显示为其所属表单的子节点。局部方法与某个表单对象或事件相关联，并可从相应的**设置**窗口中打开。有关局部方法的更多信息，请参见第 184 页的 “局部方法”。

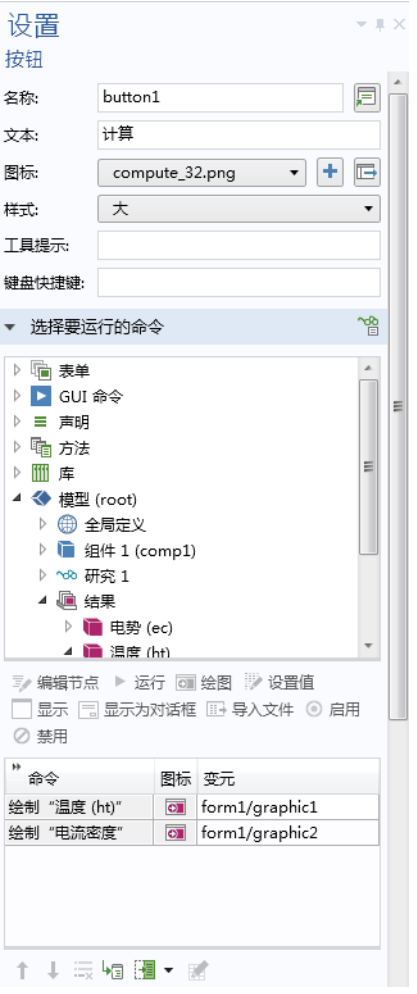
“App 开发器” 提供丰富的工具和资源，可帮助您为方法创建代码。后续章节将进一步介绍相关详细信息，借助这些工具和资源，您可以通过复制 - 粘贴代码或自动生成代码块等操作来提高工作效率。

将命令序列转换成方法

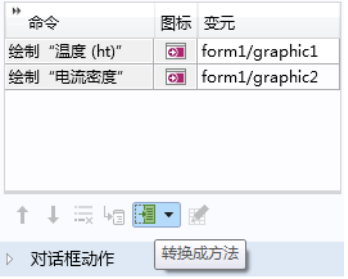
在 “表单编辑器” 中，从**设置**窗口中现有命令序列下方显示的菜单按钮中选择**转换成方法**。该命令序列将自动替换为等效的方法。您可以使用同样的方式选择**转换成表单方法**和**转换成局部方法**。

假设您已创建一个计算按钮，并希望在计算完成时听到声音提示。现在，我们来看看如何使用 “方法编辑器” 来实现这一功能。

稍后，本节还会介绍如何在不使用“方法编辑器”的情况下实现相同功能。下图显示**计算**按钮的**设置**窗口。

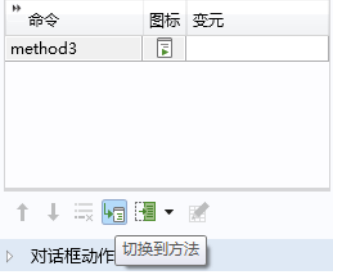


单击命令序列下的**转换成方法**按钮。

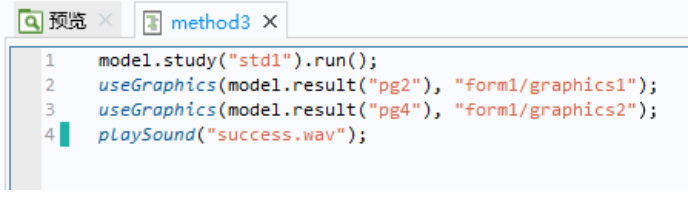


本例中的命令序列则会替换为方法 `method3`。

单击**切换到方法**按钮。此时会打开 “方法编辑器”，并且 `method3` 选项卡处于活动状态。

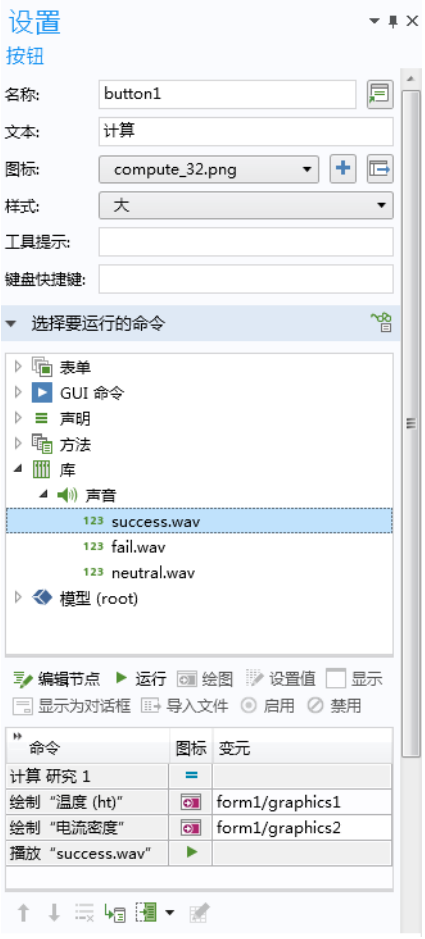


在 “方法编辑器” 中，使用下图所示的语法添加对内置方法 `playSound` 的调用，播放声音文件 `success.wav`，该文件可在 COMSOL 声音库中找到。



对于新添加的行，其左侧会显示一个绿色条。

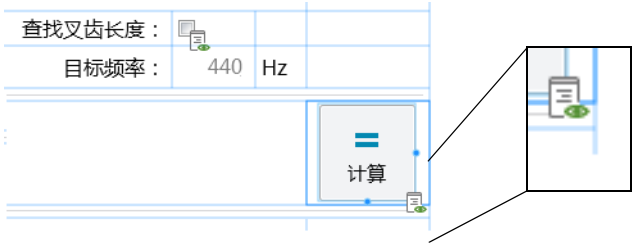
请注意，在上例中，您无需使用 “方法编辑器”。在命令序列的**库 > 声音**节点下选择文件 success.wav，然后单击树结构下方的**运行**命令按钮，如下图所示。



不过，有许多内置方法没有对应的命令序列节点。有关更多信息，请参见第 313 页的 “附录 E - 内置方法库”。

带关联方法的表单对象

带有关联方法的表单对象在 “表单编辑器” 中通过特殊图标来表示，如下图所示。在本例中，名为 **Find prong length** 的复选框和**计算**按钮都具有关联方法。

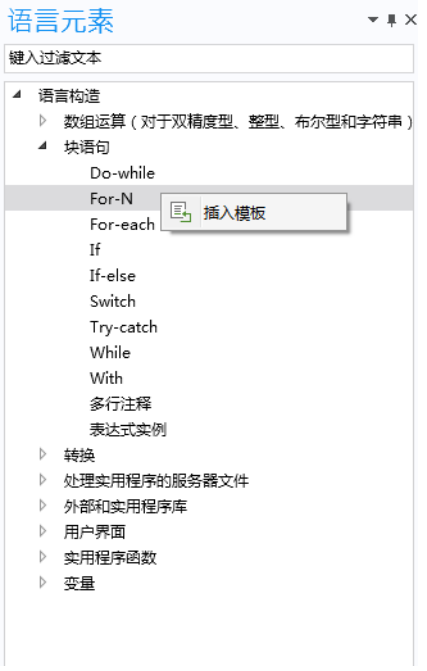


对表单对象执行 Ctrl+Alt+ 单击操作可以在 “方法编辑器” 中打开方法。如果没有与表单对象关联的方法，则会创建一个新的局部方法与该表单对象关联，并在 “方法编辑器” 中打开。

如果关联方法存在编译错误，则会显示其他图标，如下图所示。

语言元素窗口

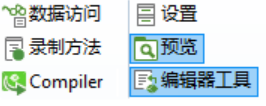
“方法编辑器”中的**语言元素**窗口显示一个语言构造列表。双击或右键单击列表中的某一项，即可将模板代码插入选定的方法中。



另请参见第 190 页的 “语言元素示例”。

方法编辑器中的编辑器工具

要显示**编辑器工具**窗口，可以在**方法**选项卡的**主**组中单击相应的按钮。



在 “方法编辑器”中使用**编辑器工具**窗口时，您可以右键单击编辑器树中的节点，生成与之关联的代码。根据节点的不同，最多可以使用八种不同的选项：

- **获取**
- **设置**

- **全部设置**
- **创建**
- **运行**
- **启用**
- **禁用**
- **编辑节点**

在选择前七个选项时，会将对应的代码添加到当前选定的方法中。选择**编辑节点**选项时，系统会跳转到模型树节点的**设置**窗口。

下图显示的节点示例包含六个选项。

在选定某个节点时，编辑器树下方的工具栏中会显示可用于生成代码的选项。当您使用 “表单编辑器” 时，**编辑器工具**窗口也是一个重要的工具。有关更多信息，请参见第 55 页的 “表单编辑器中的编辑器工具”。

键盘快捷键

假设某个方法中有一行代码通过以下方式引用模型对象：

```
model.result("pg3").feature("surf1").create("hght1", "Height");
```

如果您将鼠标指针置于 "surf1" 并在键盘上按 F11，然后右键单击并选择**切换到节点**，或单击功能区中的**切换到节点**，则会在**编辑器工具**窗口突出显示相应的**表面**图节点。

单击**编辑节点**以打开其**设置**窗口。有关键盘快捷键的更多信息，请参见第 311 页的 "附录 D - 键盘快捷键"。

方法编辑器中的数据访问

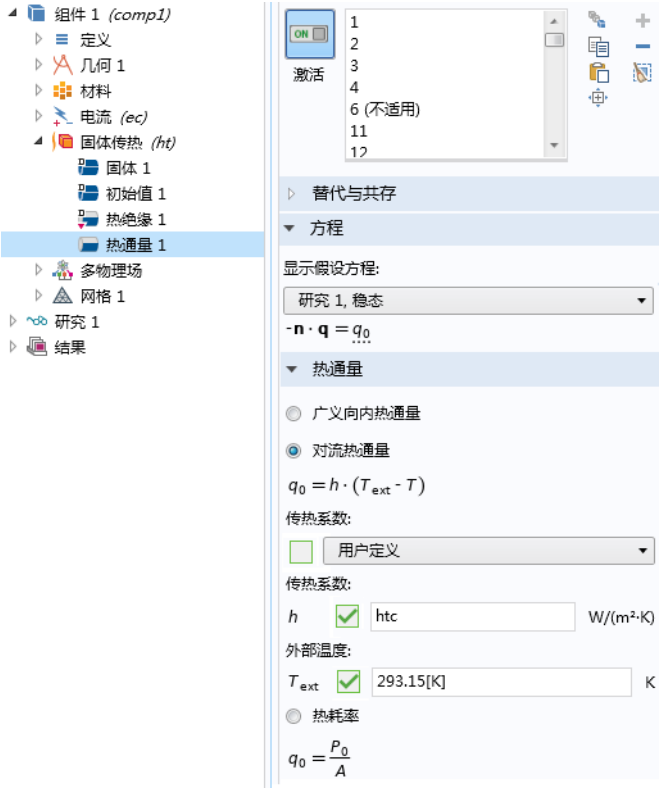
要访问模型树节点的各个属性，可以在 "模型开发器" 功能区中**开发工具**选项卡的 **App** 栏中单击**数据访问**按钮。

此外，对于某些表单对象，也可以在**设置**窗口的**源**栏标题中单击**数据访问**按钮。另请参见第 96 页的 "表单编辑器中的数据访问"。

需要以这种方式启用**数据访问**的原因在于，模型通常包含数百甚至数千个可供访问的属性，这个列表太长，并不实用。

当您单击模型树节点时，例如下图中的**热通量**节点，复选框显示在各个属性旁边。本例基于 *《COMSOL Multiphysics 简介》*中描述的母线板教学案例。

在下图中，选中了**传热系数**和**外部温度**复选框：



如果您切换到**编辑器工具**窗口，将看到在**热通量**节点下显示了更多节点。右键单击并使用**获取**或**设置**，在活动的方法窗口中生成代码，如下图所示。

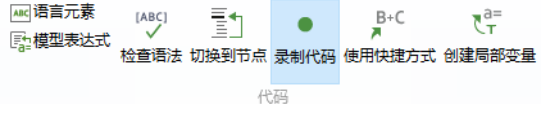
在上例中，**传热系数**和**外部温度**属性的**获取**和**设置**选项将生成以下代码：

```
model.physics("ht").feature("hf1").getString("h");
model.physics("ht").feature("hf1").getString("Text");

model.physics("ht").feature("hf1").set("h", "htc");
model.physics("ht").feature("hf1").set("Text", "293.15[K]");
```

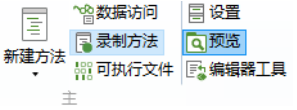
录制代码

在 "方法编辑器" 功能区的**代码**部分单击**录制代码**按钮，可以录制您在模型树中执行的一系列操作，如下图所示。



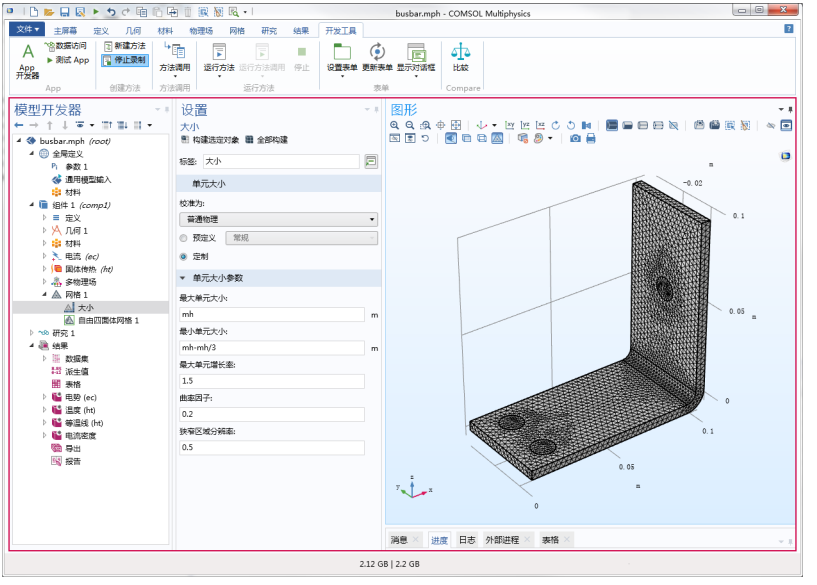
也可以录制您在 App 树中执行的某些操作，包括在 App 运行时用来修改用户界面的方法，例如更改文本标签的颜色。

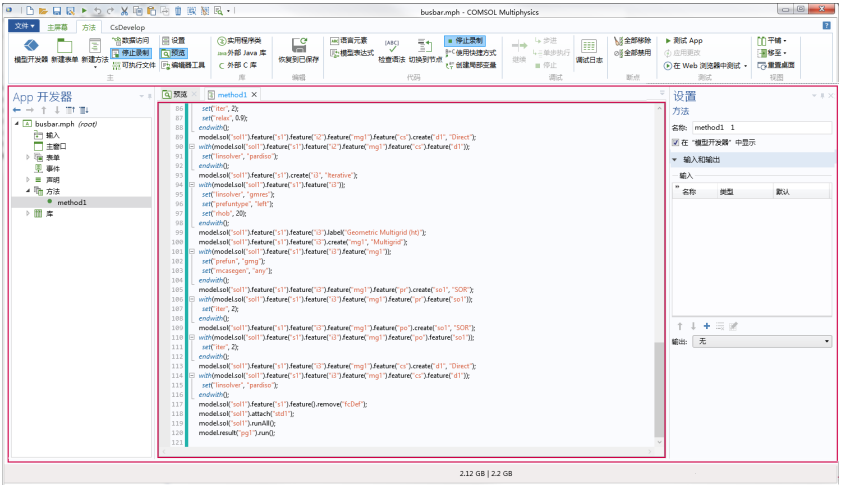
要录制新方法，可以单击 “表单编辑器” 或 “方法编辑器” 功能区**主**栏中的**录制方法**按钮。



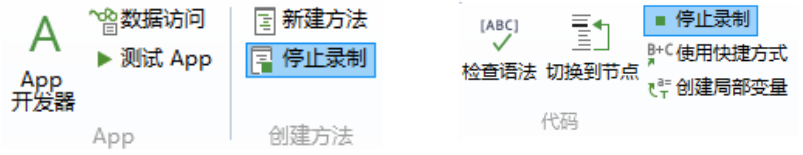
也可以在 “模型开发器” 功能区的**开发工具**选项卡中单击**录制方法**按钮。

在录制代码时， COMSOL Desktop 窗口周围会显示一个红色框架：





要停止录制代码，可以单击“模型开发器”或“App 开发器”功能区中的**停止录制**按钮。



之前的**数据访问**章节解释了如何设置母线板教学案例的**传热系数**和**外部温度**属性值。要使用**录制代码**按钮生成类似代码，请遵循以下步骤：

- 基于母线板模型（MPH 文件）创建一个简单的 App。
- 在“模型开发器”窗口中，单击**录制方法**，或打开“方法编辑器”并单击**录制代码**。
- 将**传热系数**的值改为 5。
- 将**外部温度**的值改为 300[K]。
- 单击**停止录制**。
- 如果包含录制代码的方法尚未打开，则打开它。

下面列出生成的代码：

```
with(model.physics("ht").feature("hf1"));
  set("h", "5");
  set("Text", "300[K]");
endwith();
```

在本例中，自动录制功能包含 `with()` 语句，使代码更紧凑。有关使用 `with()` 的更多信息，请参见第 192 页的 “With 语句”。

要生成与 App 对象更改对应的代码，可以使用**录制代码**或**录制方法**，然后转至“表单编辑器”，执行更改表单对象外观等操作。以下代码对应于将文本标签的颜色从默认的**继承**改为**蓝色**：

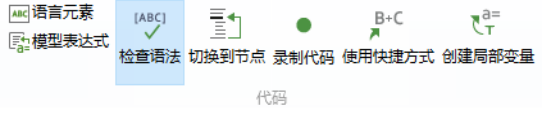
```
with(app.form("form1").formObject("textlabel1"));
  set("foreground", "blue");
endwith();
```

有关修改模型对象和 App 对象的更多信息，请参阅 Application Programming Guide。

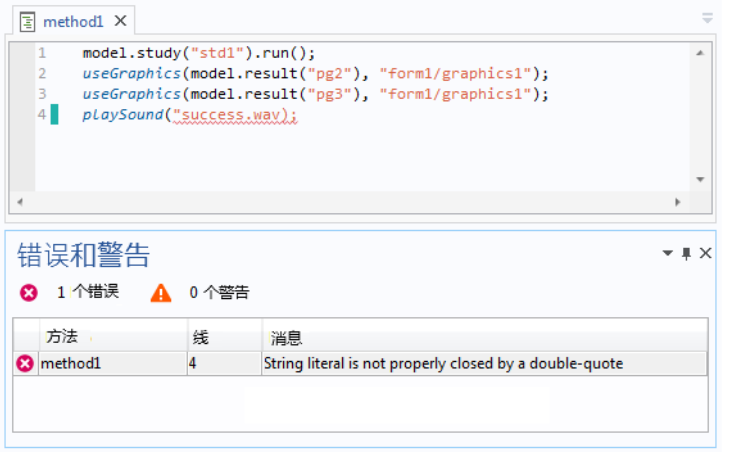
使用代码录制工具可以快速学习如何与模型对象或 App 对象进行交互。自动生成的代码可显示属性、参数及变量的名称。您可以使用字符串和字符串 - 数字转换功能为模型属性的参数赋新值。通过在录制代码时使用**数据访问**，您可以使用获取来提取参数值，并在方法中处理该值，然后使用设置将它设置回模型对象。有关**数据访问**的更多信息，请参见第 169 页的 “方法编辑器中的数据访问”。

检查语法

单击功能区中的**检查语法**按钮，可以查看**错误和警告**窗口中的消息，其中列出相关语法错误或未使用的变量。



除了**错误和警告**窗口中的消息，含语法错误的代码还会以红色波浪下划线标出，如下图所示。



查找和替换

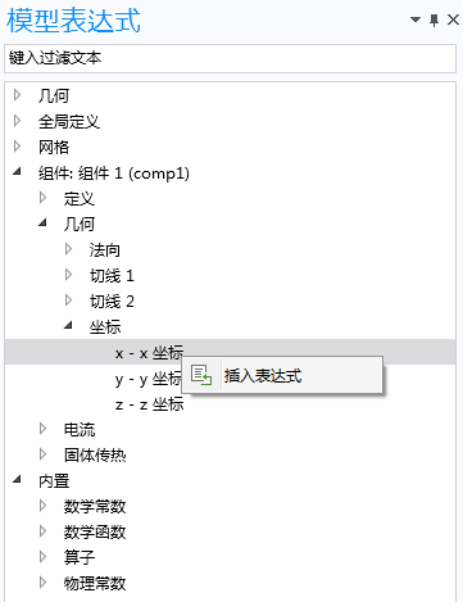
单击快速访问工具栏中的**查找**按钮，可以打开一个对话框，用于查找和替换方法中的字符串，如下图所示。

快速访问工具栏位于 COMSOL Desktop 用户界面中功能区上方的左侧。

全部选项卡用于在 “模型开发器” 和 “App 开发器” 中查找字符串和变量。

模型表达式窗口

“方法编辑器”中的**模型表达式**窗口显示一组预定义的表达式，它们用作输入和输出变元。双击或右键单击列表中的某一项，即可插入表达式：

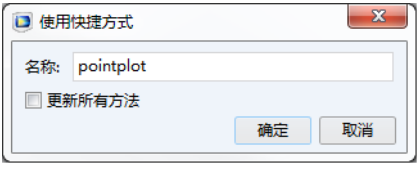


如果您仔细看看以下示例，便会发现，最后两行代码都以 model.result(“pg1”) 开头。

通过**使用快捷方式**按钮，可以将这些实例替换为变量名称，从而简化代码。

在上例中，鼠标指针已放置在出现的第一个 model.result(“pg1”) 上。单击**使用快捷方式**按钮可以将源代码转换为下图所示的内容。

带有前缀 model.result(“pg1”) 的代码已替换为变量 pointplot。单击**使用快捷方式**按钮后会打开**使用快捷方式**对话框，您可以在**名称**框中输入合适的变量名称，本例中为 pointplot。



此变量作为快捷方式存储在**声明**节点中，如下图所示，右侧是对应的**设置**窗口。

语法突出显示、代码折叠和缩进

代码中不同的语言元素显示为不同的样式。请参见下图中的示例：

```
64  with(model.result("pg1"));
65    set("looplevel", new String[]{"7"}); // 7th frequency
66  endwith();
67  useGraphics(model.result("pg1"), "graphics1");
68  zoomExtents("graphics1");
69
70  if (customProgress) {
71    setProgressBar("/progressform/progress1", 100);
72  }
73  else {
74    setProgress(100);
75  }
76  play_sound();
77
78  if (customProgress) {
79    closeDialog("progressform");
80  }
81  else {
82    closeProgress();
83  }
```

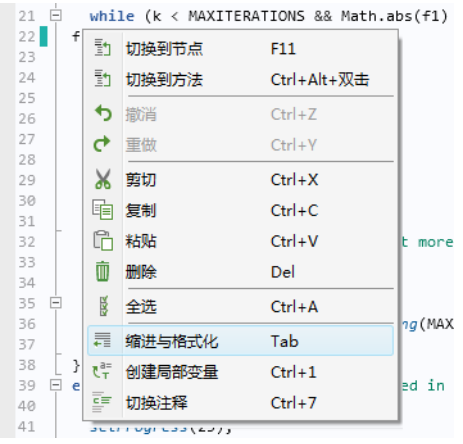
本例包含五种样式：

- 关键字，如 `if`、 `else`、 `for`、 `while`、 `double` 和 `int`，以蓝色粗体显示
- 内置方法以蓝色斜体显示
- 字符串以红色字体显示
- 注释以绿色字体显示
- 代码其余部分则以黑色字体显示

您可以在**首选项**对话框中定制语法突出显示主题。请参见下一节方法编辑器首选项。

您可以展开和折叠对应于代码块的代码部分，例如 `for`、`while`、`if` 和 `else` 语句的一部分。此特征可以被禁用，如接下来的 “方法编辑器首选项”一节所述。

在编写代码时，使用键盘上的 Tab 键可以自动缩进代码行，并根据需要插入空格。此外，您也可以右键单击方法编辑器并选择**缩进与格式化**，如下图所示。



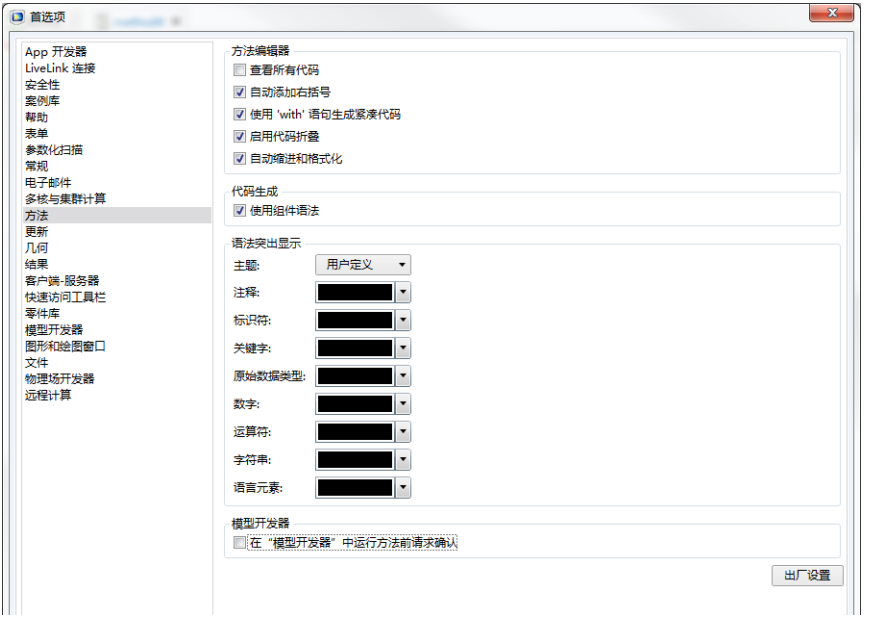
当键盘焦点离开 “方法编辑器”时，也会自动应用缩进和空格。您可以在**首选项**的**方法**栏中，通过清除**自动缩进和格式化**复选框来禁用这一功能。

使用在单击右键时出现的上下文菜单 （如上图所示），您可以打开或关闭选定的整个代码块的注释。具体操作是从菜单中选择**切换注释**，或使用键盘快捷键 Ctrl+7。

方法名称

方法的**名称**是不含空格的文本字符串。其中可以包含字母、数字和下划线。不支持保留名称 `root` 和 `parent`，并且不能使用 Java® 编程语言关键字。

要访问方法的**首选项**，您可以选择**文件 > 首选项**，然后选择**方法**栏。



默认情况下，“方法编辑器”仅显示最相关的代码。要查看方法中的所有代码，请选择**查看所有代码**复选框。

自动添加右括号复选框用于控制 “方法编辑器”是否自动添加右括号，如花括号 {}、方括号 [] 和圆括号 ()。

使用 ‘with’语句生成紧凑代码复选框用于控制 `with` 语句在自动生成的代码中的使用。有关更多信息，请参见第 192 页的 “With 语句”。

如果选中**启用代码折叠**复选框，便可以展开或折叠与 `for`、 `while`、 `if` 和 `else` 语句关联的代码块对应的代码部分。

选中**自动缩进和格式化**复选框将确保代码始终保持缩进和格式化。

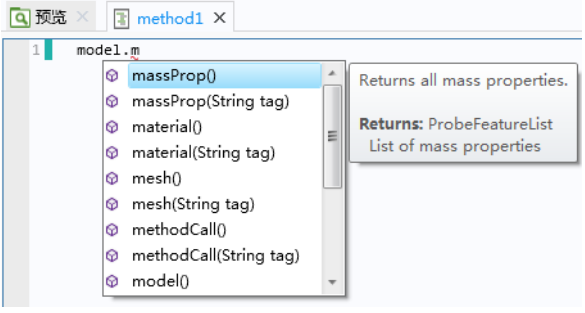
使用组件语法选项可以生成方法语法，其中还包含模型组件范围。

在**语法突出显示**栏中，**主题**列表包含两个预定义的主题：**现代** （默认设置）和**经典**。选择**用户定义**可定义语法突出显示模式，您可以为各个语言元素指派不同的颜色。

如果您不希望在以这种方式运行方法时进行确认，则清除**在 “模型开发器”中运行方法前请求确认**选项。

使用 Ctrl+/ 与 Tab 补全代码

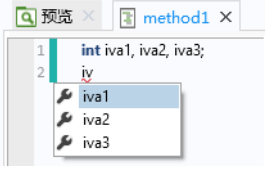
当您在 “方法编辑器” 中键入代码时，“App 开发器” 可以提供一些建议来实现代码补全。一组可用的代码补全选项显示在单独的列表中，并在您键入代码时自动打开。在某些情况下，当您在列表中选定某个条目时，会在单独的窗口中显示详细信息。您可以通过键盘快捷键 Ctrl+/ 来请求代码补全功能。当您访问一部分模型对象时，可以获取代码补全列表，如下图所示：



通过在列表中使用箭头键选择一个条目即可选定一个代码补全项，然后按 Tab 或 Enter 键来确认选择。

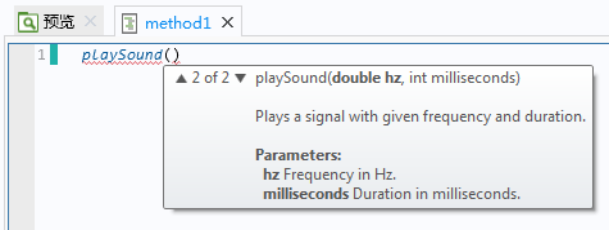
如果该列表较长，您可以键入要查找的补全项的前几个字符来进行过滤。

例如，如果您输入变量或方法名称的前几个字符，然后按 Ctrl+/，则会显示可用的代码补全选项：



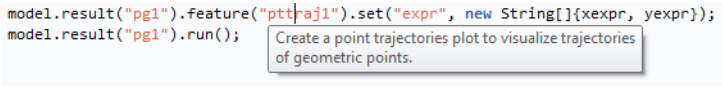
在上例中，仅显示与字符串 `iv` 匹配的变量。从本例可以看出，方法的局部变量也显示在代码补全建议中。

您还可以使用 Ctrl+/ 来了解与模型对象不直接相关的内置方法的语法。键入命令名称并使用 Ctrl+/ 来打开一个窗口，其中包含各种可用的调用签名的相关信息。

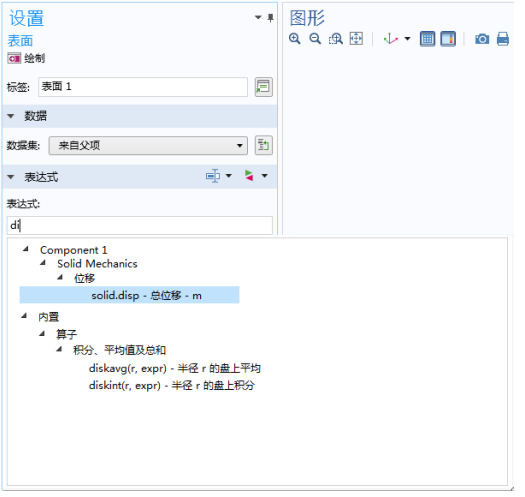


如果获取可用内置方法的列表，可以使用**语言元素**窗口，如第 166 页所述，或参见第 313 页的 “附录 E - 内置方法库”。

将光标悬停在方法调用、属性名称、声明或快捷方式的各部分上方时，工具提示中会显示类似信息。



键盘快捷键 Ctrl+/ 也可在 “模型开发器”中使用。例如，在**结果**的**表达式**框中键入内容时，使用 Ctrl+/ 可以查看匹配的变量，如下图所示。

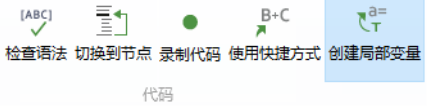


创建局部变量

您可以自动设置局部变量的类型。例如，键入

```
x = model.geom()
```

然后在功能区的 “方法”选项卡的 “代码”组中单击**创建局部变量**按钮。



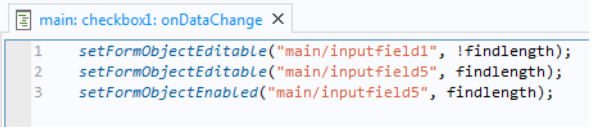
此时代码会更改为

```
GeomList x = model.geom()
```

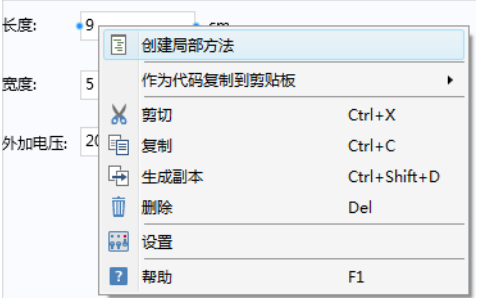
其中， `GeomList` 为数据类型 `model.geom()`。

局部方法

您可以为按钮、菜单项和事件添加局部方法。局部方法不会在 App 树的**方法**节点下显示相关节点。在局部方法的方法窗口中，其选项卡会显示关联用户界面组件的路径，如下图所示的复选框对象。

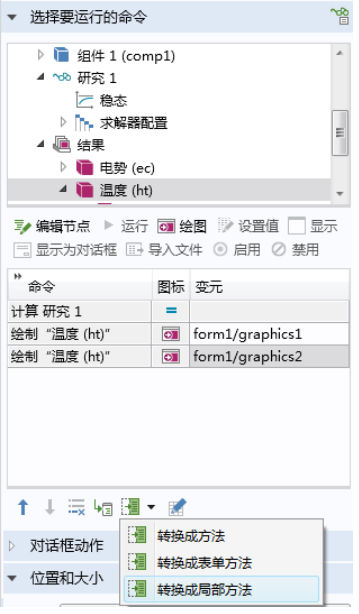


在 “表单编辑器”中，您可以右键单击表单对象，并从菜单中选择**创建局部方法**，如下图所示。

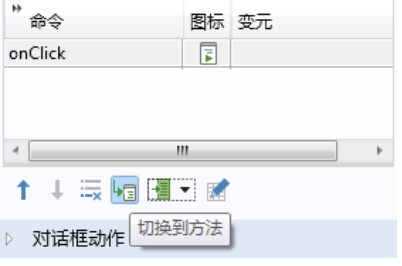


按钮、菜单项和全局事件的局部方法

对于按钮、功能区、菜单、工具栏项和全局事件，您可以通过从命令序列下方的工具栏菜单按钮中选择**转换成局部方法**来添加局部方法，如下图所示。



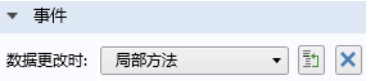
此按钮的功能与第 17 页的 "新建方法" 一节中描述的**转换成方法**和**转换成表单方法**按钮相似。唯一的区别在于它创建的局部方法在 App 树的全局方法列表中不可见。在创建新方法后，还会在 "方法编辑器" 中打开它。您可以使用 "Ctrl+Alt+ 单击" 这一快捷方式来创建局部方法。单击**切换到方法**按钮将打开局部方法。下图显示对按钮关联的局部方法进行调用的情况。



为了避免在局部方法中出现任何破坏代码的风险，当命令序列中显示局部方法时，您无法使用**转换成方法**按钮。

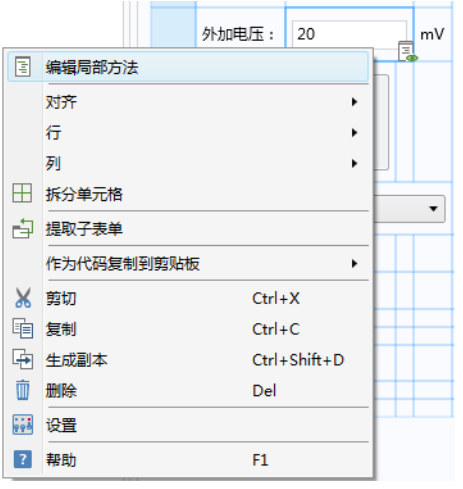
表单和表单对象事件的局部方法

要为表单或表单对象事件添加局部方法，可在**设置**窗口的**事件**栏中单击**创建局部方法**按钮。选定的**数据更改时**方法会从**无**改为**局部方法**，如下图所示，并打开“方法编辑器”。



要在“方法编辑器”中打开现有的局部方法，可以单击**切换到源**按钮。单击**移除局部方法**按钮可删除局部方法。

作为 Ctrl+Alt+ 单击操作的备选方式，您也可以右键单击表单对象并从上下文菜单中选择**编辑局部方法**或**编辑方法**。

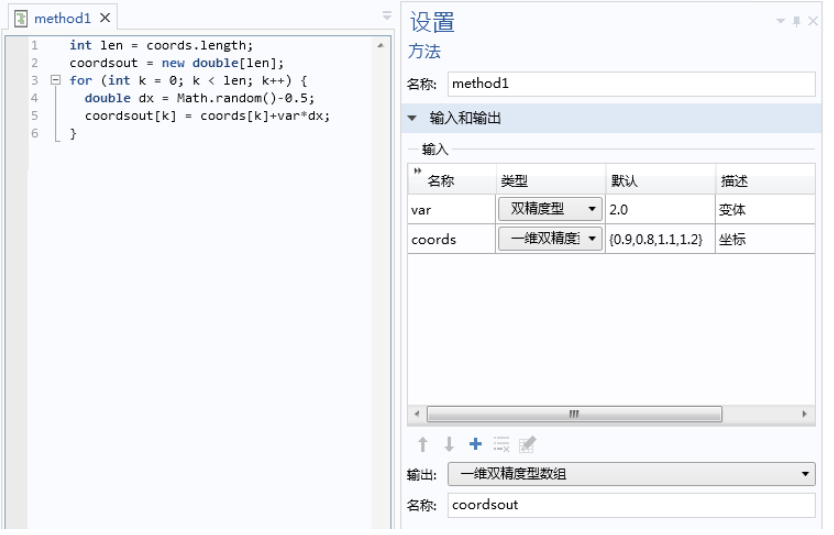


有关更多信息，请参见第 129 页的“事件”。

带输入和输出变元的方法

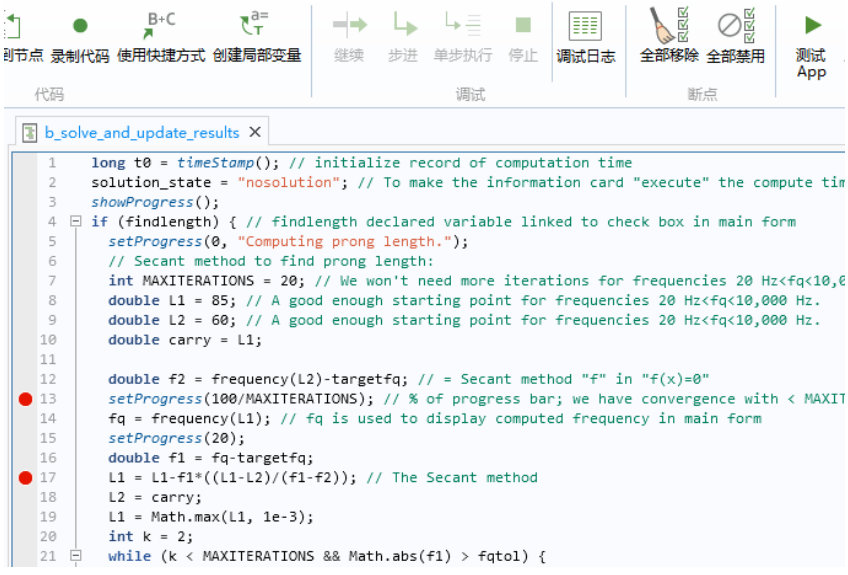
方法可以包含多个输入变元和一个输出变元。您可以在活动方法窗口的**设置**窗口中定义输入和输出变元。如果**设置**窗口不可见，请单击功能区的**方法**选项卡中的**设置**。下图显示的方法包含两个输入变元（var 与 coords）和一个输出变

元 coordsout。该方法用于将随机值添加到数组 coords，随机度由输入变量 var 控制。新值存储在数组 coordsout 中。



当您从一个方法调用另一个方法时， Ctrl+Alt+ 双击操作可以打开该方法对应的窗口。方法可以调用其本身以形成递归。

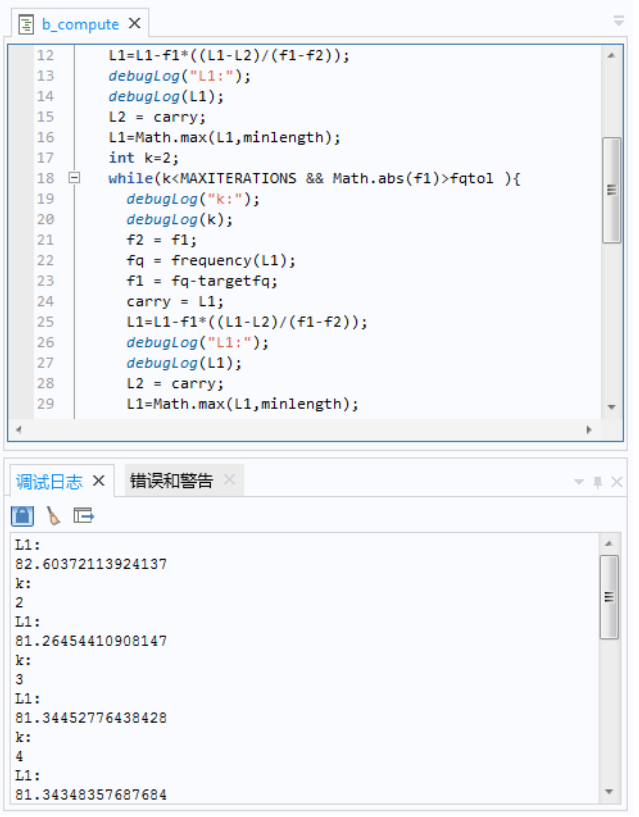
要对代码进行调试，您可以单击代码行号左侧的灰色列以设置断点，如下图所示。



功能区中的**调试**组包含用于调试方法的工具。当您运行 App 时，方法将在断点处停止。单击**步进**按钮以进入方法的下一行。上图显示的方法当前在黄色突出显示的行停止。

单击**继续**以运行该方法直到下一个断点。单击**停止**以停止正在运行的方法。单击**单步执行**以步进到下一个方法 （如果有）。使用**全部移除**可移除所有断点。

除了移除，您还可以通过单击**全部禁用**来禁用所有断点。单击**调试日志**可以在单独的**调试日志**窗口中显示调试消息，如下图所示。



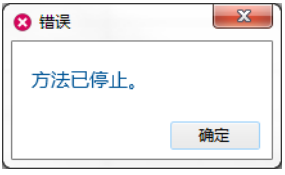
使用 debugLog 命令可以在**调试日志**窗口中显示变量的值。以下代码说明如何使用 debugLog 命令来显示一维双精度数组的字符串和分量的值。

```
int len=xcoords.length;
if (selected==0) {
  for (int i = 0; i < len; i++) {
    double divid=double(i)/len;
    xcoords[i] = Math.cos(2.0*Math.PI*divid);
    ycoords[i] = Math.sin(2.0*Math.PI*divid);
    debugLog(“x:”);
    debugLog(xcoords[i]);
    debugLog(“y:”);
    debugLog(ycoords[i]);
    debugLog(“selected is 0”);
  }
}
```

有关内置调试方法的更多信息，请参见第 322 页的 “调试方法” 和 Application Programming Guide。

停止方法

您可以在测试 App 时使用键盘快捷键 Ctrl+Pause 来停止方法的执行。此时会显示一个对话框，如下所示。



模型对象

模型对象提供了许多方法，可用于设置和运行操作序列等。“方法编辑器” 的**转换成方法**、**录制代码**、**编辑器工具**和**语言元素**实用程序可以使用这类模型对象方法来生成语句。有关模型对象及其方法的更多信息和示例代码，请参阅 *《COMSOL Multiphysics 简介》*一书中的 “附录 C - 语言元素和保留名称”、Application Programming Guide 和 Programming Reference Manual。

语言元素示例

Java® 编程语言用于编写 COMSOL 方法，这意味着您通常可以使用 Java® 语句和语法。本节包含一些最常见语言元素的简单示例。有关更多信息和示例，请参阅 Application Programming Guide 和 Programming Reference Manual。

模型对象中的一元和二元运算符

下表描述在访问模型对象时 （例如在定义材料属性和边界条件时）可以使用的一元和二元运算符，以及在结果中用于后处理和可视化的表达式。

优先级	符号	描述
1	( ) { } .	分组、列表、范围
2	^	幂
3	! - +	一元：逻辑非、差、和

优先级	符号	描述
4	[ ]	单位
5	* /	二元：乘法、除法
6	+ -	二元：加法、减法
7	< <= > >=	比较：小于、小于等于、大于、大于等于
8	== !=	比较：等于、不等于
9	&&	逻辑与
10	||	逻辑或
11	,	列表中的元素分隔符

方法中的一元和二元运算符（JAVA® 语法）

下表描述在方法的 Java® 代码中最重要的一元和二元运算符。

优先级	符号	描述
1	++ --	一元：后缀加法和减法
2	++ -- + - !	一元：加法、减法、正号、负号、逻辑非
3	* / %	二元：乘法、除法、取模
4	+ -	二元：加法、减法
5	!	逻辑非
6	< <= > >=	比较：小于、小于等于、大于、大于等于
7	== !=	比较：等于、不等于
8	&&	二元：逻辑与
9	||	二元：逻辑或
10	?:	三元条件
11	= += -= *= /= %= >>= <<= &= ^= |=	赋值
12	,	列表中的元素分隔符

访问声明节点中的变量

声明节点中定义的变量可以作为方法中的全局变量，无需进一步声明。

内置基本数学函数

方法中使用的基本数学函数依赖于 Java® 数学库。如下例所示：

```
Math.sin(double)
Math.cos(double)
Math.random()
Math.PI
```

IF 语句

```
if(a<b) {
  alert(toString(a));
} else {
  alert(toString(b));
}
```

FOR 语句

```
// Iterate i from 1 to N:
int N=10;
for (int i = 1; i <= N; i++) {
  // Do something
}
```

WHILE 语句

```
double t=0,h=0.1,tend=10;
while(t<tend) {
  //do something with t
  t=t+h;
}
```

WITH 语句

```
// Set the global parameter L to a fixed value
with(model.param());
  set(“L”, “10[cm]”);
endwith();
```

以上代码等同于：

```
model.param().set(“L”, “10[cm]”);
```

访问全局参数

通常，您需要使用**编辑器工具**窗口来生成用于设置全局参数值的代码。在 “方法编辑器”中，右键单击参数并选择**设置**。

将全局参数 L 的值设为 10 cm：

```
model.param().set(“L”, “10[cm]”);
```

获取全局参数 L 并将其存储在双精度变量 Length 中：

```
double Length=model.param().evaluate(“L”);
```

本案例中的计算基于模型树根节点中定义的基本**单位制**。

返回参数 L 的单位 （如果有），则使用：

```
String Lunit=model.param().evaluateUnit(“L”);
```

要将双精度值写入全局参数，需要将其转换为字符串。原因在于全局参数是模型表达式，可以包含单位。

将变量 Length 的值乘以 2，然后将结果写入参数 L 并加上单位 cm。

```
Length=2*Length;
model.param().set(“L”, toString(Length)+”[cm]”);
```

要使用不同于基本 “单位制”的单位返回参数的值，可以使用：

```
double Length_real = model.param().evaluate(“L”,”cm”);
```

如果参数的值比较复杂，可以将实部和虚部作为长度 2 的双精度矢量返回：

```
double[] realImag = model.param().evaluateComplex(“Ex”,”V/m”);
```

比较字符串

要比较 Java® 中的字符串值，需要使用 .equals() 而非 == 运算符。这是因为 == 运算符会比较字符串是否为相同的对象，而忽略它们的值。以下代码演示了如何比较字符串：

```
boolean streq=false;
String a=”string A”;
String b=”string B”;
streq=a.equals(b);
// In this case streq==false

streq=(a==b);
// In this case streq==false

b=”string A”;
streq=a.equals(b);
// In this case streq==true
```

通知和消息

alert、confirm 和 request 方法可以显示一个对话框，其中包含文本字符串和可选的用户输入。以下示例使用 confirm 询问用户在 App 中使用直接求解器还

是迭代求解器。随后， alert 函数会基于用户的回答，在消息对话框中显示针对所选求解器类型估计的内存需求：

```
String answer = confirm(“Which solver do you want to use?“,
“Solver Selection“,”Direct“, “Iterative“);
if(answer.equals(“Direct“)) {
  alert(“Using the direct solver will require about 4GB of memory when
solving.”);
} else {
  alert(“Using the iterative solver will require about 2GB of memory when
solving.”);
}
```

在模型开发器中运行方法

在 “模型开发器”中运行方法与从 App 中调用方法类似，最重要的区别在于，从 “模型开发器”运行时，方法可以直接修改当前会话中的模型对象。通过从 “模型开发器”运行方法，可以将包含多个手动步骤的建模任务自动化。例如，在包含多个研究的模型中，您可以先为 “研究 1”的计算过程录制代码；然后基于 “研究 1”的解计算 “研究 2”；以此类推。在 “模型开发器”中，您可以通过功能区中的**方法调用**或**开发工具**选项卡直接调用方法，本节稍后将对此进行介绍。您也可以通过**设置表单**间接调用方法，例如，通过单击按钮调用方法。有关**设置表单**的更多信息，请参见第 117 页的 “在 “模型开发器”中使用表单”。

不同于从 App 调用的方法，从 “模型开发器”调用的方法无法使用内置图形方法，例如 `printGraphics`、 `useGraphics` 和 `zoomExtents`。造成这一限制的原因是**设置表单**不能包含图形对象 （此限制将在未来版本中解除）。

“模型开发器”中调用的方法可以包含输入和输出变元。通过在**全局定义**下添加**方法调用**节点，可以为直接调用的方法 （而不是从**设置表单**间接调用的方法）提供输入变元，请参见第 199 页的 “方法调用”。

控制模型树节点的活动状态

要控制在 “模型开发器”中运行方法后将哪个模型树节点设为活动状态，可以使用内置方法 `selectNode`。例如，用于修改几何的方法可以在最后一行使用以下代码：

```
selectNode(model.component(“comp1”).geom(“geom1”));
```

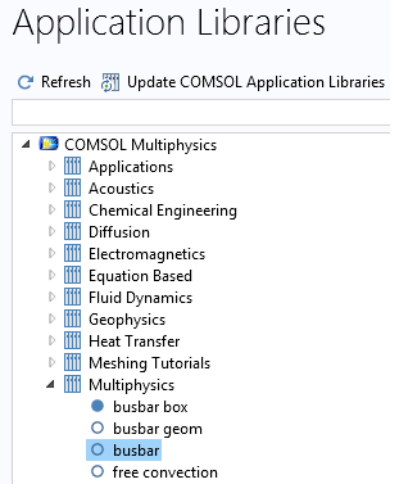
该代码用于显示几何并选择**几何**节点。

在 App 中使用 `selectNode` 方法时将不起作用。

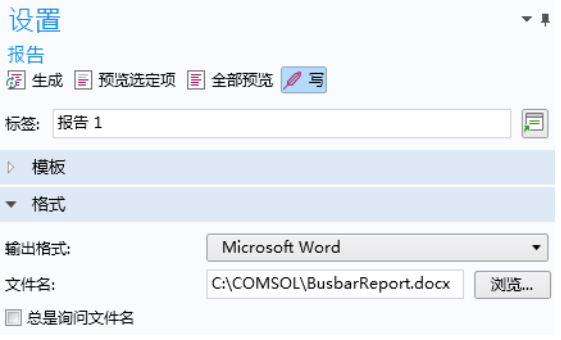
计算后自动生成报告

在举例说明从 “模型开发器”调用方法时，我们需要考虑这样一个过程：先计算解，再生成报告。这一过程可以自动实现，方法是先录制 “模型开发器”中的相应操作，然后运行方法。

我们从 *《COMSOL Multiphysics 简介》*一书中描述的母线板示例着手。您可以从 “案例库” 中加载该示例 MPH 文件，如下图所示。

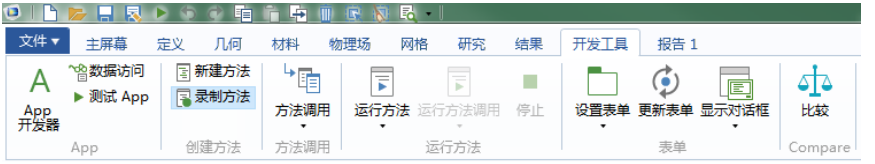


- **打开**模型。
- 在模型树中，右键单击**结果**节点下的**报告**节点。
- 选择**简要报告**。
- 将**输出格式**改为 Microsoft® Word （本例也可以使用默认的 HTML 格式）。
- 单击**浏览**按钮，并在系统中您拥有写权限的位置选择一个文件名，例如：`C:\COMSOL\BusbarReport.docx`
- 单击**写**以生成报告并将其保存到文件。



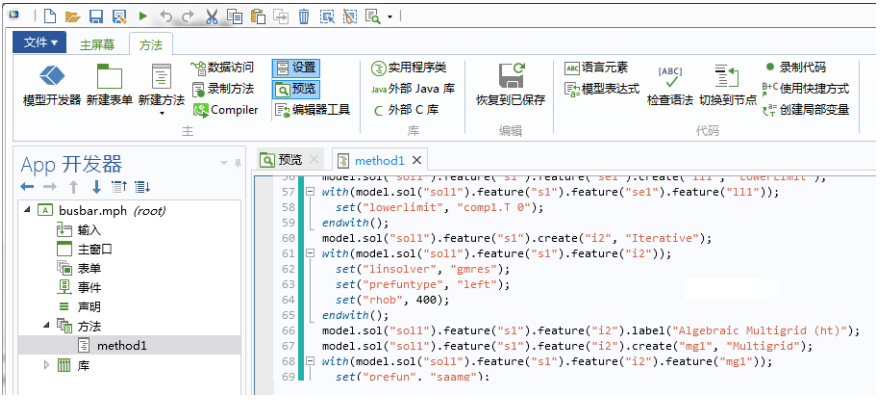
- 关闭自动打开的 Microsoft® Word 文档。

- 单击功能区中的**开发工具**选项卡 （位于**模型开发器**），然后单击**录制方法**按钮。



- 在模型树中，右键单击**研究 1** 节点并选择**计算** （或使用**计算**的功能区选项）。
- 在模型树中，右键单击**报告 1** 节点并选择**写**。
- 单击功能区中的**开发工具**选项卡，然后单击**停止录制**按钮。

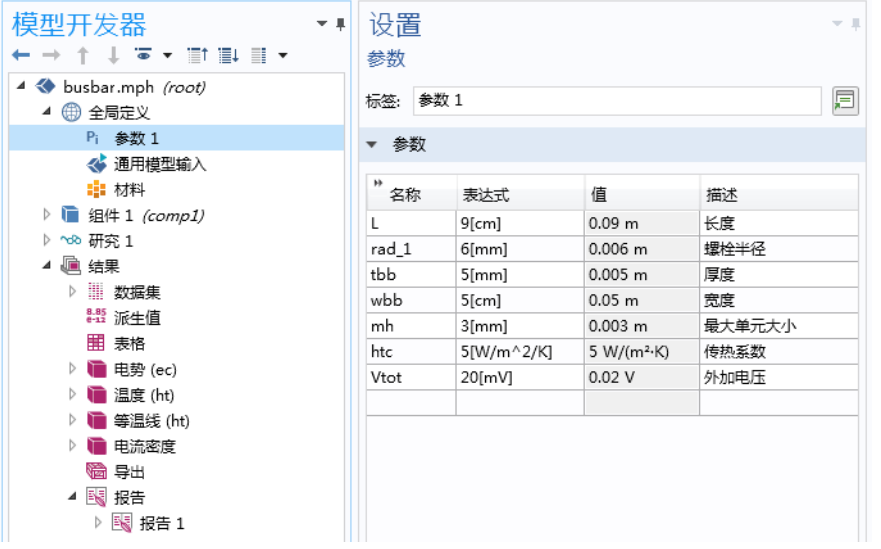
- 关闭 Microsoft® Word 文档。
- 现在，您可以通过单击功能区中的 **App 开发器**按钮切换到 “App 开发器”，并在 App 树和 “方法编辑器” 中查看录制的方法。



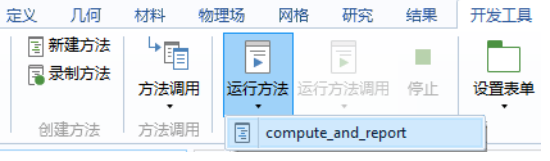
有关从**研究**节点生成的代码的更多信息，请参阅 Application Programming Guide。

- 您可以在此检查和编辑生成的代码，以及更改方法名称 （例如改为 **compute_and_report**）。

- 单击功能区中的**模型开发器**按钮以切换回 “模型开发器”。
- 在**全局定义** > **参数**的 “设置” 窗口中，将**长度**改为 15[cm]。



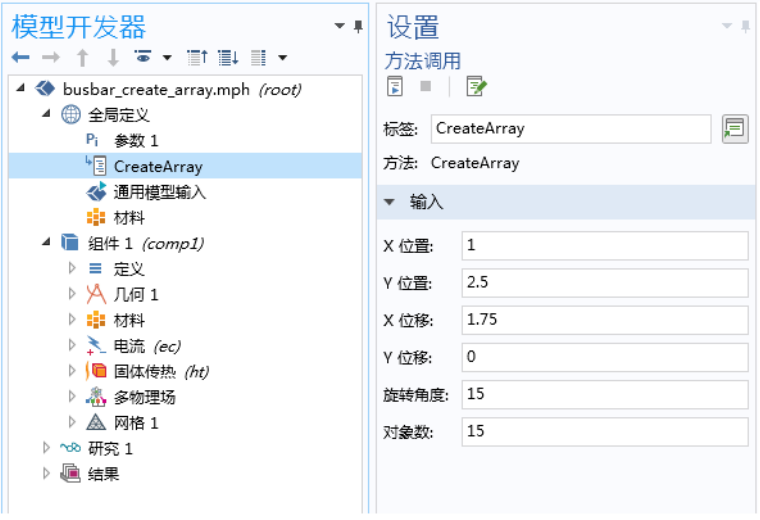
- 在功能区中单击**开发工具**选项卡，然后从**运行方法**菜单中选择 **compute_and_report**。



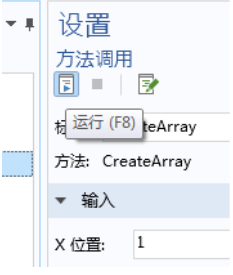
请注意，您可以创建多个方法，然后从 “模型开发器” 中进行调用。

方法调用

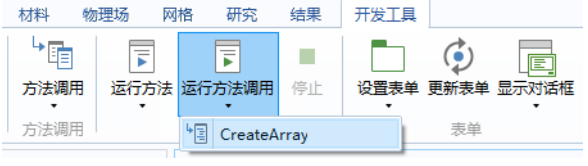
在“模型开发器”中针对一组特定输入变元值的方法调用可以通过在**全局定义**节点下添加**方法调用**节点来实现。要添加**方法调用**节点，右键单击**全局定义**并选择一个已创建的方法。下图显示的**方法调用**所对应的方法用于创建几何阵列。



您可以运行、停止或编辑**方法调用**，在**设置**窗口中单击相应的工具栏按钮即可实现，如下图所示。



“模型开发器”功能区中的**开发工具**选项卡也提供此功能，如下图所示。



下图显示 “App 开发器”中对应的方法**设置**窗口，其中包含输入变元定义。

设置

方法

名称: create_array

☑ 在 "模型开发器" 中显示

▼ 输入和输出

输入

名称	类型	默认	描述	单位
xb	双精度型	1	X 位置	
yb	双精度型	2.5	Y 位置	
dx	双精度型	1.75	X 位移	
dy	双精度型	0	Y 位移	
ang	双精度型	15	旋转角度	
scl	双精度型	1	缩放	
nobjects	整型	13	对象数	

输出: 无

您可以为同一方法添加多个**方法调用**节点，每个调用都可以包含一组不同的输入变元值。

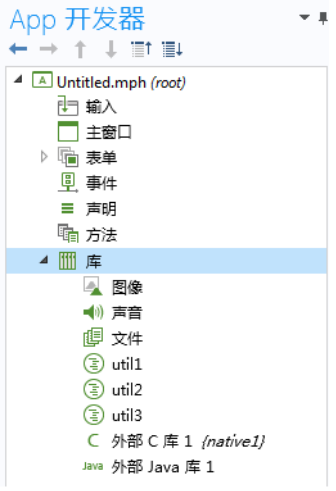
在 “模型开发器”中，无法直接使用方法中的输出变元。不过，您可以通过调用内置方法 `message` 在 COMSOL Desktop 环境的**消息**窗口中显示方法中使用的变量。下例演示如何在**消息**窗口中显示两个双变量 `width` 和 `depth` 的值：

```
message(“Width: “+toString(width));
message(“Depth: “+toString(depth));
```

如需进行调试，您可以改用 `debugLog` 方法，如第 188 页的 “调试”一节所述。

库

在 App 树中，**库**节点包含要嵌入到 MPH 文件的图像、声音及文件，使您不必将其随 App 一起分发。此外，**库**节点还可以包含 Java® 实用程序类节点以及外部 Java® 和 C 库的节点。有关使用实用程序类和外部库的更多信息，请参阅 Application Builder Reference Manual。



通过使用 `embedded:///file1`、 `embedded:///file2` 等语法，可以在表单对象或方法中引用嵌入文件。例如，要引用图像文件 `compute.png`，可以使用语法 `embedded:///compute.png`。

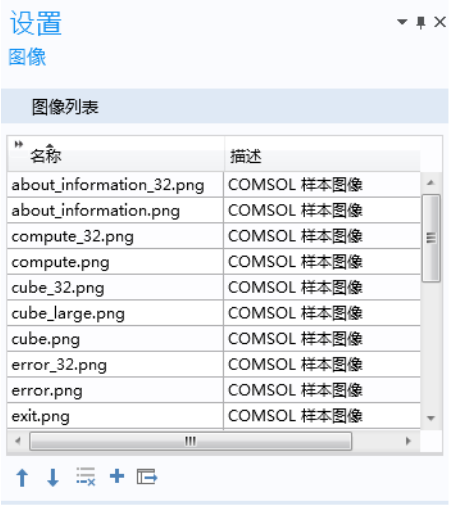
请注意，您无需在文件名中包含扩展名，可以使用任意名称。要最大程度地减小 MPH 文件的大小，可以删除未使用的图像、声音或其他文件。

> 要管理 App 用户在运行时加载的文件，您有多种选择，例如使用**文件**声明和**文件导入**表单对象。有关要在运行时加载的文件的更多信息，请参见第 149 页的 “文件”、第 255 页的 “文件导入” 和第 290 页的 “附录 C - 文件处理和文件方案语法”。

图像

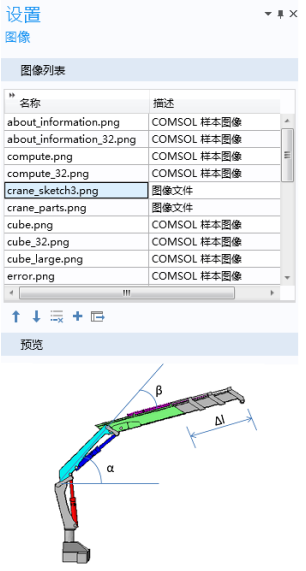
图像库包含许多 PNG 文件格式的预加载样本图像。如果您希望嵌入其他图像文件，可以单击**图像列表**下方的**将文件添加到库**按钮。 COMSOL 安装文件夹中提供了大量的图像供您选择，位于 `data/images`。这些图像用作图标，可以在图

像表单对象或方法中引用。用作图标的图像具有两种大小：16x16 像素 （小）和 32x32 像素 （大）。



支持的图像格式包括 JPG、 GIF、 BMP 和 PNG。

要预览图像，可以在**图像列表中单击图像的名称。**图像会显示在**预览**栏中，如下图所示。



要导出选定的图像，可以单击**预览**按钮右侧的**导出选定的图像文件**按钮。

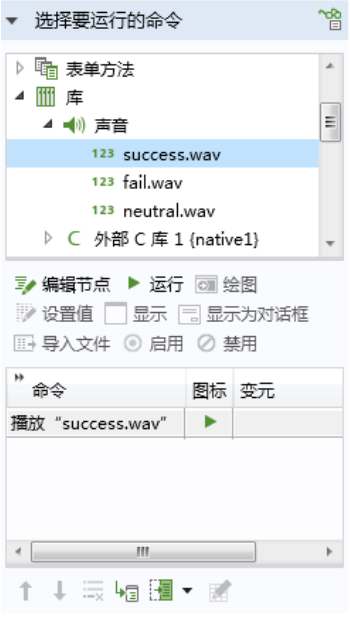
声音

声音库包含一些 WAV 格式的预加载声音文件。如果您希望嵌入其他声音文件，可以单击**声音列表**下方的**将文件添加到库**按钮。 COMSOL 安装文件夹中提供了大量的声音供您选择，位于 `data/sounds`。

要播放声音，可以单击声音的名称，然后单击**声音列表**下方的**预览**按钮。

单击**预览**按钮右侧的**导出选定的声音文件**按钮，可以导出选定的声音。

要在 App 中播放声音，可以在按钮、功能区、菜单或工具栏项的**设置**窗口中添加命令。在**选择要运行的命令**栏中选择声音，然后单击树下方的**运行**按钮。即可将**播放**命令添加到命令序列中，如下图所示。

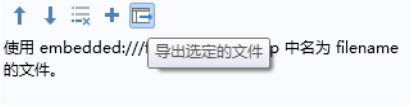


在方法中，您可以使用内置方法 `playSound` 来播放声音，例如：

```
playSound("success.wav");
```

默认情况下，**文件**库为空。单击**将文件添加到库**按钮可将任意类型的文件嵌入App。

单击**将文件添加到库**按钮右侧的**导出选定的文件**按钮，可以导出选定的文件。



可以使用 `embedded:///data1.txt`、 `embedded:///data2.txt` 等语法在方法中引用嵌入文件。有关更多信息，请参见第 149 页的 “文件”、第 290 页的 “附录 C - 文件处理和文件方案语法”和第 314 页的 “文件方法”。

附录 A - 表单对象

本附录提供有关表单和表单对象的信息，对第 46 页的 “表单编辑器”一节的内容进行扩展。以下列表中后跟 * 的项已在该节中进行了详细描述。因此，本附录主要讨论其余各项。

表单对象列表

- 输入
  - 输入框 *
  - 按钮 *
  - 切换按钮
  - 复选框
  - 组合框
- 标签
  - 文本标签 *
  - 单位 *
  - 方程
  - 线
- 显示
  - 数据显示 *
  - 图形 *
  - 网页
  - 图像
  - 视频
  - 进度条
  - 日志
  - 消息日志
  - 结果表
- 子表单
  - 表单
  - 表单集合
  - 卡片堆

- 复合
  - 文件导入
  - 信息卡片堆
  - 数组输入
  - 单选按钮
  - 选择输入
- 其他
  - 文本
  - 列表框
  - 表格
  - 滑块
  - 超链接
  - 工具栏
  - 隔片

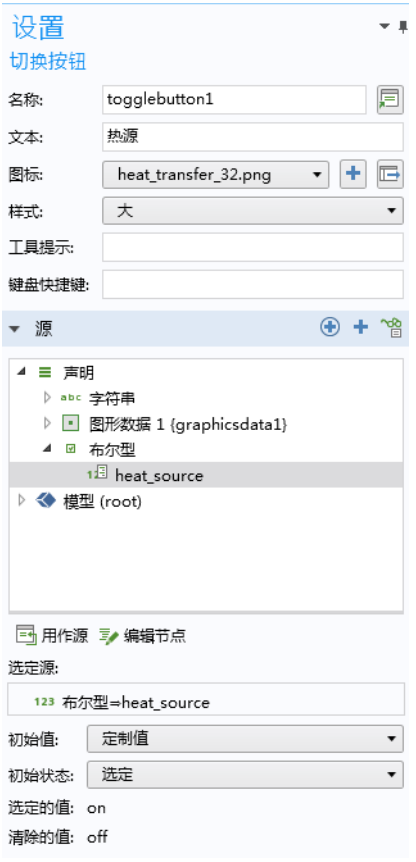
切换按钮

切换按钮对象具有两种状态：选定和取消选定，如下图所示。

本节中的信息也适用于**菜单切换项**和**功能区切换项**。

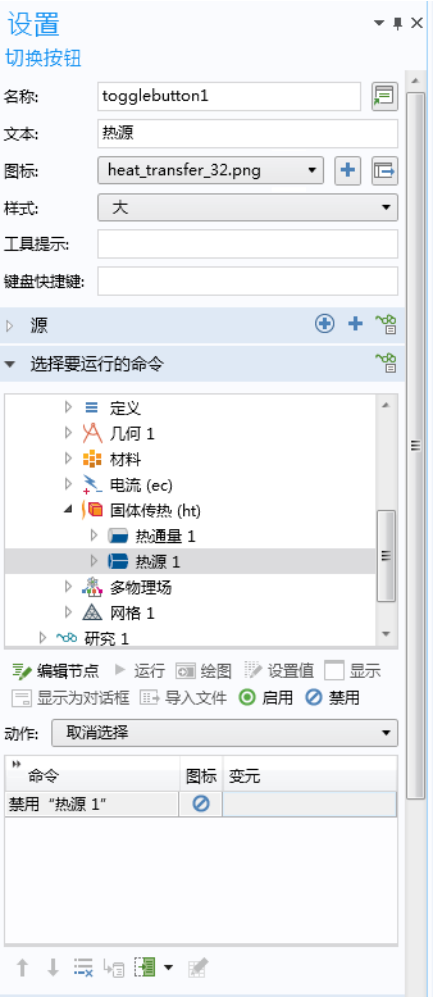
使用切换按钮启用和禁用热源

通过将切换按钮的两种状态链接到布尔变量，可以对状态进行存储。下图显示一个根据自身状态来启用和禁用热源的按钮的**设置**窗口。**源**栏中的布尔变量 `heat_source` 处于选定状态。

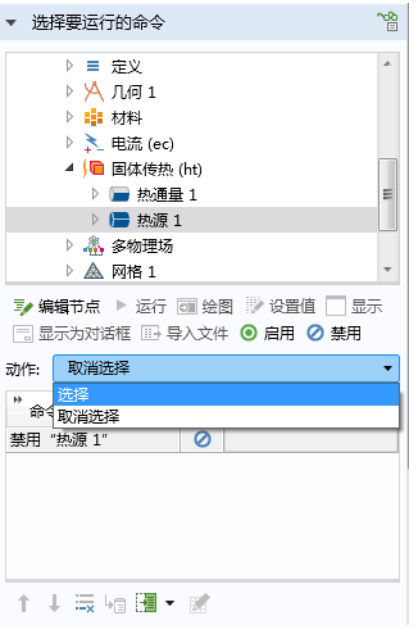


启用对应的布尔变量 `heat_source` 等于 `true`，也就相当于选定切换按钮。禁用对应的布尔变量 `heat_source` 等于 `false`，也就相当于取消选定切换按钮。

源栏下方是**选择要运行的命令**栏，其中包含**动作**选择，用于表示**选择**和**取消选择**两个不同的命令。在下图显示的**设置**窗口中，**取消选择**动作包含**禁用热源**命令。



下图显示 “选择”动作的命令序列，其中包含 “启用热源”命令。



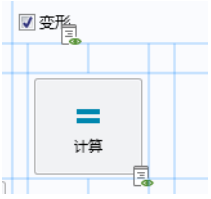
切换按钮与复选框类似，也链接到布尔变量。对于切换按钮，可以使用命令序列来定义动作，而对于复选框，则使用事件来定义动作。请参见下一节中的描述。

复选框

复选框具有两个值：on 表示选定， off 表示清除。复选框的状态存储在**声明**节点的布尔变量中。

使用复选框控制可视化

下图来自一个 App，通过是否选中复选框来控制变形图的禁用或启用状态。



左侧屏幕截图显示正在运行的 App。右侧屏幕截图显示栅格布局模式下对应的表单对象。

在以下示例中，复选框的状态存储在布尔变量 `deformation` 中，其**设置**窗口如下图所示。

下图显示复选框的**设置**窗口。

通过从**源**栏的树中选择已声明的布尔变量并单击**用作源**，可以将复选框与之关联。

默认情况下，复选框的文本标签会从关联布尔变量的**描述**框中获取名称。

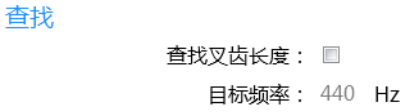
变量 `deformation` 的**初始值**被**选定的值** (`on`) 或**清除的值** (`off`) 覆盖，且无需进行编辑。在方法中使用时， `on` 和 `off` 值分别是 `true` 和 `false` 的别名。这些值可以在 `if` 语句中用作布尔变量。

以下代码语句来自局部方法，该方法在布尔变量 `deformation` 的值更改时针对**数据更改时**事件运行。

```
model.result("pg1").feature("surf1").feature("def").active(deformation);
useGraphics(model.result("pg1"), "graphics1");
```

使用复选框启用或禁用表单对象

下图显示某个 App 的一部分，根据是否选中复选框来禁用或启用某些输入框。



下图显示与布尔变量 `findlength` 关联的复选框的**设置**窗口，该布尔变量用来存储复选框的状态。

以下代码语句来自局部方法，该方法在布尔变量 `findlength` 的值更改时针对**数据更改时**事件运行。

```
setFormObjectEditable(“main/inputfield1”, !findlength);
setFormObjectEditable(“main/inputfield5”, findlength);
setFormObjectEnabled(“main/inputfield5”, findlength);
setFormObjectEditable(“main/inputfield6”, findlength);
setFormObjectEnabled(“main/inputfield6”, findlength);
solution_state = “inputchanged”;
```

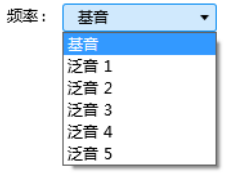
组合框

组合框既可以作为下拉列表框与可编辑文本框的组合，也可以作为无编辑功能的下拉列表框。

使用组合框更改结果中的参数

我们可以采用一个 App 来阐述如何使用组合框，用户可以从六种不同的振型中选择一个，以在结构振动分析中将其可视化。本例使用 “固体力学” 物理场接口和特征频率研究，这种组合适用于任何此类分析。

这六个振型对应于用户从组合框中选择的六个不同的特征频率：



在本例中，组合框用来控制字符串变量 mode 的值。下图显示该变量的**设置**窗口。

选择源

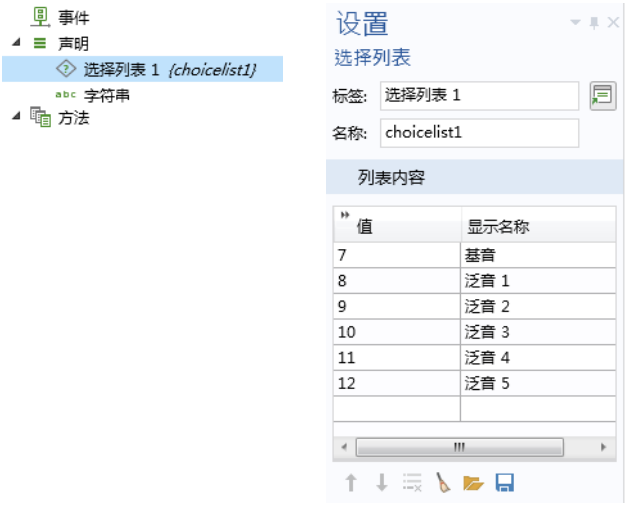
下图显示组合框的**设置**窗口。

在**源**栏中，您可以选择一个由组合框控制值的标量变量，然后单击**用作源**。在组合框**设置**窗口的**初始值**列表中，选择一个方法来定义组合框的默认值。其中包含以下选项：**第一个许用值** （默认设置）和**定制默认值**。对于**定制默认值**选项，需要在关联框中输入默认值。输入的默认值必须为许用值。

选择列表

振动模态 1-6 对应于普通的刚体模态，不是该 App 要分析的范围，因此我们需要的第一个模态是 7。选择列表支持您仅将**显示名称**列中的字符串设为可见，从而对用户隐藏模型中的实际模态值；前几个非刚体模态名为 “基音”、“泛音 1”、“泛音 2”，等等。

在**选择列表**栏中，您可以添加选择列表，为组合框提供许用值。与本例关联的**选择列表**声明如下图所示。



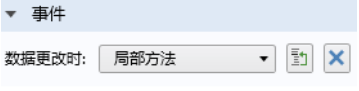
字符串变量 mode 可具有以下六个值之一：7、8、9、10、11 或 12。**显示名称**列中的文本字符串显示在组合框中。

在组合框的**设置**窗口中，您可以选中**允许其他值**复选框，以便在组合框中键入任意值。此类组合框可以接受任意值，不限于由选择列表定义的值。不过，在本例中，只允许使用六个预定义的值。

有关选择列表的更多信息，请参见第 147 页的 “选择列表”。

事件

在**事件**栏中，指定当用户更改组合框的值时要运行的方法，并因此将字符串变量用作源。在本案例中，变量 mode 的值已发生更改，将运行局部方法，如下图所示。



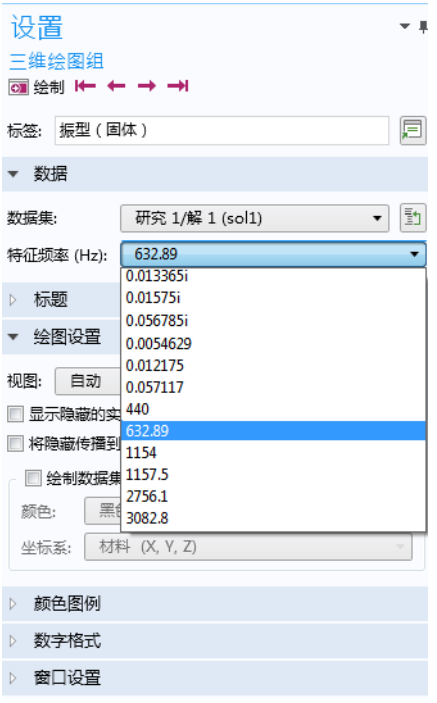
下面列出局部方法的代码。

```
with(model.result("pg1"));
  set("looplevel", new String[]{mode});
endwith();
model.result("pg1").run();
```

这段代码将字符串 mode 的值链接到绘图组 pg1 的特征频率设置。此时，字符串 svar 取值为 “7“、“8“、“9“、“10“、“11“ 或 “12“。

以上代码可通过使用 “方法编辑器”的录制功能自动生成：

- 转到 “模型开发器”，在**开发工具**选项卡中单击**录制方法**。
- 默认情况下，在使用 “特征频率”研究进行结构力学分析时，会创建**振型**绘图组。在此绘图组中，可以将**特征频率**模态从 7 改为 8。在下图中，对应于在**振型**绘图组的**设置**窗口中，从 440 Hz 改为 632.89 Hz。



- 单击**停止录制**。

生成的代码如下所示。

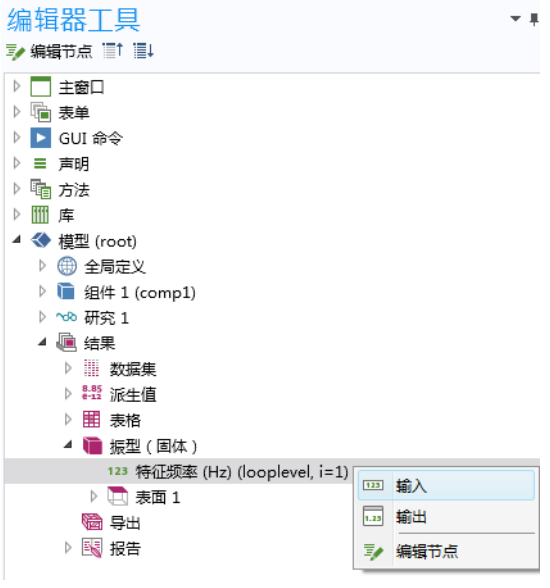
```
with(model.result(“pg1”));
  set(“looplevel”, new String[]{“8”});
endwith();
model.result(“pg1”).run();
```

现在，将字符串 “8”改为 mode 变量，完成上述代码。这段代码将存储在方法 method1 中。从 method1 中复制代码，可以创建与组合框关联的局部方法。然后删除 method1。

使用数据访问

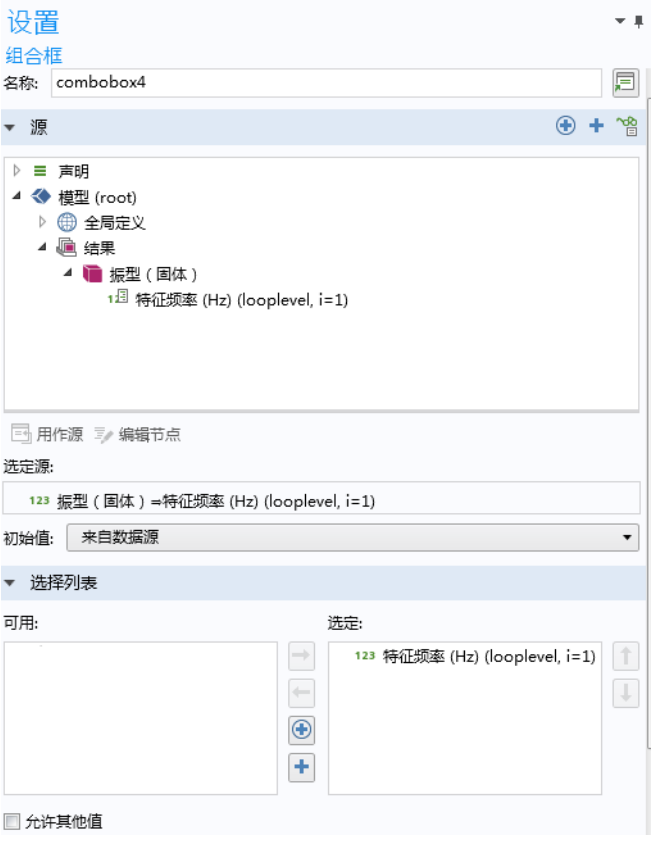
另一种更快但不太常规的使用组合框的方式，是将**数据访问**与**编辑器工具**结合使用。对于本节使用的示例，您可以启用**数据访问**，然后在**振型**绘图组的**设置**窗口中选择**特征频率**，如下图所示。

在**编辑器工具**窗口中，**特征频率**参数显示为**循环级别**。右键单击**循环级别**并选择**输入**，即可创建组合框。



通用名称**循环级别**用于解参数。如果一个解有多个参数，您便可以选择多个循环级别。

下图显示对应组合框的**设置**窗口。



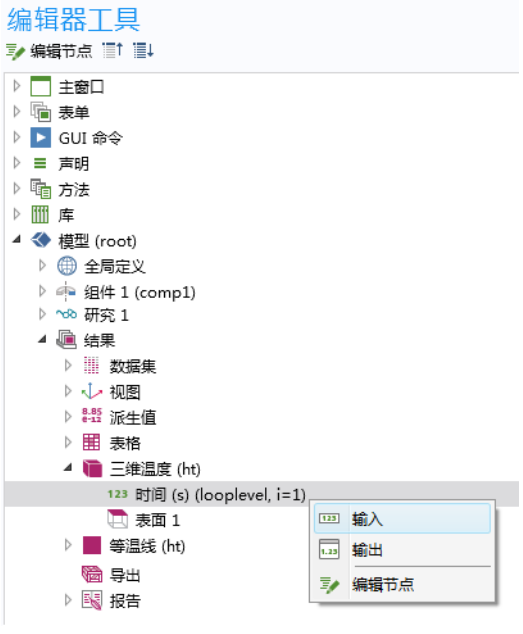
在使用**编辑器工具插入组合框时，会自动生成选择列表循环级别**。请注意，以这种方式生成的选择列表不会显示在**声明**节点下，也不能被用户修改。为增加灵活性，例如为每个参数或特征频率值命名，您需要手动声明选择列表，如上一节所述。

使用组合框更改时间

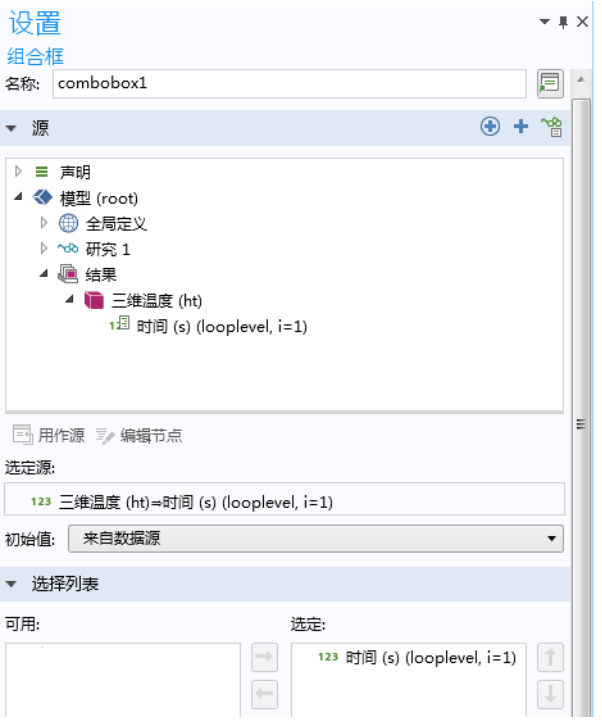
瞬态研究步骤中指定的时间参数列表可用于**结果**节点下的多个位置。在 App 中，可以使用与上一节有关参数的描述类似的方式来访问各个时间参数，即，将**数据访问**与**编辑器工具**结合使用。

在下图显示的**设置**窗口中，使用了**数据访问**来访问温度图中的**时间**参数列表。

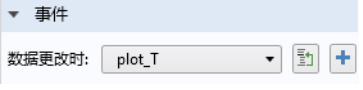
在**编辑器工具**中，现在可以操作**时间**列表，如下图所示。



通过选择**输入**，您可以创建组合框并将其用作**源**，如下图所示。



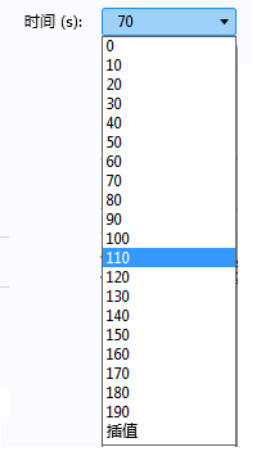
组合框有多种用途，例如，更新不同时间参数所对应的绘图。要在用户使用组合框来选择新的时间参数时自动更新绘图，可在该组合框**设置**窗口底部添加一个事件。在下图中，调用了 `plot_T` 方法来更新温度图。



以下代码行显示方法 `plot_T` 的内容：

```
model.result("pg1").run();
```

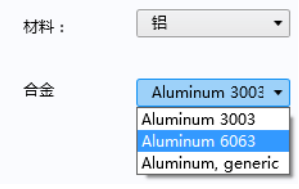
最终结果是在 App 用户界面生成一个组合框，如下图所示，它会在用户从**时间**列表中选择新值时自动更新温度图。

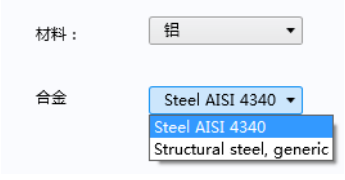


使用组合框更改材料

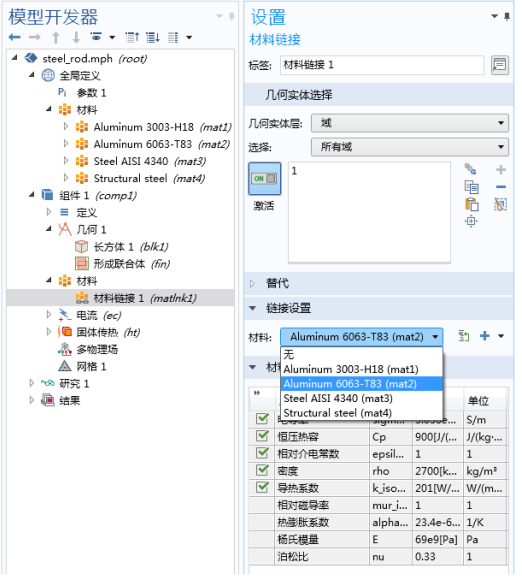
我们以一个通过组合框来选择材料的 App 为例。在本例中，还可以使用激活条件 （请参见第 148 页的 “激活条件”）来增加用户界面设计的灵活性。

以下是来自某个 App 的屏幕截图，用户可以使用名为**材料**的组合框在 **Aluminum** 与 **Steel** 两种材料之间进行选择。另一个组合框名为**合金**，根据**材料**列表中所做的选择来显示 **Aluminum** 合金或 **Steel** 合金列表。





材料选择通过在嵌入模型中使用全局材料和材料链接来实现，如下所示。



每种材料都使用字符串编制了索引：mat1、 mat2... mat5。一个事件用于侦听全局变量 alloy 值的变化情况，该值由组合框控制。该值发生更改时，会运行下面列出的方法。

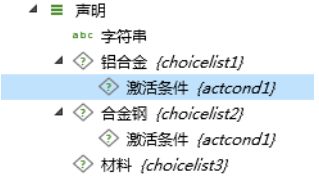
```
with(model.material("matlnk1"));
  set("link", alloy);
endwith();
```

下图显示两个字符串变量（`material` 与 `alloy`）的声明，它们分别由**材料**与**合金**组合框进行控制。

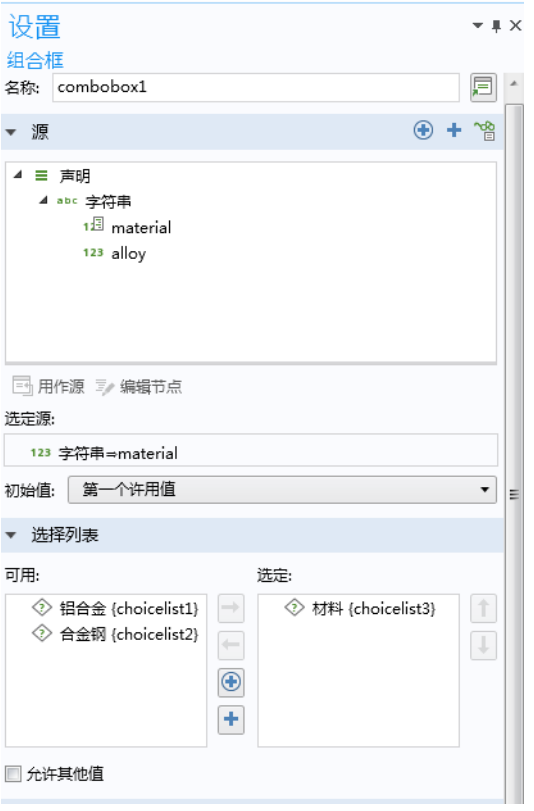
该 App 使用了三个选择列表：**Aluminum 合金**、 **Steel 合金**及**材料**。

激活条件

Aluminum 合金与 **Steel 合金**选择列表中使用了激活条件，如下图所示。

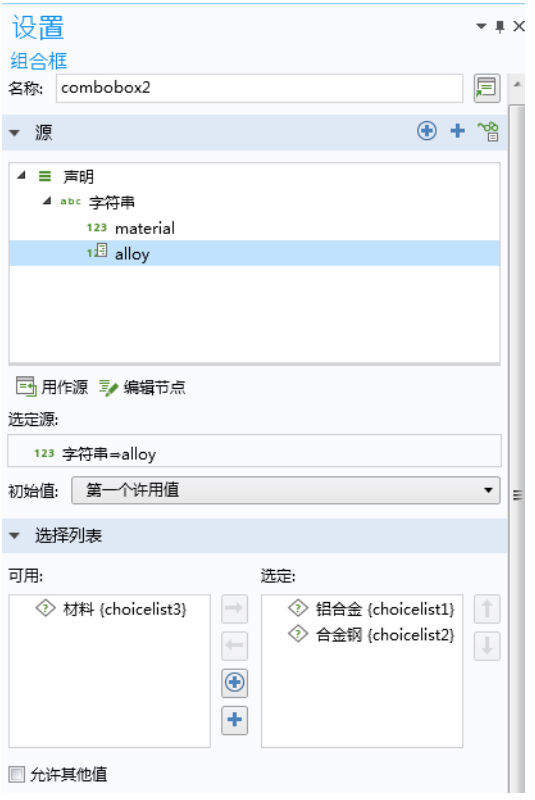


材料组合框的**设置**窗口如下所示。



请注意，**材料**组合框使用 `material` 字符串变量作为源。**材料**选择列表用于为 `material` 字符串变量定义一组离散的许用值。**材料**选择列表的**设置**窗口如下所示。

合金组合框的**设置**窗口如下图所示。

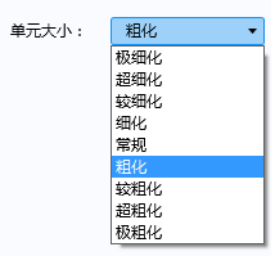


请注意，**合金**组合框同时使用 **Aluminum 合金**与 **Steel 合金**选择列表。**Aluminum 合金**的选择列表如下图所示。

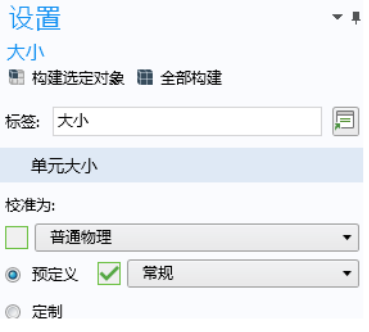
下图显示 **Aluminum 合金**选择列表的激活条件。

使用组合框更改单元大小

在创建组合框时，可以使用**数据访问**功能来重现 “模型开发器”中存在的组合框特征。我们以这样一个 App 为例，其中使用组合框来更改网格的单元大小，如下图所示。



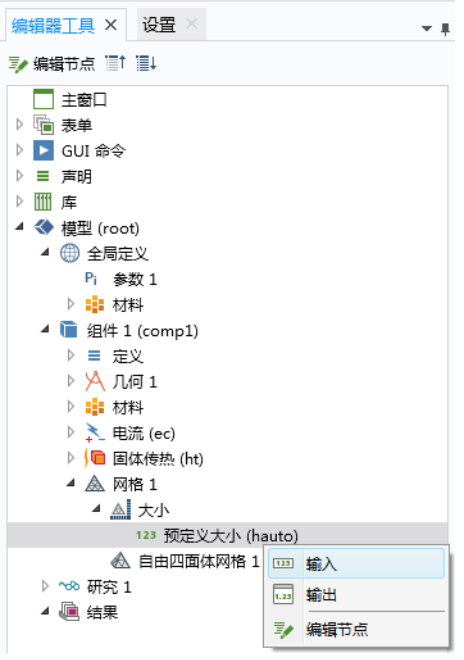
切换到 “模型开发器”，然后选择**网格**节点 （我们假设该模型只有一个网格）。在**网格**节点的**设置**窗口中，选择**用户控制网格** （如果尚未选中）。在**网格**节点下的**大小**节点中，选择**预定义**选项。在功能区中单击**数据访问**。此时便可以访问预定义单元大小的组合框，如下图所示。



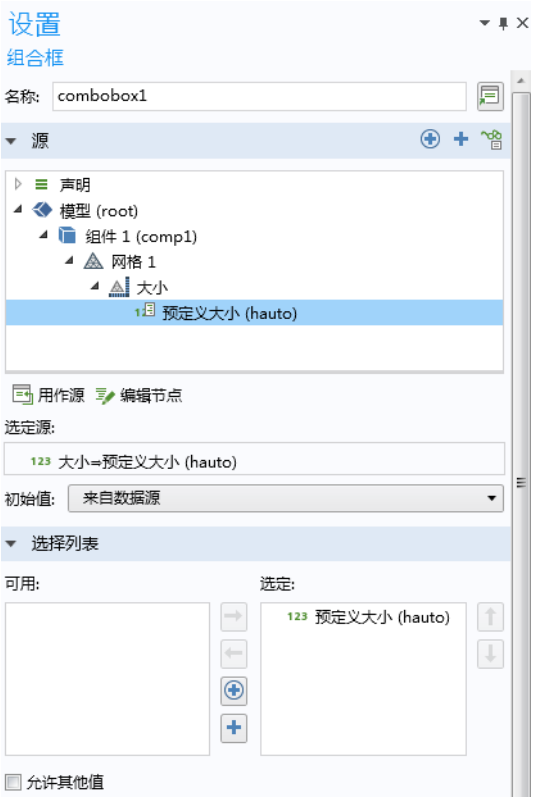
选中列表左侧的绿色复选框，将其设为 “App 开发器”中组合框的源。然后，当您返回 “App 开发器”时，将会发现网格大小的选择列表现在显示为新组合框**设置**中潜在的**源**。

您可以通过两种方式来插入组合框对象：

- 从功能区的**插入对象**菜单中选择**组合框**。在组合框的**设置**窗口中，从**源**栏选择**预定义大小 (hauto)** 节点，然后单击**用作源**按钮。
- 在**编辑器工具**窗口中，从**网格 > 大小**节点下选择**预定义大小 (hauto)** 节点。然后右键单击并选择**输入**，如下图所示。

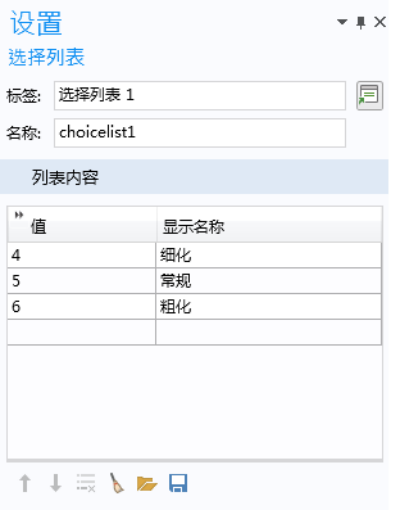


该组合框对应的**设置**窗口如下图所示。



将**初始值**改为**来自数据源**，可以确保模型的单元大小设置 （本例中为**常规**）用作 App 的默认单元大小。现在，“模型开发器”中的选择列表**预定义大小 (hauto)** 已选为 “App 开发器”中组合框的选择列表。由于该选择列表引用自“模型开发器”，因此它不在 App 树的**声明**节点下显示为选择列表。鉴于此，如果您希望一个列表受到更多的选择限制，您无法直接编辑它。您必须从组合

框的**源**中移除预定义的列表，然后创建新的选择列表，方法是在**声明**节点下声明一个选择列表。例如，您可以创建具有三个条目的选择列表，如下图所示。



要查看模型中的**单元大小**列表使用了哪些值，可以使用**录制方法**并将值从**常规**改为**细化**，接着改为**粗化**，然后再改回**常规**。单击**停止录制**并在自动生成的代码中读取值。**单元大小**属性的名称为 hauto，**细化**、**常规**和**粗化**的值分别为 4、5 和 6，这些信息可通过以下自动生成的代码行来获取。

```
with(model.mesh("mesh1").feature("size"));
  set("hauto", "4");
  set("hauto", "6");
  set("hauto", "5");
endwith();
```

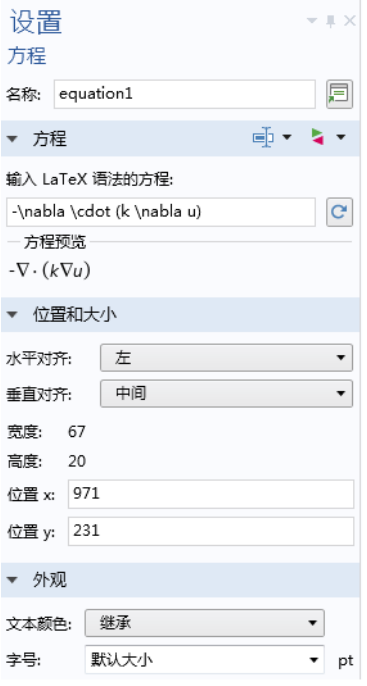
hauto 属性也可以取非整数值。有关**单元大小**的更多信息，请参见第 96 页的“输入框的数据访问”。

使用单位集代替选择列表

如果要将组合框用于更改单位，则可以使用**单位集**来代替**选择列表**（您仍可以在组合框**设置**窗口的**选择列表**栏中选择该选项）。

方程

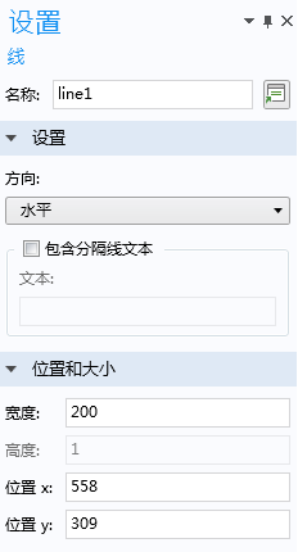
方程对象可以显示 LaTeX 方程，您可以通过在**输入 LaTeX 语法的方程**框中输入方程来进行操作。



在您离开文本框后，会显示 LaTeX 语法表示的方程预览。

线

使用**线**表单对象可以在表单中添加水平或垂直线，用于按组分隔表单对象等。对于水平线选项，您还可以添加可在线内显示的文本。



网页

网页对象可以将网页内容作为用户界面的一部分显示。

您可以通过四种不同的方式从**源**列表中指定页面来源：

- 使用默认选项**页面**在列表下方的文本区域中输入 HTML 代码，将其置于 `<html>` 与 `</html>` 开始和结束标记之间。
- 使用 **URL** 选项将其链接到 Internet 上的网页。
- 使用**文件**选项指向一个包含 HTML 代码的本地文件资源。在**文件**编辑框中键入文件名，或单击**浏览**以查找本地文件系统中的文件。
- 使用**报告**选项以嵌入 HTML 报告。“浏览器预览”对此选项不可用。

图像

使用**图像**表单对象可以在表单中添加图像。图像对象与图形对象不同，前者不能进行交互。您可以访问下拉列表，从一个库图像中选择图像文件，或单击**将图像添加到库并在此使用**按钮从本地文件系统中选择文件。下图显示图像对象的**设置**窗口，该对象引用**库**节点中定义的图像 `cube_large.png`。

如果您从文件系统选择图像文件，该文件将嵌入 App，并添加到**库**节点下的**图像**列表中。

当您更改图像的 x 和 y 位置时，宽度和高度设置由图像文件确定。

> 您可以使用 Ctrl+V 将图像从剪贴板粘贴到表单窗口。例如，您可以从 PowerPoint® 幻灯片演示软件复制粘贴图像。此类图像会自动添加到**图像**库并嵌入 App。已粘贴图像的名称会自动设为：`pasted_image_1.png`、`pasted_image_2.png` 等。

视频

视频对象用于在表单中嵌入视频文件。支持的视频文件格式为 MP4 (`.mp4`)、OGV (`.ogv`) 和 WebM (`.webm`)。但并非所有平台都支持所有的视频文件格式。通过连接到 COMSOL Server 的 Web 浏览器运行 App 时，支持的格式取决于所用的 Web 浏览器，并且同一浏览器的不同版本支持的格式也可能不同。当您使用 COMSOL Client 和 COMSOL Multiphysics 运行 App 时，计算机上安装的 Internet Explorer 版本用作显示视频对象的软件组件。

视频对象在添加到表单后，它在“表单编辑器”中表示为图像，如下图所示。

下图显示**视频**对象的**设置**窗口。

可用的设置包括：

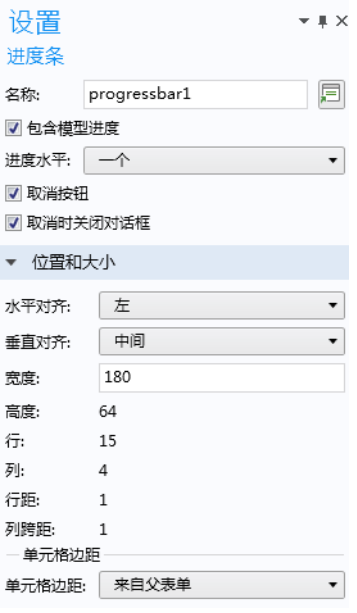
- **显示视频控件**
- **自动开始**
- **重复**
- **初始静音**

显示视频控件选项可启用视频控件，例如 “播放” 和 “停止”。

当您希望在最初关闭声音的情况下播放视频时，可以使用**初始静音**选项。例如，如果视频设为自动开始，此选项可供用户选择是否打开声音。用户可以从视频控件启用声音 （如果已选中**显示视频控件**复选框），也可以右键单击视频播放器进行操作。

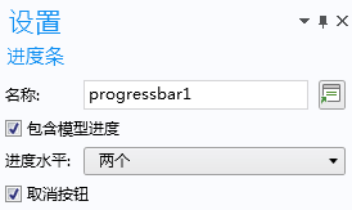
进度条

进度条对象基于由方法更新的值来显示定制的进度条，或显示一组进度条。使用进度条可以为 App 的剩余运行时间提供反馈。下图显示具有一个进度级别的进度条对象的**设置**窗口。



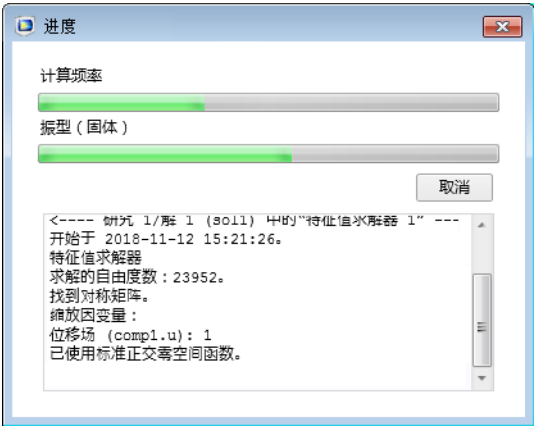
请注意， App 的状态栏中可见的内置进度条由**主窗口**节点的**设置**窗口控制。默认情况下，内置进度条显示内置 COMSOL Multiphysics 核心算法的进度，例如几何操作、网格划分及求解。您可以使用 `setProgress` 方法来定制内置进度条中显示的信息。有关更多信息，请参见第 324 页的 “进度方法” 和 Application Programming Guide。

下图显示具有两个进度级别的进度条对象的**设置**窗口。

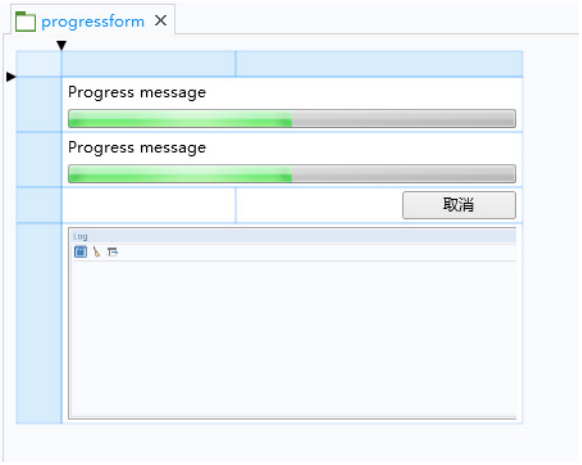


在本例中，进度条对象是表单 progressform 的一部分，用于显示两级进度条和消息日志。

下图显示正在运行的 App 对应的进度对话框。



下图显示表单 progressform。



以下代码段显示用于更新进度条和消息日志的典型内置方法。

```
// show progress dialog box:
dialog("progressform");
setProgressBar("/progressform/progress1", 0, "Computing prong length.");

// code for iterations goes here:
lastProgress = 20;
// ...

// update message log:
message("Iteration Number: " + k);
message("Frequency: " + Math.round(fq*100)/100.00);
message("Length: " + Math.round(L1*100)/100.00);

// update progress bar:
setProgressInterval("Computing frequency", lastProgress,
k*100/MAXITERATIONS);
// more code goes here:
// ...

// finished iterating:
setProgressBar("/progressform/progress1", 100);
closeDialog("progressform");
```

在上述示例中，用于更新两级进度条的核心功能取决于以下调用代码：

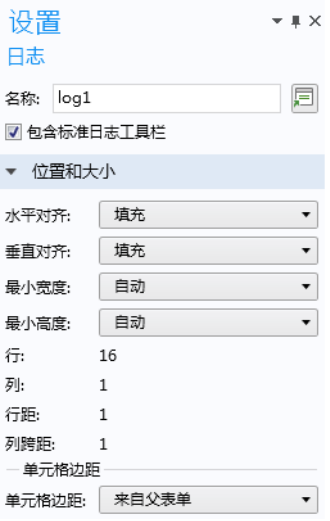
```
setProgressInterval("Computing frequency", lastProgress,
k*100/MAXITERATIONS)。
```

有关内置方法及其语法的详细信息，请参阅第 324 页的 “进度方法” 和 Application Programming Guide。

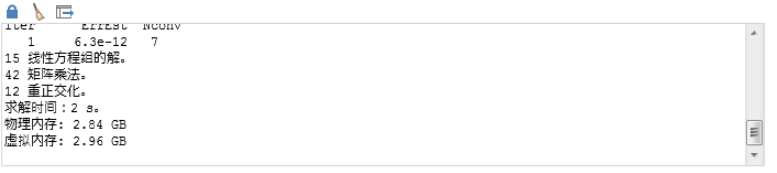
日志

日志表单对象可以添加一个日志窗口，用于显示内置 COMSOL Multiphysics 核心算法中生成的消息，例如几何操作、网格划分及求解。

默认情况下，**包含标准日志工具栏**复选框处于选中状态。此时， App 中会包含您在 COMSOL Desktop 中看到的**日志**窗口中的工具栏。



下图显示一个包含日志窗口的 App 用户界面的一部分。

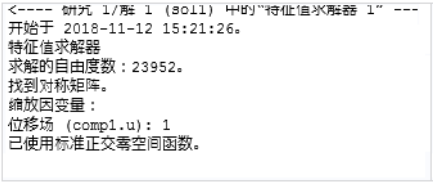


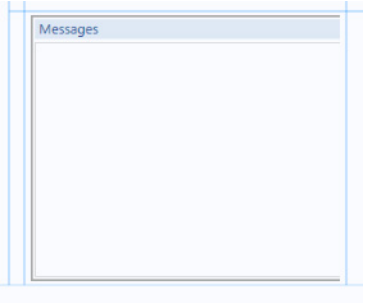
消息日志对象可以添加一个窗口，您可以在其中显示消息，以通知用户 App 执行的相关操作。您可以通过以下语法使用内置的 `message` 方法来实现此特征：`message(String message)`。另请参见第 319 页的 “GUI 相关方法”。

您也可以显示变量的值，例如：`message(double xcoordinate)`。

默认情况下，**包含标准消息日志工具栏**复选框处于选中状态。此时， App 中会包含您在 COMSOL Desktop 中看到的**消息**窗口中的工具栏。默认情况下，**显示 COMSOL 消息**复选框处于选中状态，以启用内置 COMSOL Multiphysics 核心算法中生成的消息，例如几何操作、网格划分及求解。清除该复选框以仅显示来自 App 本身的消息。您可以通过选中**在消息中添加时间戳**复选框，在消息中包含时间戳。

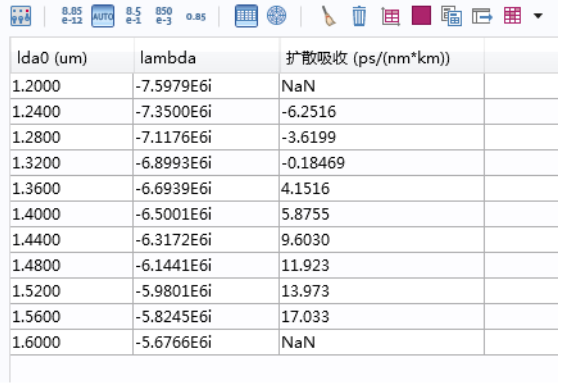
下图显示定制的消息窗口，其中包含来自方法的收敛信息 （左）以及对应的**消息日志**表单对象 （右）。





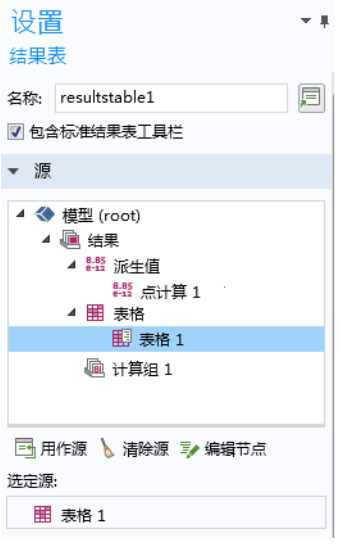
结果表

结果表对象用于在表格中显示数值结果。



lda0 (um)	lambda	扩散吸收 (ps/(nm*km))
1.2000	-7.5979E6i	NaN
1.2400	-7.3500E6i	-6.2516
1.2800	-7.1176E6i	-3.6199
1.3200	-6.8993E6i	-0.18469
1.3600	-6.6939E6i	4.1516
1.4000	-6.5001E6i	5.8755
1.4400	-6.3172E6i	9.6030
1.4800	-6.1441E6i	11.923
1.5200	-5.9801E6i	13.973
1.5600	-5.8245E6i	17.033
1.6000	-5.6766E6i	NaN

结果表数据的源取自**结果**节点，并可以作为**派生值**的子节点、**表格**或**计算组**。在下图中，**表格**节点用作源 （在树中选择此选项，然后单击**用作源**。）



结果表工具栏

默认情况下，**包含标准结果表工具栏**复选框处于选中状态。此时，会包含一个工具栏，显示以下按钮：

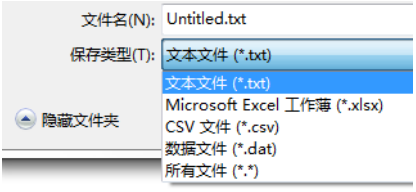
- **全精度**
- **自动记数法**
- **科学记数法**
- **工程记数法**
- **十进制表示法**
- **矩形复数**
- **极坐标形式复数**
- **将表格和标题复制到剪贴板**
- **导出**

导出按钮用于导出为以下文件格式：

- 文本文件 (`.txt`)
- Microsoft® Excel® 工作簿 (`.xlsx`)
  - 需要 LiveLink™ *for* Excel®

- CSV 文件 (`.csv`)
- 数据文件 (`.dat`)

如下图所示。



通过方法控制结果表

您可以使用内置方法 `useResultsTable()` 来更改要在特定 “结果表” 表单对象中显示的表格。有关此内置方法的更多信息，请参见第 319 页的 “GUI 相关方法”。

表单

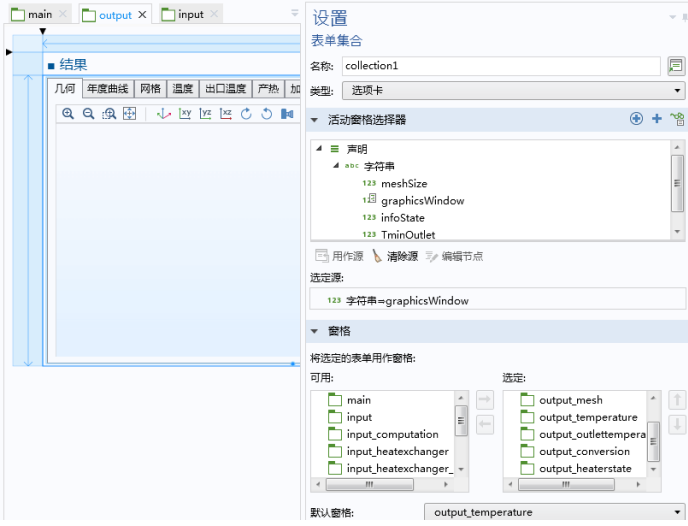
表单类型的表单对象用于在一个或多个子表单中组织主表单。要嵌入子表单，您可以从子表单**设置**窗口的**表单**引用中选择要链接的表单，从而为这一子表单创建链接。在下图显示的示例中，表单 `main` 的一个单元格已链接到表单 `input`。

下图显示引用的表单 input。

如果您在栅格布局模式下，则可以使用功能区中的**提取子表单**按钮来快速创建子表单。请参见第 113 页的 “提取子表单”。

表单集合

表单集合对象由多个表单或窗格组成，并显示在一个主表单中。在本例中，一个主窗口包含四个表单，以选项卡的形式显示。



系统提供四个不同的布局选项。可以从**类型**列表中进行选择：

- **选项卡**，这是默认设置，使用选项卡式窗格显示表单。
- **列表**，可在表单窗格左侧显示列表，您可以从中选择要显示的表单。
- **栏**，在单独的栏中显示每个表单。
- **平铺或分页**，通过两种方式中的一种来显示表单，具体取决于布尔变量的值。有关更多信息，请参见本节后续描述。

在**窗格**栏的**将选定的表单用作窗格**列表中，每个表单都表示一个窗格。它们将按列表中出现的顺序显示在 App 中。您可以单击右侧的**上移**和**下移**按钮来更改顺序。

您可以在**活动窗格选择器**栏将选项卡 （或列表条目）链接到字符串变量，以控制其活动状态。

该字符串变量需要等于表单集合中的一个表单名称，如上例中的 `temperature` 或 `conversion`。否则，它将被忽略。

在上述示例中，对于要在某个时刻运行的方法 （如按钮方法），如果您在其中更改窗格选择器 `pane` 的值，则会激活具有新值的窗格，如下例所示。

```
pane="conversion"; /* Activate the conversion pane on completion of this
method */
```

对于**类型**设为**栏**的表单集合，**活动窗格选择器**不起作用。使用**活动窗格选择器**是一个可选操作，仅当您希望通过某种方法而非单击选项卡来控制活动选项卡时才需要这样做。要移除用作**活动窗格选择器**的字符串变量，请单击树结构下方的**清除源**工具栏按钮。

平铺或分页选项通过两种方式中的一种来显示表单，具体取决于在 “设置” 窗口顶部的**平铺或分页**栏中用作源的布尔变量的值。

分页模式与**类型**设为**选项卡**的表单集合相同。在平铺模式下，所有表单同时显示在栅格中。平铺模式的布局可以通过**平铺模式设置**子栏中的设置进行控制。

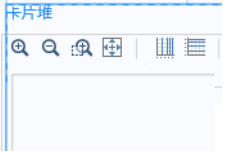
卡片堆

卡片堆是包含卡片的表单对象。**卡片**是另一种类型的表单对象，仅用于卡片堆上下文。在卡片堆的卡片之间进行翻转，可每次显示一个卡片。您可以将卡片堆与控制卡片显示的数据源关联起来。每个卡片都指定一个值，用来与卡片堆的数据源进行比较。卡片堆会显示具有匹配值的第一个卡片。如果没有匹配的卡片，则不显示任何内容。

使用卡片堆在图形对象之间翻转

我们以这样一个 App 为例，其中显示给用户的图形取决于标量变量的值。此变量可以在用户执行某些操作 （如单击单选按钮）时发生更改。该变量也可以随计算得到的值而发生更改；例如，模型树中**全局计算**节点的值。

下图显示 “表单编辑器”中的卡片堆对象。



在本例中，卡片堆包含带图形对象的卡片。

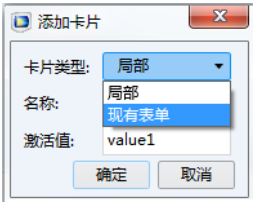
下图显示一个包含五个卡片的卡片堆**设置**窗口，其中的字符串变量 `display` 作为**活动卡片选择器**。

通过在**卡片**栏中单击卡片表中的行，然后单击表格下方的一个工具栏按钮，您可以对卡片执行以下操作：

- **删除**
- **编辑**
- **添加卡片**
- **生成副本**

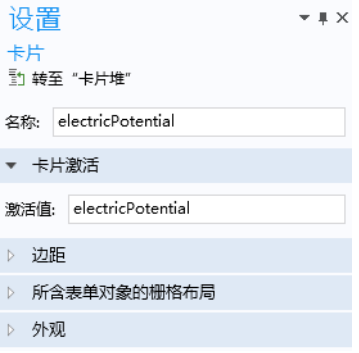
表格中的每一行都包含卡片名称，位于**卡片**列中，其关联的激活值则位于**激活值**列中。卡片堆基于卡片的激活值来决定要显示的卡片。在本例中，激活值为字符串 `geometry`、 `velocity`、 `particle1` 等。

单击**添加卡片**按钮可显示以下对话框。

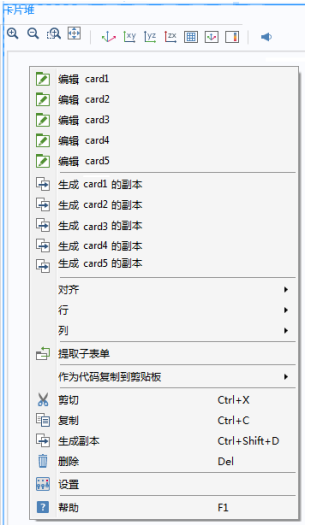


默认情况下，**卡片类型**设为**局部**，表示该卡片是在它所含的卡片堆对象中局部定义的。如果**卡片类型**设为**现有表单**，则可以直接选择一个现有表单。您可以通过单击表单节点或单击对应卡片堆**设置**窗口的**卡片**栏中的**编辑**按钮，直接从“表单编辑器”访问**现有表单**的设置。

下图显示您在**卡片**栏的表格中单击**编辑**后显示的**卡片设置**窗口。

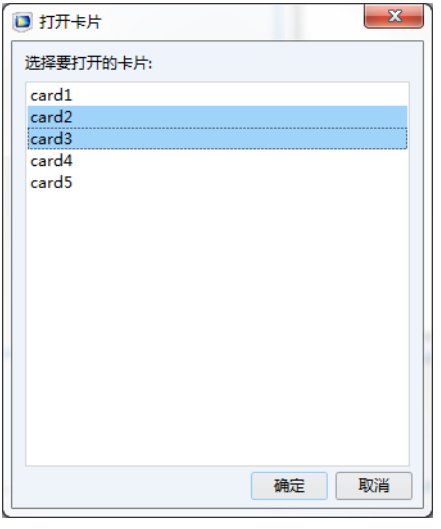


要访问局部定义的卡片，可以右键单击表单窗口中的卡片堆，从中选择不同的卡片，如下图所示。

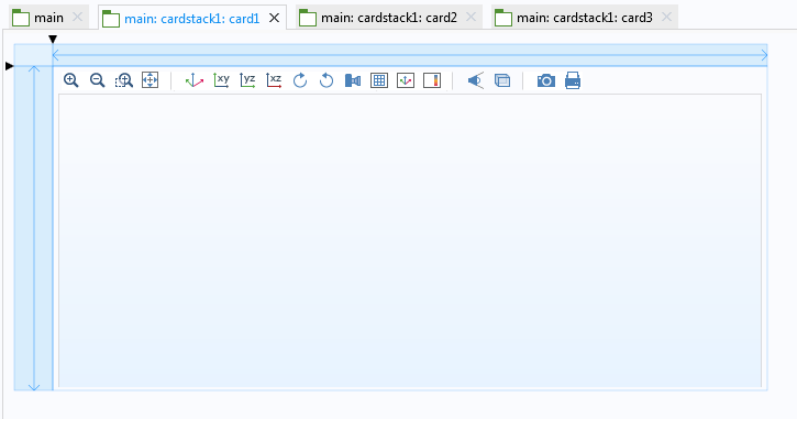


您还可以使用此菜单为卡片生成副本。

要编辑卡片，您也可以使用 Alt+ 单击操作，打开一个对话框，供您一次选择多个卡片。



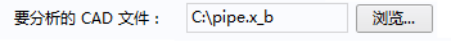
下图显示含图形表单对象的 card1。



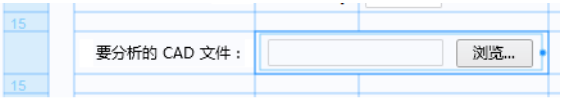
文件导入

文件导入对象用于显示文件浏览器，您可以通过关联的输入框浏览文件或输入文件路径和名称。当某个文件尚未加载到 App 时， App 用户可以在运行时通过该对象启用文件导入功能。

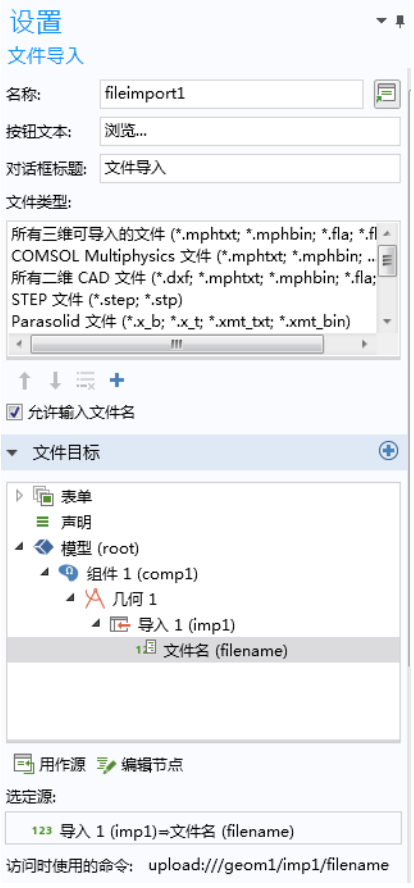
我们以这样一个 App 为例，您可以在 App 运行时选择 CAD 文件并将其导入，如下图所示。



对应的**文件导入**对象如下图所示。



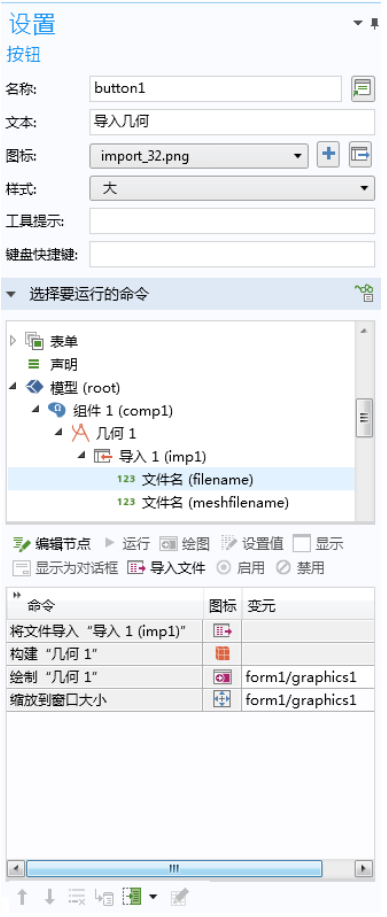
文件导入对象的**设置**窗口包含**文件目标**栏。您可以在该栏中选择支持输入文件名的任意树节点。如下图所示，其中几何**导入**节点的**文件名**处于选中状态。



在此 App 中，**文件类型**表指定只允许导入 CAD 文件。您可以通过单击**文件类型**列表下的**添加**和**删除**按钮来进一步控制允许的**文件类型**。单击**添加**按钮可以显示如下对话框：

使用文件导入对象的备选方法

如果不需要文件路径和名称的输入框，系统还提供其他文件导入方法，供用户在文件浏览器中选取文件。例如，您可以使用菜单、功能区、工具栏项或按钮打开文件浏览器。下图显示用于导入 CAD 文件的按钮的**设置**窗口。



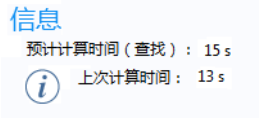
文件导入对象还可以引用**文件**声明。有关更多信息，请参见第 149 页的 "文件"。有关常规文件处理的更多信息，请参见第 290 页的"附录 C - 文件处理和文件方案语法"。

与**导入文件**命令对应的内置方法是 `importFile`。例如，您可以使用以下代码导入图像：

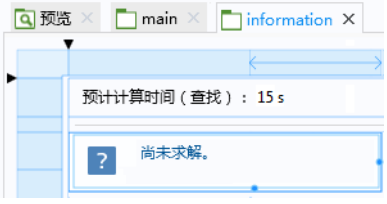
```
success=importFile("file1",new
String[]{"ALL_IMPORTABLE_IMAGES","PNG","JPEG","BMP","GIF"});
```

信息卡片堆

信息卡片堆对象是一种专用类型的**卡片堆**对象，用于显示用户在 App 中给定的输入与解之间的关系信息。下图显示一个正在运行的 App 的一部分，其中同时使用了信息卡片堆和预计计算时间相关信息。



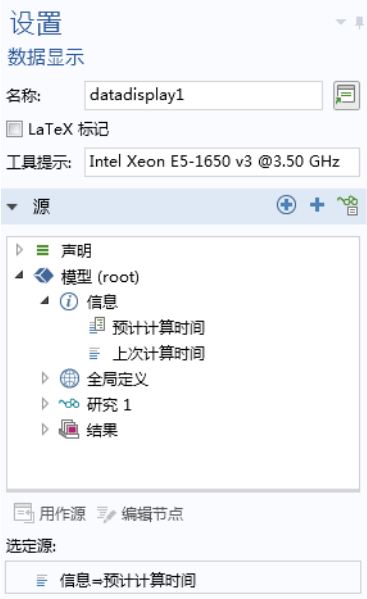
对应的表单对象如下所示：



在下图显示的**设置**窗口中，字符串变量 `solution_state` 用作源。

信息卡片堆与**卡片堆**对象有相似之处，但每个**信息卡片**都包含图标和文本。在上图中，字符串变量值 `nosolution`、`inputchanged` 和 `solutionexists` 用于控制要显示的信息卡片。

在本例中，信息卡片堆附有数据显示对象，其中**预计计算时间**的模型树信息节点用作源。下图显示相应的**设置**窗口。

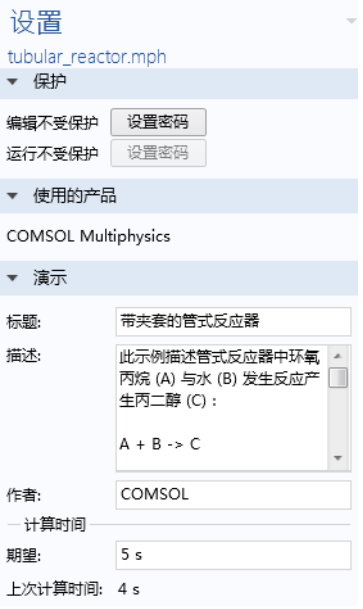


请注意，仅当您使用 “App 开发器”时才显示模型树中的信息节点。它们会在适用的情况下显示在表单对象**设置**窗口的**源**栏中。

您还可以在每个研究下找到**上次计算时间**信息节点。直接位于**模型**节点下的**上次计算**信息节点对应于上次计算的研究所用的计算时间。

信息节点可以用作输入框对象、文本对象以及数据显示对象的源。对于输入框和文本对象，需要清除**可编辑**复选框才能访问信息节点。

预计计算时间从 App 树的根节点获取数据，如下所示。



如果计算时间主要花费在方法中，例如在重复调用同一研究时，您可以使用内置方法 `timeStamp` 与 `setLastComputationTime` 来手动测量计算时间。有关更多信息，请参见第 324 页的 “日期和时间方法”。

数组输入

数组输入对象包含一个输入表格，可用于输入数组或矢量值输入数据。该对象支持字符串数组作为数据源。您可以添加可选的标签、符号和单位。

将数组输入对象用于三维点坐标输入

我们以这样一个 App 为例，用户在其中为需要计算应力的点输入三维坐标。下图显示一个包含数组输入、按钮、文本标签和数据显示对象的 App 的屏幕截图。

下图显示数组输入对象的**设置**窗口。

数组输入表单对象使用名为 `samplecoords` 的**源**，它是**双精度型**的**一维数组**。此数组是在创建**数组输入**对象之前，通过声明**一维双精度**数组创建的，其**设置**如下。

在数组输入对象的**设置**窗口：

- 在**长度**编辑框中，为数组长度输入正整数。默认值为 3。
- 从**将矢量显示为**列表中，选择**表格** （默认设置）将数组分量显示为表格，或选择**分量**将每个数组分量显示为单个带标签的输入框。
- 在**值**表中，为数组中的分量输入初始值。
- **布局选项**栏提供的设置用于为数组输入添加可选的标签和单位。

在本例中，当用户单击带**计算点的应力**标签的按钮时，会运行以下方法：

```
with(model.result().dataset("cpt1"));
  set("pointx", samplecoords[0]);
  set("pointy", samplecoords[1]);
  set("pointz", samplecoords[2]);
endwith();
```

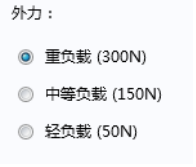
其中，值 `pointx`、 `pointy` 和 `pointz` 随后将用作应力计算中的坐标。

单选按钮

单选按钮对象包含固定数量的选项，您可以从中选择一个。当您只需要少量选项时，该对象最为有用。

使用单选按钮选择载荷

我们以这样一个 App 为例，其中包含三个预定义的载荷，用户可以从中选择一个，如下图所示。



下图显示对应的**设置**窗口，其中全局参数 F 用作源。

方向可以设为**垂直** （默认设置）或**水平**。

在**初始值**列表中，选择单选按钮的初始选择方式。其中包含以下选项：**来自数据源**、**第一个许用值** （默认设置）和**定制值**。对于**定制值**选项，需要从选择列表给出的许用值列表中进行选择。

您可以在**选择列表**栏中添加选择列表，为单选按钮对象提供许用值，其中的每个有效值都表示一个单选按钮。

单选按钮名称取自其关联选择列表的**显示名称**列。下图显示本例中使用的选择列表。

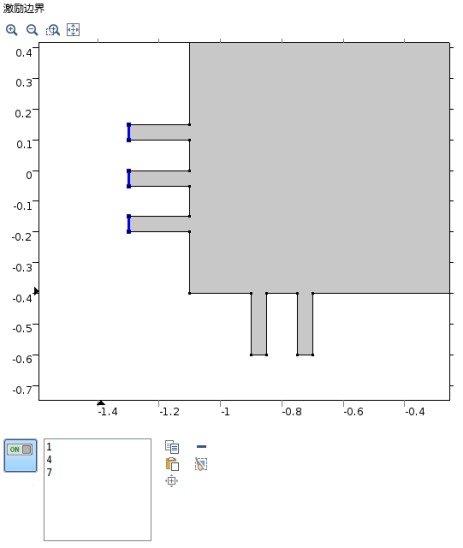
使用单位集代替选择列表

如果要将单选按钮用于更改单位，则可以使用**单位集**来代替**选择列表** （您仍可以在单选按钮对象**设置**窗口的**选择列表**栏中选择该选项）。

选择输入

在 “App 开发器”中，您可以通过**选择输入**对象或**图形**对象来允许 App 用户交互更改属于**显式**选择的实体。有关选择的更多信息，请参见第 80 页的 “选

择”。在下面的示例中，嵌入模型包含通过**显式**选择定义的边界条件。同时使用**选择输入**对象和**图形**对象，使用户能够选择要由入射波激励的边界。



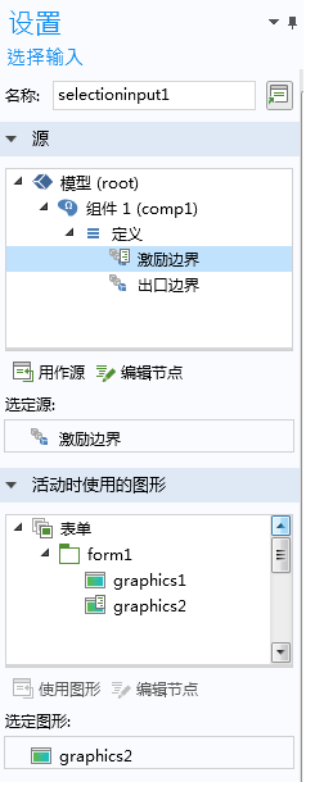
用户可以通过以下方式来选择边界：直接在对应于**图形**对象的图形窗口中单击，或者在对应于**选择输入**对象的边界编号列表中添加几何实体编号。

为了能通过单击边界直接将其选中，您可以将图形对象链接到用于为边界分组的显式**选择**，如下图所示。选中显式选择并单击**用作源**。

在下图中，有两个显式选择：**激励边界**和**出口边界**，并且图形对象 graphics2 已链接到**激励边界**选择。

当图形对象以这种方式直接链接到显式选择时，该图形对象会显示几何，用户可以通过单击边界与之进行交互。这些边界随后会添加到相应的显式选择中，或从中移除。

为了能按编号进行选择，您可以将选择输入对象链接到显式选择，如下图所示。



在选择输入对象中，您可以复制、粘贴、移除、清除以及放大选择。

您可以选择在没有任何选择输入对象的情况下，使用图形对象作为选择的源。您还可以将图形对象和选择输入对象链接到同一个显式选择。

文本

文本对象是包含默认文本的文本框，文本内容来自字符串变量或**信息**节点。下图显示文本对象的**设置**窗口。

从**源**栏的树结构中选择字符串变量或**信息**节点，然后单击**用作源**。在**值**编辑框中输入初始文本。默认情况下，**初始值**文本来自该编辑框。要使用字符串变量作为**初始值**文本，可将**初始值**设置改为**来自数据源**。

默认情况下，**可编辑**复选框处于清除状态。如果将其选中，文本对象便可以用来在运行的 App 中键入注释等。如果文本被用户更改，它会存储到用作数据源的字符串变量中，而不论**初始值**如何设置。

默认情况下，**自动换行**复选框处于选中状态。清除此复选框后，可以禁用文本换行。如果文本未显示完整，将出现一个滚动条。

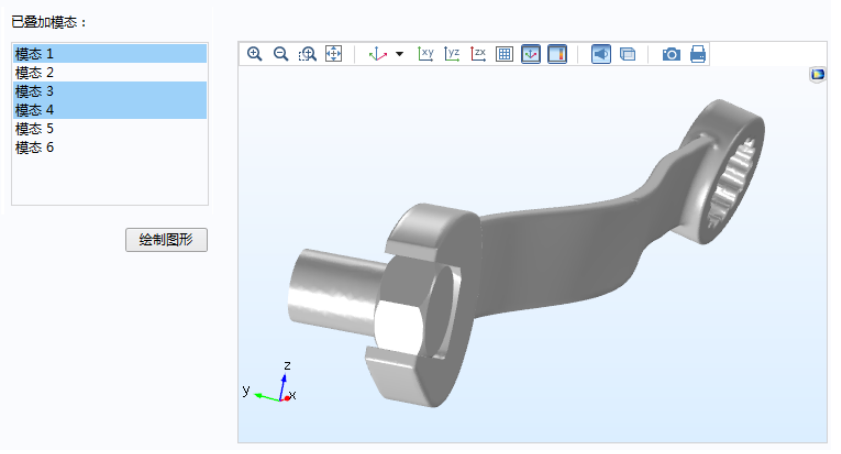
有关**信息**节点的更多信息，请参见第 93 页的 “数据显示”。

列表框

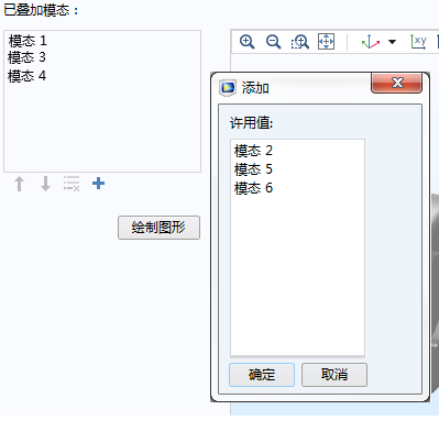
列表框对象与单选按钮对象类似，不同之处在于，该对象支持同时选择多个选项。

使用列表框叠加振动模态

我们以这样一个 App 为例，您可以从其中的列表框选择某个机械零件的前六个振动模态，将它们进行叠加并生成可视化效果，如下图所示。



下图显示了一种备选方式，列表也可以显示为对话框。



本例中的列表框**设置**窗口如下图所示。

值选择自列表可供您在两个备选项之间选择一个来显示列表：**列表框**或**对话框**。您可以使用任何标量或数组声明作为源。从树结构中选择一项，然后单击**用作源**。如果您使用字符串数组作为源，则可以在运行的 App 中使用 Shift+ 单击或

Ctrl+ 单击操作从列表中选择多个项。对于其他源，您只能从列表中选择一个值。本例使用一维字符串数组 svar1D。对应的**设置**窗口如下所示。

在**选择列表**栏中，您可以添加选择列表，为列表框提供许用值。下图显示本例中使用的选择列表。

振动模态 1-6 对应于普通的刚体模态，不是该 App 要分析的范围，因此**值**列中的项从 7 开始。选择列表支持您仅将**显示名称**列中的字符串设为可见，从而对

用户隐藏模型中的实际模态值。前几个非刚体模态名为“模态 1”、“模态 2”等。

下面的方法使用 COMSOL Multiphysics 运算符 `with()` 来生成叠加模态的可视化效果。本例进行了一定程度的简化，其中忽略了幅值与相位对模态的影响。

```
String withstru="0";
String withstrv="0";
String withstrw="0";
for(int i=0;i<svar1D.length;i++){
  withstru=withstru + "+" + "with(" + svar1D[i] + ",u)";
  withstrv=withstrv + "+" + "with(" + svar1D[i] + ",v)";
  withstrw=withstrw + "+" + "with(" + svar1D[i] + ",w)";
}

with(model.result("pg7").feature("surf1").feature("def"));
  setIndex("expr", withstru, 0);
  setIndex("expr", withstrv, 1);
  setIndex("expr", withstrw, 2);
endwith();
useGraphics(model.result("pg7"),"/form1/graphics8");
zoomExtents("/form1/graphics8");
```

假设用户使用列表框选择了模态 1、 3 和 5，该方法会创建一个表达式 `with(1,u)+with(3,u)+with(5,u)`。这个表达式稍后会在位移图中用来计算 x 位移（因变量 `u`）。该方法还会以同样的方式分别为与 y 和 z 位移关联的变量 `v` 和 `w` 自动创建表达式。请注意，在上例的结果中使用的命令 `with()` 与用来缩短语法的内置 `with()` 命令（如第 327 页的“With、Get 和 Set 方法”中所述）不同。

使用单位集代替选择列表

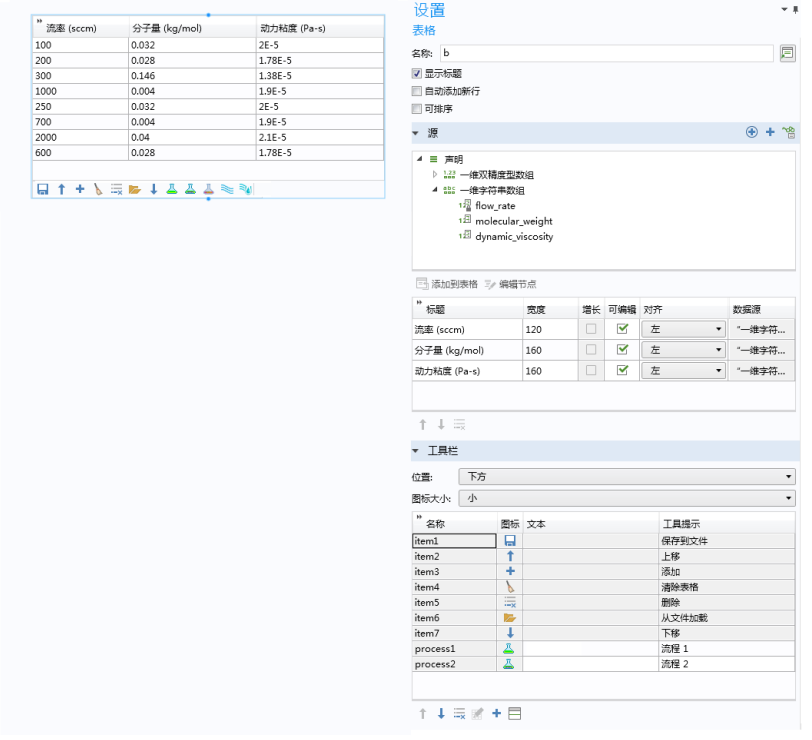
如果要将列表框用于更改单位，则可以使用**单位集**来代替**选择列表**（您仍可以在列表框**设置**窗口的**选择列表**栏中选择该选项）。

表格对象表示包含行和列的表格，可用于定义输入或输出。下图显示一个正在运行的 App 示例，其中使用表格对象来接受三列输入。

流率和流体属性：

流率 (sccm)	分子量 (kg/mol)	动力粘度 (Pa-s)
100	0.032	2E-5
200	0.028	1.78E-5
300	0.146	1.38E-5
1000	0.004	1.9E-5
250	0.032	2E-5
700	0.004	1.9E-5
2000	0.04	2.1E-5
600	0.028	1.78E-5

下图显示对应的表单对象及其**设置**窗口。



在本例中，数据源引用三个一维字符串数组。您可以选择任意类型的数组作为源，然后单击**用作源**。

您可以通过三个复选框来控制表格的整体外观：

- **显示标题**
- **自动添加新行**
- **可排序**

自动添加新行复选框用于确保在用户填写表格时，始终可以添加空行。如果用作表格源的所有一维字符串数组在其声明**设置**窗口的**新元素值**中都包含非空值，则此功能会被停用。在这种情况下，只能通过在关联的表格工具栏中单击**添加**按钮 （如果此按钮可用）来添加新行。

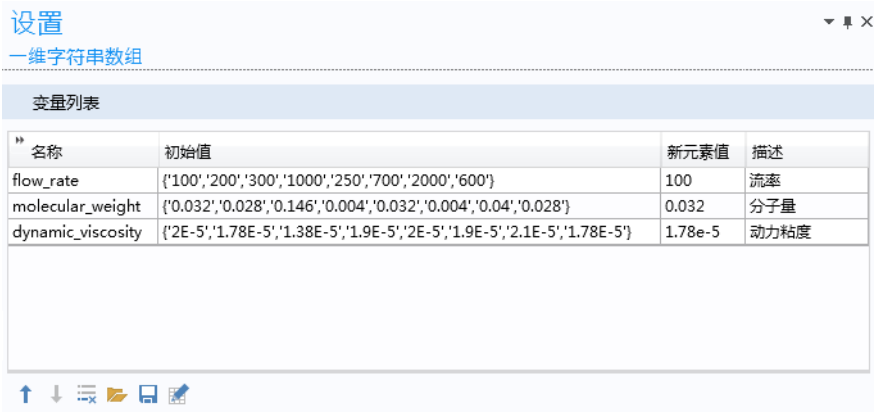
可排序复选框支持相对于特定的列进行排序，通过单击相应的列标题即可完成。

源栏中的表格包含以下六列：

- **标题**
- **宽度**
- **增长**
- **可编辑**
- **对齐**
- **数据源**

该表格中的每一行都定义表格对象中的一列。**增长**选项支持单个列在表单大小调整时进行增长。此选项仅适用于栅格模式，并且表格的**水平对齐**需要设为**填充**。

在本例中，字符串数组用来定义三列对应行的初始值，如下图所示：



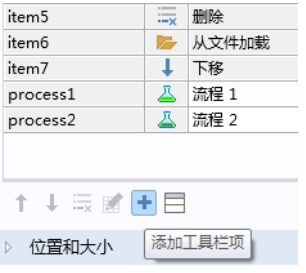
工具栏

在这个栏中，您可以选择使用哪些工具栏项 （按钮）来控制表格内容。**位置**列表用于定义工具栏相对于表格的位置，并提供以下选项：

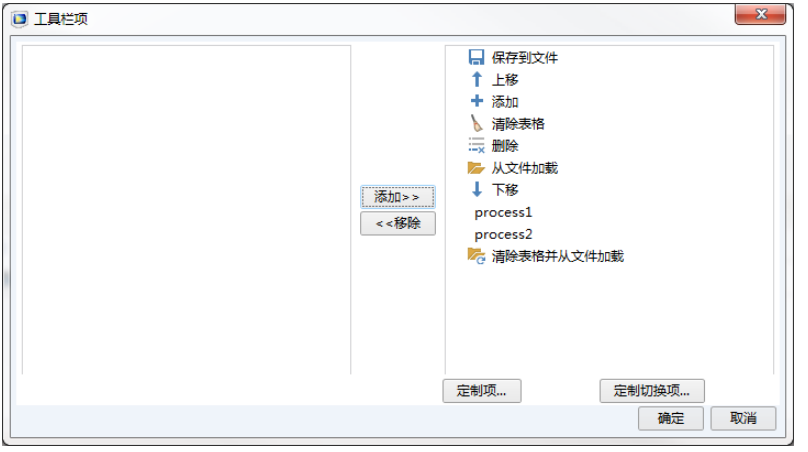
- **下方**
- **上方**
- **左**
- **右**

图标大小设置可供您选择**小**图标或**大**图标。

要在工具栏中添加项，可以在表格下方单击**添加工具栏项**按钮。



此时会显示以下对话框。

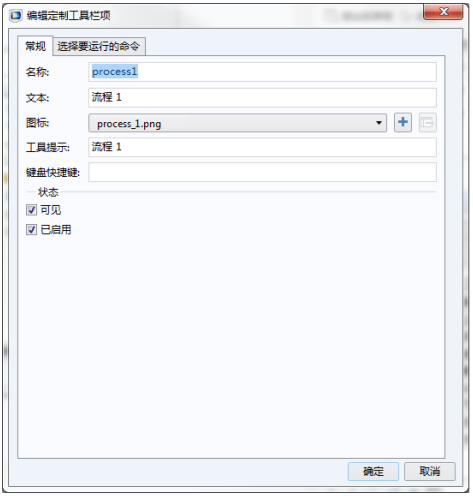


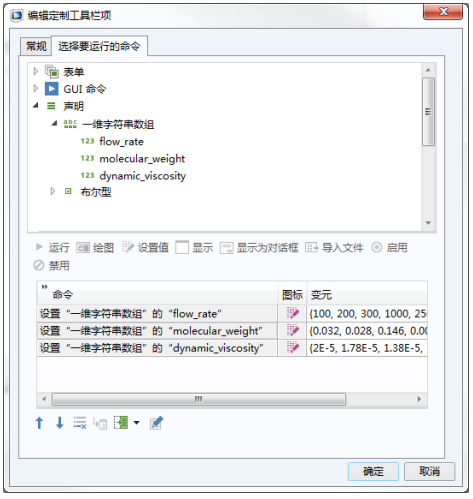
您可以添加以下项：

- **保存到文件**
- **上移**
- **下移**
- **添加**
- **删除**
- **清除表格**
- **清除表格并从文件加载**
- **从文件加载**

此外，您也可以通过在**工具栏项**对话框中单击**定制项**或**定制切换项**，来添加定制项。下图显示用来定义定制按钮的**编辑定制工具栏项**对话框。对于常规项，

该对话框包含两个选项卡；对于切换项，则包含三个选项卡。在本例中，按钮 **Process 1** 用于设置特定过程的默认值。

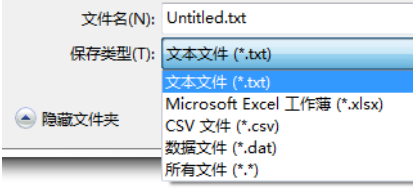




选择要运行的命令选项卡与菜单、功能区、工具栏项及按钮的对应选项卡类似。**从文件加载**和**保存到文件**按钮所执行的加载和保存操作适用于以下文件格式：

- 文本文件 (`.txt`)
- Microsoft® Excel® 工作簿 (`.xlsx`)
  - 需要 LiveLink™ *for* Excel®
- CSV 文件 (`.csv`)
- 数据文件 (`.dat`)

如下图所示。



CSV 文件支持的分隔符为逗号、分号和制表符， DAT 与 TXT 文件支持的分隔符为空格和制表符。

滑块

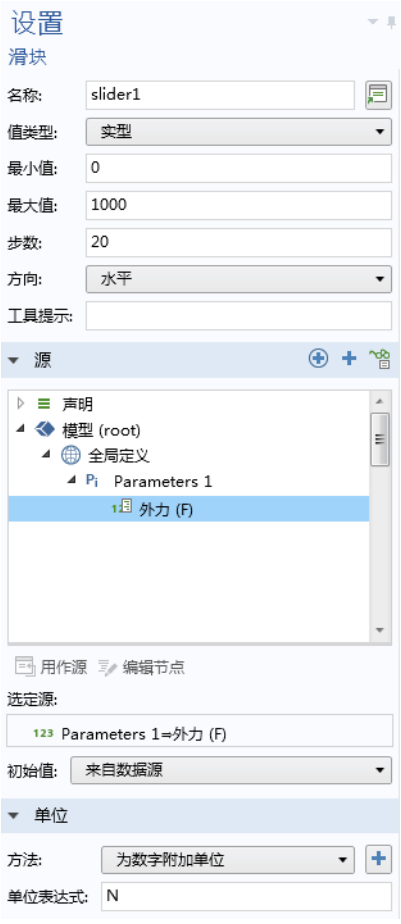
滑块是一种表单对象，可通过使用滑块控件来选择数值输入。

使用滑块更改结构载荷的大小

我们以这样一个 App 为例，其中载荷的大小可通过滑块控件进行更改，如下图所示。

在本例中，滑块附带一个输入框，用于显示选定的值。

滑块的**设置**窗口如下图所示。



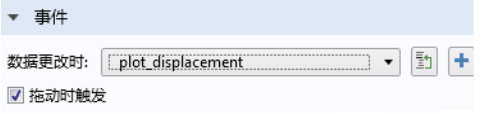
在本例中，滑块使用全局参数 F 作为源。您可以选择任何参数、变量或声明的标量变量作为源。从 App 树中选择一项，然后单击**用作源**。

从**值类型**列表中选择**整型**或**实型** （默认设置），具体取决于滑块数据源的数据类型。

您可以通过定义滑块的**最大值**、**最小值**和**步数**来确定数据源的值范围。**方向**选项包括**水平**或**垂直**。您还可以设置**工具提示**，当您将光标悬停在滑块上方时显示。**为数字附加单位**选项可供您将单位与滑块关联起来。在一个数字作为值传递给源变量之前，单位会使用标准括号表示法附加到该数字，例如 [N]。在上例中，输入框和滑块均激活了**为数字附加单位**设置。作为**为数字附加单位**的备选项，您也可以选择**附加来自单位集的单位**。有关更多信息，请参见第 150 页的“单位集”。

在**初始值**列表中，选择**来自数据源**或**定制值**以确定滑块的初始值。

在**事件**栏中，除了可以指定为**数据更改时**事件调用的方法，还可以选中**拖动时触发**复选框。此设置用于确定是在拖动滑块时还是仅在释放滑块时，应连续调用该事件方法。



如果由**数据更改时**事件调用的方法的计算量非常大，以至在拖动滑块时出现延迟，在这种情况下，此设置非常有用。

超链接

超链接对象用于在表单中嵌入超链接。下图显示一个超链接示例。

下图显示相应的**设置**窗口。

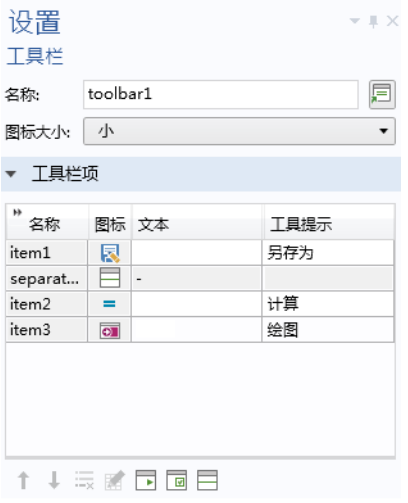
超链接对象支持您可以在 Web 浏览器中使用的 URL 类型，包括：

- 网页：当用户单击网页超链接时，该页面会在用户的默认浏览器中打开。 URL 字符串的格式需为：`protocol://address`，其中的协议为传输协议；例如 HTTP 或 HTTPS。网址可以是部分地址，也可以是完整地址，但建议您使用完整的网址。
- 电子邮件：电子邮件地址通过以下格式指定：`mailto:emailaddress`。此操作会启动用户的默认电子邮件应用程序并准备一封新邮件，其中，收件人编辑框会设为指定的地址。这种从 COMSOL App 交互式发送电子邮件的方式与使用内置方法不同。有关电子邮件内置方法的更多信息，请参见第 316 页的 “电子邮件方法”。

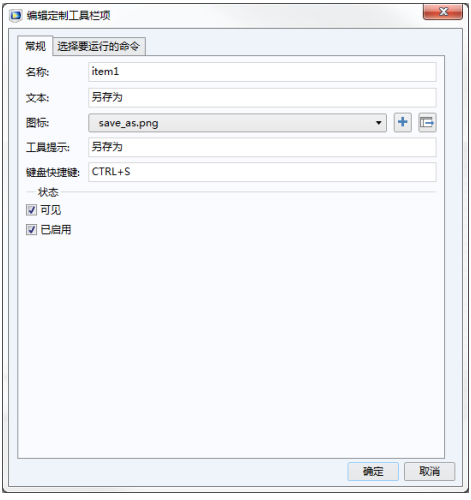
工具栏

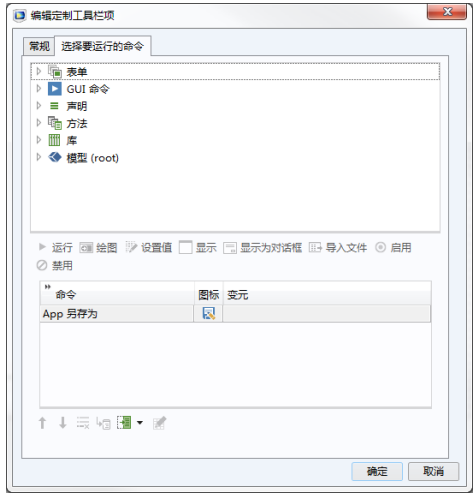
工具栏对象包含带工具栏按钮的工具栏的明细。下图显示一个包含**另存为**、**计算**和**绘制**按钮的工具栏。

此工具栏的**设置**窗口如下图所示。



工具栏项表中的每一行都包含一个对应于工具栏按钮的**项**，或一根**分隔线**。使用表格下方的按钮可以添加项或分隔线、更改行顺序或删除行。单击**编辑**按钮可以显示与每一行关联的**设置**窗口。下图显示 **item1 另存为**项的**设置**窗口。





工具提示框中的文本会在您将光标悬停在工具栏按钮上方时显示为工具提示。**文本**框中的文本将显示在图标 （如果有）旁，否则仅显示文本。**图标**列表、**键盘快捷键**框和**选择要运行的命令**树表示与按钮对象相同的功能。有关更多信息，请参见第 57 页的 “按钮”。

隔片

隔片对象在用户界面中不可见，并且只能用于栅格布局模式。该对象可定义一个固定大小的空间，用于确保相邻的表单对象有足够的空间来显示其中的内容。通常，您可以在表格或图形对象旁使用隔片来确保它们能够正确显示。如

果用户要调整窗口的大小，使它小于隔片大小，则系统会显示滚动条来维持窗口的有效大小。下图显示隔片对象的**设置**窗口。

附录 B - 在 App 之间进行复制

App 树中的许多节点都可以在各个 App 之间进行复制粘贴，包括：表单、表单对象、方法、Java® 实用程序方法、外部库、文件声明、选择列表声明、菜单、菜单项、功能区栏、功能区选项卡和功能区项。

当您在 App 之间复制粘贴表单、表单对象和项时，复制的对象可能包含对其他对象和项的引用。这类引用在复制到 App 后未必有意义。当您从剪贴板粘贴对象时，以下一组规则适用：

- 在复制对象时，会包含表单对象或菜单项中引用的声明，但不一定会粘贴该声明。仅当没有兼容的声明时才会粘贴它。如果存在兼容的声明，则会使用现有声明。兼容声明是指具有相同名称和类型的声明。例如，字符串声明与整数声明不兼容。现有声明可能具有无效的默认值，但在粘贴时不会对此进行检查。
- 引用的全局参数可能具有不同的单位，但仍被认为是兼容的。
- 在复制对象时，不会自动包含直接从其他表单对象引用的表单或表单对象。如果现有对象具有相同名称，则直接引用将指向现有对象。如果复制的对象包含原始引用，则会在引用中使用该对象，而不是具有相同名称的任何现有对象。已复制引用的名称将被更改，以避免名称冲突。
- 不会自动复制模型树中的任何对象，例如，指向几何的图形对象，或指向通过 “数据访问” 呈现的低级设置的输入框。如果引用指向目标 App 的模型树中存在的对象，则使用该引用。
- 在粘贴时，将试图移除对不存在的对象的引用。但按钮的命令序列属于例外，如果其中的命令指向不存在的引用，仍会保留所有命令并将其标记为无效。
- 复制粘贴操作包含局部方法。但不会更新方法的代码。在复制全局方法时，这一点同样适用。
- 按钮或菜单项的命令序列中命令的变元将保留原样。
- 所有图像引用都会在适用的情况下自动复制并添加到图像库。如果现有图像具有相同名称，则会用它来代替复制的版本。
- 不会自动复制引用的文件、声音或方法。不过，可以手动复制粘贴方法。
- 名称与现有对象冲突的所有粘贴对象都将被重命名。对重命名的对象 （来自其他粘贴对象）的任何引用都将得到更新。

附录 C - 文件处理和文件方案语法

文件处理可以说是 App 的一个重要特征。例如， App 可能需要将包含实验数据的电子表格文件作为输入，导入 CAD 文件，或生成并导出报告。“App 开发器”提供了各种工具用来读取和写入整个或部分文件。具体的实现方式根据运行 App 的系统的不同而有所区别。不同计算机上的文件系统可能有所不同，例如运行 COMSOL Multiphysics 的计算机 （在其中开发 App）和安装 COMSOL Server 的计算机 （App 在部署后即可运行）。有关从文件读取以及向其中写入各种数据的更多详细信息，请参阅 Application Programming Guide。

使用 COMSOL Server 处理文件

通常，当您使用 Web 浏览器或 Windows® 版 COMSOL Client 运行 App 时，无法在本地目录读取和写入文件。 App 及其方法在服务器上运行，无法获取客户端文件系统 （Web 浏览器或 COMSOL Client 在其中运行）信息。

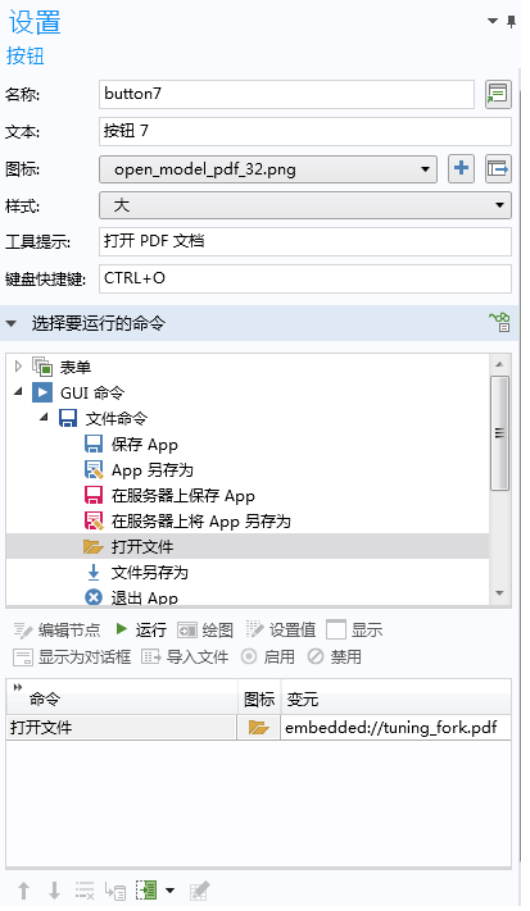
不过，在通过 Web 浏览器和 COMSOL Client 运行 App 时，我们可以使用一些技巧与客户端文件系统进行文件传输。

文件导入对象可以用来向用户请求文件。随后，用户可以浏览到客户端文件系统中的文件，然后将它上传到 COMSOL Server 文件系统，该文件便可以被 App 及其方法使用。例如，用户可以使用这种方式在运行时提供 CAD 文件或实验数据文件。请参见第 295 页的 “文件导入”一节获取相关信息。

举例来说，在按钮的命令序列中，您可以通过运行**导出**或**报告**节点的子节点，导出由嵌入模型生成的数据。请参见第 303 页的 “文件导出”一节获取相关信息。

使用文件命令保存和打开文件

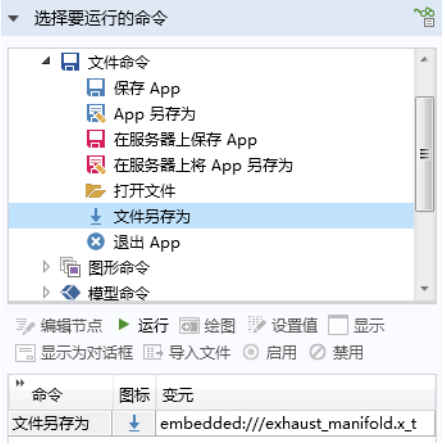
在命令序列使用的编辑器树中，**文件命令**文件夹包含用于保存和加载 App 和文件，以及退出 App 的命令。**打开文件**命令将从服务器选取由方法、模型生成的



任何文件或嵌入 App 的文件，并在客户端上使用关联的应用程序打开它。例如，该命令可以用来打开客户端文件系统中的 PDF 文件，或在客户端显示从模型导出的文本文件或图像。在上图中，**打开文件**命令用来打开 App 的 PDF 文档。相应的 PDF 文件通过存储到**库 > 文件**节点中，从而嵌入 App。该位置的文件通过 `embedded:///` 文件方案语法被引用，有关详细描述，请参见下一节第 293 页的 “文件方案语法”。

要通过方法打开文件，可以使用内置方法 `fileOpen`；另请参见第 316 页的 “操作系统方法”。

要保存文件，可以使用**文件另存为**命令，它与**打开文件**相似。该命令将从服务器文件系统获取任何文件，并显示一个**另存为**对话框，供用户浏览客户端位置来保存文件。此操作类似于在 Web 浏览器中通过链接下载文件。在下图中，**文件另存为**命令用来保存已存储在**库 > 文件**节点中的 CAD 模型。



要通过方法保存文件，可以使用内置方法 `fileSaveAs`；另请参见第 322 页的“GUI 命令方法”。有关保存和导出文件的更多信息，请参见第 303 页的“文件导出”。

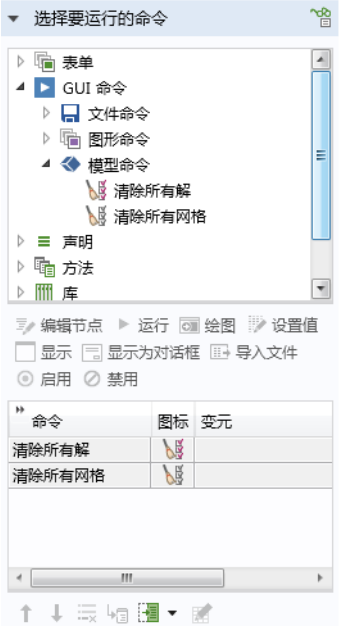
保存 App 和 **App 另存为**命令可用于特定表单对象的命令序列。 **App 另存为**命令将显示**另存为**对话框，供用户指定一个客户端路径来保存整个 App。

同样，**在服务器上保存 App** 和**在服务器上将 App 另存为**命令可用于将整个 App 保存到服务器文件系统。有关相应内置方法的信息，请参见第 322 页的“GUI 命令方法”。

总的来说，用户可以在客户端文件系统上传和下载文件，但由于 Web 浏览器和系统安全设置，如果您不浏览到文件的源或目标位置， App 便无法在后台静默执行该操作。

模型命令

在命令序列使用的编辑器树中，**模型命令**文件夹包含两个命令：**清除所有解**和**清除所有网格**。这两个命令分别用来在保存 App 之前清除解和网格数据，从而减小 MPH 文件的大小。



文件方案语法

为了使 App 可移植，“App 开发器”支持您通过文件方案使用虚拟文件位置。文件方案可以视为一个指针，指向文件系统上的文件，而 App 无需获取文件的实际存储位置（可在**首选项**对话框中设置，见下文）。

不同的文件方案有不同的用途：

- `user` 文件方案用于应在同一个用户多次运行 App 的过程中保持一致的文件。
- `common` 文件方案的功能类似，但它用于应在所有用户之间共享的文件。
- `temp` 文件方案用于应在 App 关闭后被移除的文件。
- `embedded` 文件方案用于在 App 自身存储文件。如果您希望将 App 设为自包含并将其发送给其他人，该文件方案会非常有用。
- `upload` 文件方案用于用户在 App 运行时上传的文件，例如用户浏览的 CAD 文件。

下表汇总了所有可用的文件方案。

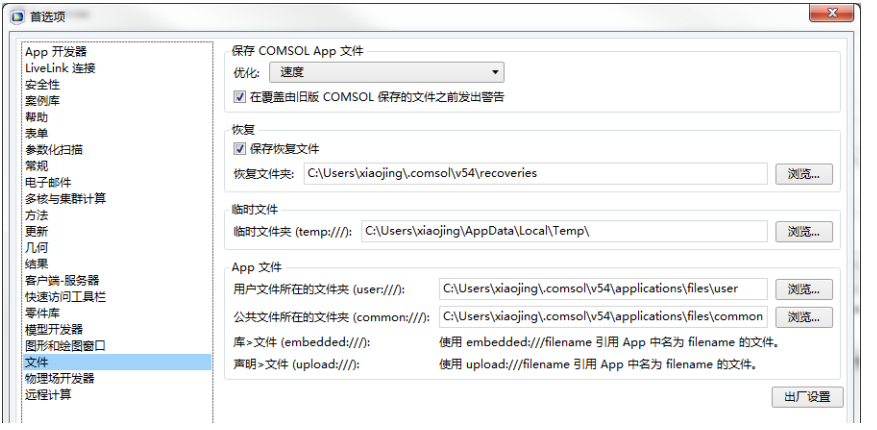
方案	引用对象	默认路径	典型用法
`embedded:///`	使用 “库 > 文件” 嵌入 App 的文件	N/A	实验数据、CAD 文件、网格文件、插值数据
`upload:///`	用户要在运行时上传的文件	由 “文件声明” “设置” 窗口中的 “目标目录” 确定	实验数据、CAD 文件、网格文件、插值数据
`temp:///`	随机临时目录中的文件，该目录对于每个启动的 App 实例都是唯一的。这些文件会在 App 关闭后被删除。	临时文件夹的随机子目录，由 “首选项 > 文件” 中的设置确定	由命令序列或方法生成的临时文件，或导出至客户端（与 COMSOL Server 结合使用）上保存的文件中的数据
`user:///`	当前用户的所有 App 共享的目录中的文件	由 “首选项 > 文件” 中的设置确定	要在会话之间保存的方法中的输出
`common:///`	所有用户共享的目录中的文件	由 “首选项 > 文件” 中的设置确定	许多用户或 App 之间共享的文件

有关可通过 `embedded:///` 语法访问的 “库” 节点中的文件的更多信息，请参见第 201 页的 “库”。

下表汇总了不同文件方案的用法。表格中的复选标记表明该方案可用，(r) 表示这是推荐方案。

用法	EMBEDDED	UPLOAD	TEMP	USER	COMMON
文件用作输入	√ (r)	√			√
文件为输出			√ (r)	√	
用于读取文件的方法	√ (r)	√	√	√	√
用于写入文件的方法			√ (r)	√	
文件是客户端	√	√	√ (r)	√	√

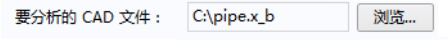
您可以从**文件**菜单访问**首选项**对话框，并在**文件**页面设置临时、用户和公用文件路径的首选项，如下图所示。



文件导入

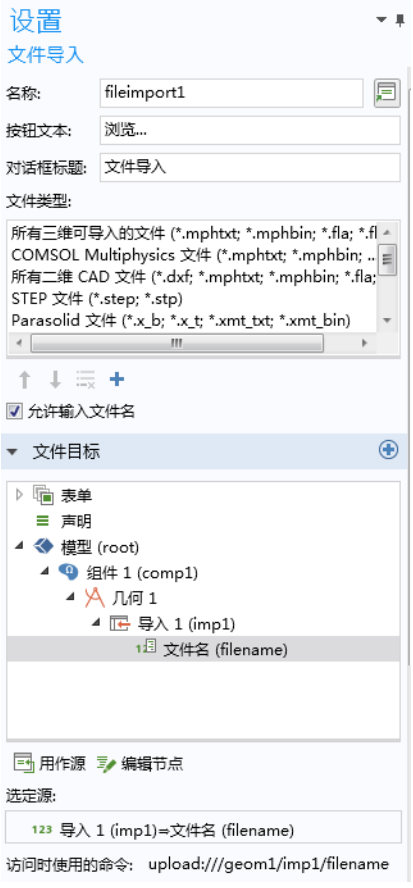
使用模型树和文件导入对象导入 CAD

文件导入对象用于显示文件浏览器，您可以通过关联的输入框浏览文件或输入文件路径和名称。当某个文件尚未加载到 App 时， App 用户可以在运行时通过该对象启用文件导入功能。您可以直接将**文件导入**对象链接到模型树中的文件**导入**节点；例如 CAD **导入**节点。我们以这样一个 App 为例，您可以在 App 运行时选择 CAD 文件并将其导入，如下图所示。



对应的**文件导入**对象如下图所示。

文件导入对象的**设置**窗口包含**文件目标**栏。您可以在该栏中选择支持输入文件名的任意树节点。如下图所示，其中**导入**节点的**文件名**处于选中状态。



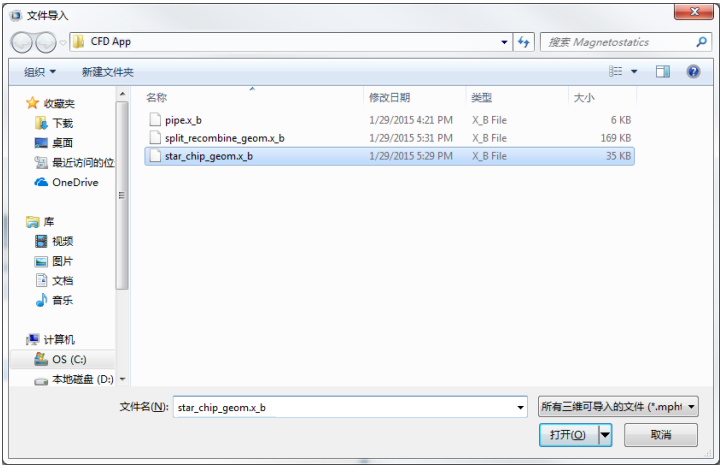
如果您不希望使用**文件导入**对象，可以直接通过按钮或者菜单、功能区或工具栏中的项来引用**文件名**；此外，也可以创建一个方法，将内置方法 `importFile` 作为事件调用，例如

```
importFile("file1");
```

假设存在文件声明 `file1`。

下图显示用于几何导入的功能区项及其**设置**窗口。

在上面的**设置**窗口中，**将文件导入 “导入 1”** 命令将打开文件浏览器，供用户选择文件，如下图所示。



后续命令用于构建和绘制几何，使用 “缩放到窗口大小” 进行缩放，最后设置字符串变量的值 （本例中用来控制卡片堆）。

有关**文件导入**对象的更多信息，请参见第 255 页的 “文件导入”。

通过方法导入文件

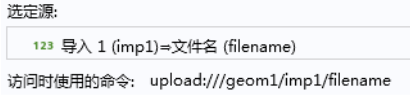
我们继续沿用上一节的示例，假设我们单击**设置**窗口中的**转换成新方法**。对应的代码行显示如何通过方法实现 CAD 导入：

```
importFile(model.geom(“geom1”).feature(“imp1”), “filename”);
model.geom(“geom1”).run();
useGraphics(model.geom(“geom1”), “form1/cardstack1/card3/graphics1”);
zoomExtents(“form1/cardstack1/card3/graphics1”);
mode = “Visualization”;
```

第一行表明使用内置方法 `importFile`。有关使用 `importFile` 方法及其他方法来处理文件的更多信息，请参见第 314 页的 “文件方法” 和 Application Programming Guide。

文件访问和文件声明

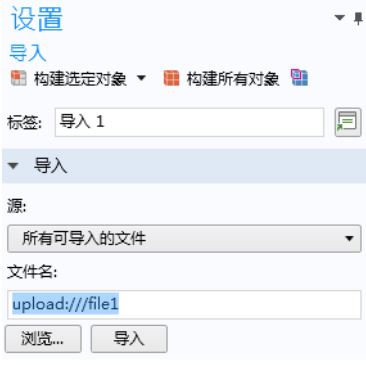
在**文件导入**对象的**设置**窗口底部，您可以看到访问通过方法导入的文件所用的文件方案语法 （位于**访问时使用的命令**旁）。下图显示一个使用**文件目标**和**文件名**的示例。



访问此文件时需要使用文件方案语法 `upload:///geom1/imp1/filename`。

还有一种备选方案，您可以使用**声明**节点下的**文件**声明。（不过，**文件**声明主要用于通过方法代码进行文件导入。）在这种情况下，用户选定的文件可通过 `upload:///file1`、 `upload:///file2` 等语法在表单对象或方法中被引用。文件名句柄 （`file1`、 `file2` 等）稍后可用于引用由用户在运行时选取的实际文件名。另请参见第 149 页的 “文件”。

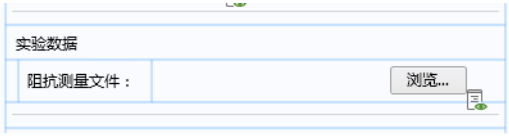
该语法还可用于 “模型开发器” 节点中的任何文件浏览器文本框。下图显示在一个使用几何导入的模型中，**导入**模型树节点的**文件名**框使用的文件引用。



不过，更快的方式是将文件导入对象直接链接到**文件名**框，如前面第 295 页的 “使用模型树和文件导入对象导入 CAD” 一节所述。

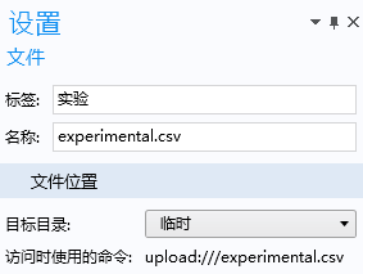
导入实验数据

我们以这样一个 App 为例，用户在 App 运行时提供一个包含实验数据的文件。下图显示这个 App 在栅格布局模式下的文件导入对象。



下图显示对应的文件导入对象的**设置**窗口及其指向文件声明的链接。

在此 App 中，**文件类型**表指定只允许导入 CSV 文件。**文件**声明的**设置**窗口如下图所示。

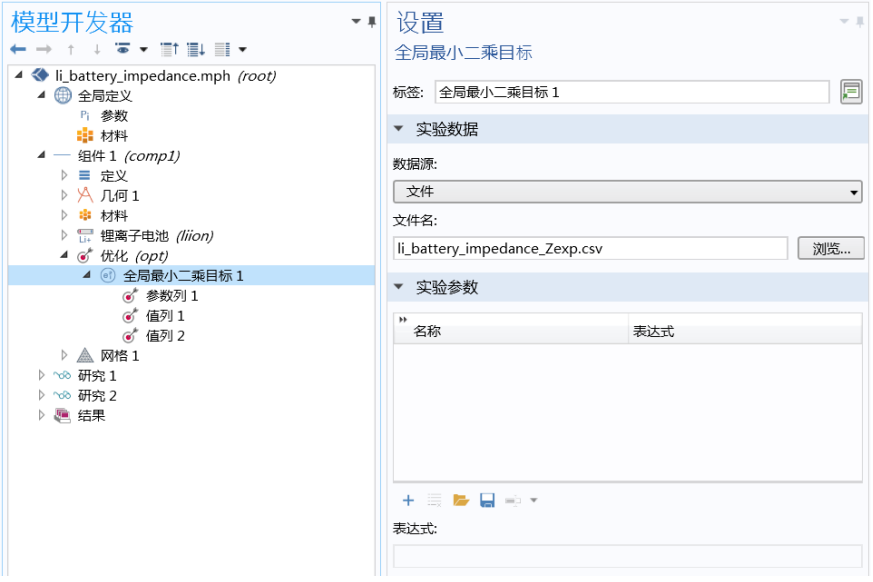


该文件声明用作导入数据的 “目标”，已写入文件 `upload:///experimental.csv`。

请注意，声明中使用的文件扩展名 `.csv` 为可选项，用户在运行时选取的文件可以具有任何名称。例如，在运行时选取的文件名可以为 `my_data.csv`，但在方法代码中引用时，总是使用抽象文件句柄名称 `experimental.csv`。

为了在无需提供实验数据的前提下便能运行 App，可以在 App 中嵌入含有默认实验数据的文件。 App 通过 `embedded:///` 文件方案语法可以访问所用的数据文件，如下图所示。

本例中使用 “优化模块”， App 可以对实验数据执行最小二乘法拟合。



以下方法通过处理逻辑来确定是否存在用户提供的实验数据文件，进而判断是否应使用默认数据集。

```
if (exists(“upload:///experimental.csv”)) {
   with(model.physics(“opt”).feature(“glsobj1”));
     set(“fileName”, “upload:///experimental.csv”);
   endwith();
}
else{
  String s_data = confirm(“No experimental data file was uploaded. Do you
want to use the embedded data?”, “Experimental Data”, “Yes”, “Cancel
Parameter Estimation”);
  if(s_data.equals(“Cancel Parameter Estimation”)){
    return;
  }
}
```

如果存在用户提供的文件，则在物理场接口 `glsobj1` 中，代码会将 `embedded:///experimental_default.csv` 替换为 `upload:///experimental.csv`。

有关文件导入的更多信息，请参见 Application Programming Guide。

使用模型树导出文件

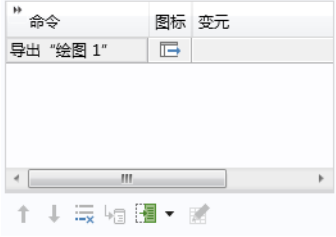
在例如按钮等的命令序列中，您可以导出由嵌入模型生成的数据，具体操作是运行**导出**或**报告**节点的子节点。

在模型树中，**导出**节点可包含以下类型的子节点，用于导出文件：

- **数据**
- **绘制**
- **网格**
- **表格**
- **三维图像**
- **二维图像**
- **一维图像**
- **动画**

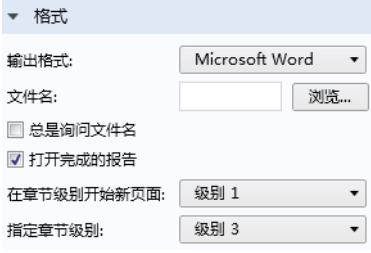
这些节点的**设置**窗口都包含一个带**文件名**编辑框的**输出**栏。下图显示**导出** > **绘图**节点的**设置**窗口。

您可以将**文件名**编辑框保留为空，如上图所示。在例如按钮等的命令序列中，您可以运行对应的**导出 > 绘图**节点，即可在运行时打开文件浏览器窗口，供用户选择位置和文件名，如下图所示。



在开发 App 时，您可能需要重复使用 "模型开发器" 来检查导出的数据。在这种情况下，可以使用**文件名**编辑框来测试文件，选中**总是询问文件名**复选框后，会在运行时打开文件浏览器。

与**导出**子节点类似，**报告**子节点所含的**格式**栏中也有一个**文件名**编辑框，如下图所示。



通过运行**报告**子节点，可以打开文件浏览器窗口，供用户选择报告的位置和文件名。

要更细化地控制文件的导入和导出，您可以改用文件方案。

使用临时文件导出文件

一些 App 可能需要生成临时文件，我们可以使用 `temp:///` 文件方案来实现。临时文件存储在随机临时目录中，该目录对于每个启动的 App 实例都是唯一的。这些文件会在 App 关闭后被删除。临时文件可以由命令序列或方法生成，也可以是要在客户端 （与 COMSOL Server 结合使用时）保存的输出。

下例显示**导出 > 绘图**节点的**设置**窗口，用于将绘图数据导出为数值。

其中**输出**栏的**文件名**已设为 `temp:///lineplot.txt`。

要将本例中的绘图保存到光盘，需要创建一个按钮。在按钮**设置**窗口的**选择要运行的命令**栏中，通过选择上面创建的**导出 > 绘图**节点并单击**运行**，首先创建一个输出图形文件。其次，选择 **GUI 命令 > 文件命令 > 文件另存为**，然后再次单击**运行**。

在按钮**设置**的**输出**栏中，将文件名设为通过**导出绘图**命令创建的临时文件的名称，本例中为 `temp:///lineplot.txt`。

文件另存为命令提供一个专用的**编辑变元**对话框，供您轻松访问所有嵌入文件以及所有文件方案的快捷方式。

相应的方法代码如下：

```
model.result().export("plot1").run();
fileSaveAs("temp:///lineplot.txt");
```

使用临时文件导出文件

请注意，在上述示例中，第一步是通过调用 `model.result().export("plot1").run()` 将文件写入临时文件。此步骤由 App 自动完成。在第二步中，`fileSaveAs` 方法打开一个文件浏览器，供 App 用户选择文件位置，例如计算机本地文件系统上的文件夹或网络文件夹。要使 App 在 Web 浏览器中正常运行，还需要执行额外的步骤。由于典型 Web 浏览器的安全设置，App 无法将文件自动保存到任意位置；但支持将文件保存到一些特定位置，包括临时文件夹，其位置在**首选项**对话框设置中指定。其他位置包括用户和公用文件夹，也在**首选项**设置中指定。

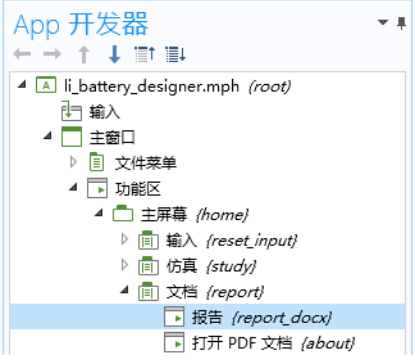
有关文件导出的更多示例，请参见 Application Programming Guide。

使用底层功能创建报告

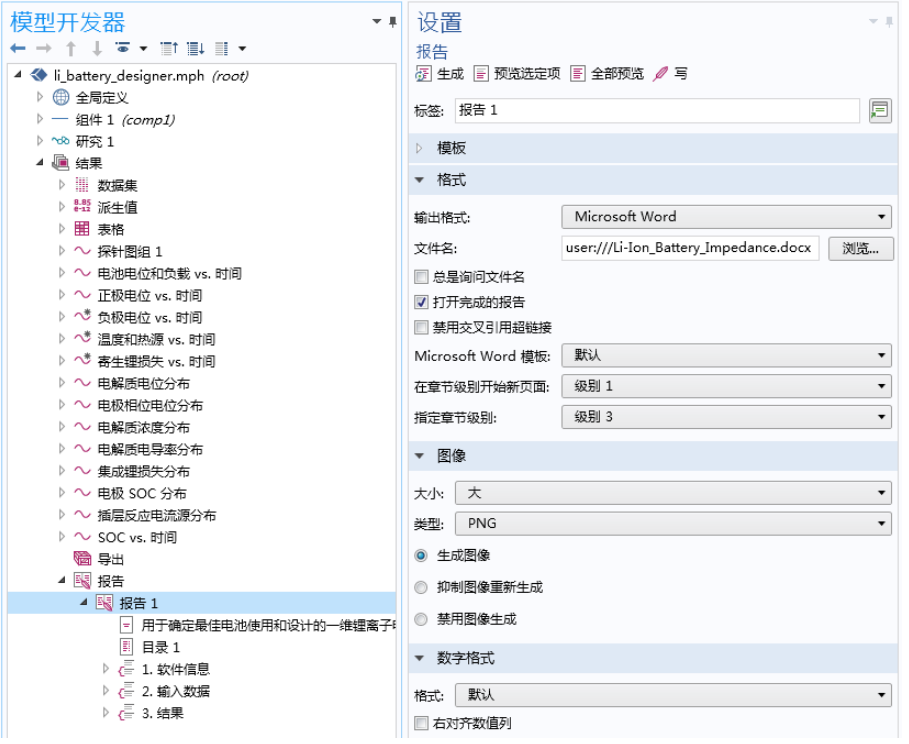
本节描述如何使用底层功能来创建报告。如果了解更直接的方法，请参见第 303 页的 “文件导出”。

在下例显示的 App 中，用户可以保存 Microsoft® Word® 格式 (`.docx`) 的报告。下图显示该 App 的功能区中的选项卡。此选项卡的**文档**栏有一个**报告**按钮。

关联的 App 树节点如下图所示。



下图通过 “模型开发器”的**报告**节点**设置**窗口中的**文件名**编辑框，显示了语法 `user://file` 的用法。



在此 App 中，**打开完成的报告**复选框处于选中状态，这意味着在创建报告后将打开一个 Word® 文档。 App 用户随后可以从 Word® 文件菜单保存该报告。

在本例中，可以通过类似方式来使用文件方案 `common:///`。当 App 重复使用相同的文件时， **user** 与 **common** 文件方案特别有用。

下图显示**报告**功能区项的**设置**窗口。

`p_report` 方法包含以下代码：

```
model.result().report("rpt1").run();
```

文件方案语法还能直接在方法中使用。以下是一个用于导出 HTML 报告的方法中的代码。

```
String answerh = request("Enter file name","File Name", "Untitled.html");
if(answerh != null){
  model.result().report("rpt1").set("format","html");
  model.result().report("rpt1").set("filename","user:///"+answerh);
  model.result().report("rpt1").run(); }}
```

附录 D - 键盘快捷键

下表列出 “App 开发器”中提供的键盘快捷键。

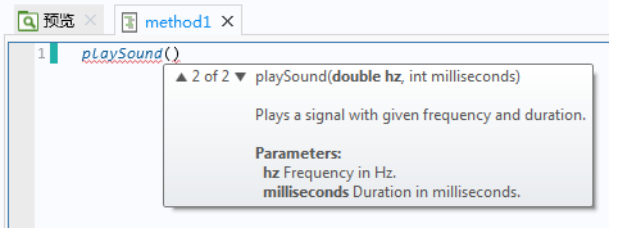
快捷方式	操作	APP 开发器	表单编辑器	方法编辑器
Ctrl+A	全选	√	√	√
Ctrl+D	取消全选		√	
Ctrl+C	复制	√	√	√
Ctrl+V	粘贴		√	√
Ctrl+X	剪切	√	√	√
Del	删除	√	√	√
Ctrl+N	新建 App	√	√	√
Ctrl+S	保存 App	√	√	√
Ctrl+F8	测试 App	√	√	√
Alt+ 单击	编辑特定的表单对象		√	
Ctrl+Pause	停止方法	√		
Ctrl+Shift+F8	应用更改	√	√	√
Ctrl+R	录制代码			√
F11	切换到节点			√
Ctrl+K	创建快捷方式	√	√	√
F1	显示帮助	√	√	√
F2	重命名适用的节点	√		
F3	禁用适用的节点	√		
F4	启用适用的节点	√		
Ctrl+ 上箭头	将适用的节点上移	√		
Ctrl+ 下箭头	将适用的节点下移	√		
Ctrl+Z	撤消	√	√	√
Ctrl+Y	重做 （Mac 系统中为 Control+Shift+Z）	√	√	√
F5	继续 （在调试程序中）			√
F6	步进 （在调试程序中）			√
F7	单步执行 （在调试程序中）			√

快捷方式	操作	APP 开发器	表单编辑器	方法编辑器
F8	检查语法			√
Ctrl+F	查找和替换方法中的文本			√
Ctrl+/、Ctrl+Space 或 Ctrl+OEM2	自动完成方法代码			√
Ctrl+U	将选定的代码设为小写			√
Ctrl+Shift+U	将选定的代码设为大写			√
Ctrl+B	在选定的行中切换断点			√
Ctrl+M	在匹配的圆括号、方括号或大括号之间切换			√
Ctrl+Shift+M	选择匹配的圆括号、方括号或大括号中的所有字符			√
Ctrl+ 向上滚动滚轮	在方法代码窗口中放大			√
Ctrl+ 向下滚动滚轮	在方法代码窗口中缩小			√
Ctrl+ 所有箭头键	对所选表单对象的位置进行微调		√	
所有箭头键	对所选表单对象的位置进行微调		√	
Ctrl+Alt+A	转至 “App 开发器”窗口		√	√
Ctrl+Alt+M	转到 “模型开发器”	√	√	
Ctrl+Alt+ 单击	创建局部方法或打开与表单对象关联的方法		√	
Ctrl+Alt+ 双击	从 “方法编辑器”代码中打开方法			√
Alt+F4	关闭窗口	√	√	√
Ctrl+F4	关闭文档		√	√
Ctrl+Shift+F4	关闭所有文档		√	√
Ctrl+7	打开和关闭注释			√
按下 Ctrl 并单击。同时按住键盘键和鼠标按键，并拖动鼠标。	复制表单对象		√	

附录 E - 内置方法库

本附录列出 “方法编辑器” 中提供的所有内置方法，在模型对象和 App 对象上操作的方法不在此列。有关使用内置方法的详细信息以及所用语法的完整信息，请参阅 Application Programming Guide 和 Programming Reference Manual。

您还可以通过另一种方式来学习这些方法的语法，可以键入方法的名称，然后按 Ctrl+/ 或 Ctrl+Space 来使用代码补全功能。此时会打开一个窗口，其中包含有关语法和方法签名的信息。



模型实用程序方法

模型实用程序方法支持将 MPH 文件的模型对象零件加载到方法中做进一步处理。

名称	描述
`clearModel`	清除模型对象内容。
`createModel`	创建具有给定标记的新模型。
`removeModel`	移除模型。无法移除嵌入的模型。
`modelTags`	针对所有加载的模型 （包括嵌入模型）返回一组模型标记。
`uniqueModeltag`	返回未使用的模型标记。
`getModel`	返回具有指定标记的模型。
`loadModel`	从文件加载具有指定标记的模型。
`loadProtectedModel`	从文件加载具有指定标记的密码保护模型。
`loadRecoveryModel`	从恢复目录 / 文件夹结构加载模型。
`saveModel`	将模型保存到文件。文件名可以是文件方案路径；如果安全设置允许，也可以是服务器文件路径。
`getComsolVersion`	以字符串形式返回当前软件版本。

文件方法

名称	描述
`readFile`	以字符串的形式返回给定文件中的内容。
`openFileStreamReader`	返回 `CsReader` 对象，该对象可用于从给定文件名称逐行或逐字符读取内容。
`openBinaryFileStreamReader`	返回 `CsBinaryReader` 对象，该对象可用于从给定文件逐字节读取内容。
`readMatrixFromFile`	将给定文件的内容读入双精度矩阵。该文件的电子表格类型格式与模型树 “导出”节点中的相同。
`readStringMatrixFromFile`	将给定文件的内容读入字符串矩阵。该文件的电子表格类型格式与模型树 “导出”节点中的相同。
`readCSVFile`	将包含逗号分隔值的文件 （CSV 文件）读入字符串矩阵。期望该文件使用 RFC 4180 格式的 CSV。
`writeFile`	将数组数据写入给定的文件。如果使用电子表格的格式，则可以通过 `readMatrixFromFile` 或 `readStringMatrixFromFile` 读取数据。
`openFileStreamWriter`	返回 `CsWriter` 对象，它可以写入给定的文件。
`openBinaryFileStreamWriter`	返回 `CsBinaryWriter` 对象，该对象可用于逐字节写入给定的文件。
`writeCSVFile`	将给定的双精度数组或字符串数组写入 CSV 文件。CSV 使用 RFC 4180 格式。
`exists`	测试是否存在给定名称的文件。
`deleteFile`	如果存在给定名称的文件，则将其删除。该文件会从服务器删除。
`copyFile`	在服务器上复制文件。源和目标名称都可以使用文件方案路径。
`importFile`	显示文件浏览器对话框，并将选定的文件上传到具有给定名称的文件声明中。或者，也可以将选定的文件上传到给定模型对象实体的 “文件名”文本框中。
`writeExcelFile`	写入从 Excel 文件的指定工作表的指定单元格开始的给定字符串数组数据。
`readExcelFile`	将从 Excel 文件的指定工作表的指定单元格开始的内容读取到二维字符串数组。

名称	描述
`getFilePath`	返回与特定文件方案路径对应的服务器代理文件的绝对服务器文件路径，如果给定路径的服务器代理文件不存在，则返回 null。 此方法可用于将使用 `temp:///` 方案的文件等的路径传递给外部代码或应用程序。
`getClientFileName`	返回客户端文件系统中已上传文件的原始名称 （如果没有与给定文件方案路径匹配的上传文件，则返回 null）。 此方法仅用于提供用户界面反馈；例如，获取有关正在使用的上传文件的信息。无法保证原始文件仍存在于客户端上，即使当前客户端与原始客户端相同也是如此。
`getClientFilePath`	返回客户端文件系统中已上传文件的原始路径 （如果没有与给定文件方案路径匹配的上传文件，则返回 null）。 此方法仅用于提供用户界面反馈；例如，获取有关正在使用的上传文件的信息。无法保证原始文件仍存在于客户端上，即使当前客户端与原始客户端相同也是如此。

操作系统方法

名称	描述
`executeOSCommand`	使用给定的命令 （完整路径）和参数执行操作系统命令。该命令在适用的情况下在服务器端运行。
`fileOpen`	在客户端使用关联程序打开文件。另请参见 “文件方法” 一节。
`getUser`	返回运行 App 的用户的用户名。如果 App 未从 COMSOL Server 运行，则返回首选项设置常规 > 用户名 > 名称的值。
`openURL`	在客户端上的默认浏览器中打开 URL。
`playSound`	在客户端播放声音。

电子邮件方法

名称	描述
`emailFromAddress`	从 COMSOL Server 或首选项设置返回电子邮件发件人地址。
`sendEmail`	向指定的收件人发送电子邮件，其中包含指定的主题、正文文本，可以不含附件，也可以包含从嵌入模型的 “报告”、“导出” 和 “表格” 节点创建的多个附件。
`userEmailAddress`	返回与当前登录用户关联的用户电子邮件地址，如果该用户尚未配置电子邮件地址，则返回空字符串。

电子邮件类方法

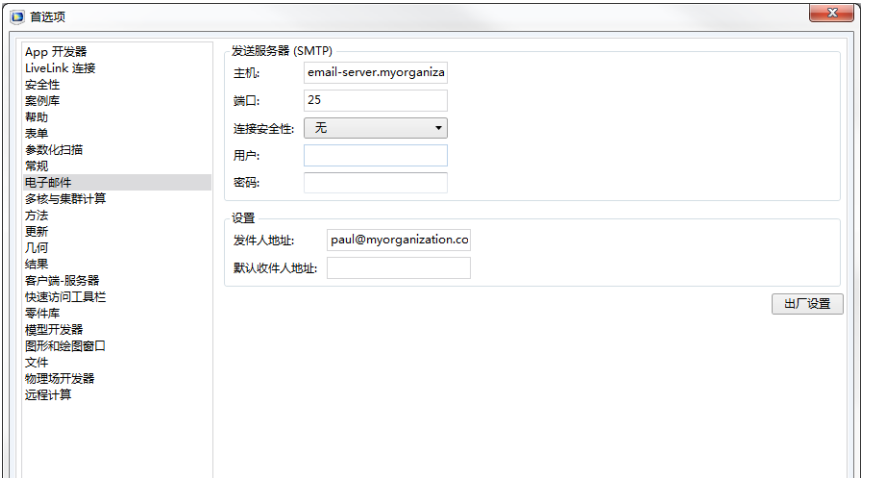
`EmailMessage` 类可用于创建定制的电子邮件。

名称	描述
`EmailMessage`	创建新的 `EmailMessage` 对象。
`EmailMessage.setServer`	设置用于此电子邮件的电子邮件 (SMTP) 服务器主机和端口。
`EmailMessage.setUser`	设置用于电子邮件 (SMTP) 服务器身份验证的用户名和密码。必须在 `setServer` 方法之后调用此方法。
`EmailMessage.setSecurity`	设置电子邮件 (SMTP) 服务器通信的连接安全类型。
`EmailMessage.setFrom`	设置发件人地址。
`EmailMessage.setTo`	设置收件人地址。

名称	描述
`EmailMessage.setCc`	设置抄送地址。
`EmailMessage.setBcc`	设置密件抄送地址。
`EmailMessage.setSubject`	设置电子邮件主题行。请注意，不允许使用换行符。
`EmailMessage.setBodyText`	将电子邮件正文设为纯文本。电子邮件可以包含文本和 HTML 正文。
`EmailMessage.setBodyHtml`	将电子邮件正文设为 HTML 文本。电子邮件可以包含文本和 HTML 正文。
`EmailMessage.attachFile`	从文件添加附件。附件 MIME 类型由文件扩展名确定。
`EmailMessage.attachFile`	从具有指定 MIME 类型的文件添加附件。
`EmailMessage.attachFromModel`	添加从模型的报告、导出或表格特征创建的附件。
`EmailMessage.attachText`	添加具有指定子 MIME 类型的文本附件，如纯文本或 HTML 文本。
`EmailMessage.attachBinary`	从具有指定 MIME 类型的字节数组添加附件。
`EmailMessage.send`	将电子邮件发送到电子邮件 (SMTP) 服务器。电子邮件对象只能发送一次。

电子邮件首选项

您可以打开**首选项**对话框，在**电子邮件**页面设置电子邮件发送 (SMTP) 服务器的首选项，如下图所示。



COMSOL Server 也提供一组类似的电子邮件首选项。

GUI 相关方法

名称	描述
直接调用方法	使用方法的名称从 “方法”列表调用该方法；例如 `method1()`、 `method2()`。
`callMethod`	从 “方法”列表调用方法的备选方式；用于内部及名称冲突的情况。
`useGraphics`	在第二个变元中的名称或名称路径给定的图形表单对象中绘制给定的实体 （绘图组、几何、网格或显式选择）。
`useForm`	在当前主窗口中显示具有给定名称的表单。等同于表单对象的 `use` 方法；请参见下文。
`closeDialog`	关闭具有给定名称并显示为对话框的表单。
`dialog`	将具有给定名称的表单显示为对话框。等同于表单对象的 `dialog` 方法；请参见下文。
`alert`	停止执行并显示包含给定文本的通知邮件。
`alert`	停止执行并显示包含给定文本和标题的通知邮件。
`confirm`	停止执行并显示包含给定文本和标题的确认对话框。同时还显示两个或三个按钮，如 “是”、“否”和 “取消”。
`error`	停止执行并打开一个包含给定消息的错误对话框。
`request`	停止执行并显示一个包含文本框的对话框，用于请求用户输入。
`message`	将消息发送到消息日志。
`evaluateToResultsTable`	计算由第二个变元的名称或名称路径给定的表格对象中的给定实体 （一个派生值），该值随后将作为派生值计算的默认目标。如果第三个变元为 true，则在添加新数据之前会清除表格。否则，会追加数据。
`evaluateToDoubleArray2D`	计算给定的实体 （一个派生值），并返回实数表中作为双精度矩阵生成的非参数列部分。系统会兼顾数值特征中的所有设置，但会忽略当前表格中连接到数值特征的设置。
`evaluateToIntegerArray2D`	计算给定的实体 （一个派生值），并返回实数表中作为整型矩阵生成的非参数列部分。系统会兼顾数值特征中的所有设置，但会忽略当前表格中连接到数值特征的设置。

名称	描述
evaluateToStringArray2D	计算给定的实体 （一个派生值），并返回潜在的复值表中作为字符串矩阵生成的非参数列部分。系统会兼顾数值特征中的所有设置，但会忽略当前表格中连接到数值特征的设置。
useResultsTable	显示 resultsTable 表单对象中的 tableFeature 的值。
getChoiceList	返回 ChoiceList 类型的对象，表示声明分支下的选择列表节点。 ChoiceList 类型具有关联的方法，可供您轻松更改值和显示名称，请参阅 Application Programming Guide。
setFormObjectEnabled	为名称或名称路径指定的表单对象设置启用状态。
setFormObjectVisible	为名称或名称路径指定的表单对象设置可见状态。
setFormObjectText	为第二个变元中的名称或名称路径指定的表单对象设置文本。如果无法为指定的表单对象设置文本，此方法会抛出错误。
setFormObjectEditable	为名称或名称路径指定的表单对象设置可编辑状态。仅功能仅适用于文本框对象。
setMenuBarItemEnabled	为第一个变元中的名称或名称路径 （来自菜单栏）指定的菜单栏项设置启用状态。
setMainToolbarItemEnabled	为第一个变元中的名称或名称路径 （来自主工具栏）指定的主工具栏项设置启用状态。
setFileMenuItemEnabled	为第一个变元中的名称或名称路径 （来自文件菜单）指定的文件菜单项设置启用状态。
setRibbonItemEnabled	为第一个变元中的名称或名称路径 （来自主窗口）指定的功能区项设置启用状态。
setToolbarItemEnabled	为第一个变元中的名称或名称路径指定的工具栏表单对象项设置启用状态。
useView	将视图应用到由第二个变元中的名称或名称路径给定的图形内容。
resetView	将第二个变元中的名称或名称路径给定的图形内容中的视图重置为初始状态。
getView	返回由第二个变元中的名称或名称路径给定的图形内容当前使用的视图。
setWebPageSource	为第一个变元中的名称或名称路径指定的表单对象设置源。

名称	描述
`getScreenHeight`	返回客户端系统中主屏幕的高度 （以像素计），如果使用 Web 客户端，则返回浏览器窗口的高度。
`getScreenWidth`	返回客户端系统中主屏幕的宽度 （以像素计），如果使用 Web 客户端，则返回浏览器窗口的宽度。

GUI 命令方法

名称	描述
`clearAllMeshes`	清除所有网格。
`clearAllSolutions`	清除所有解。
`clearSelection`	清除给定图形对象中的选择内容。
`exit`	退出 App。
`fileOpen`	在客户端使用关联程序打开文件。
`fileSaveAs`	将文件下载到客户端。另请参见 “文件方法”一节。
`printGraphics`	打印给定的图形对象。
`saveApplication`	保存 App。
`saveApplicationAs`	使用其他名称保存 App。（或另存为 MPH 文件。）
`scenelight`	在给定的图形对象中切换场景光。
`selectAll`	选择给定图形对象中的所有对象。
`transparency`	在给定的图形对象中切换透明度。
`zoomExtents`	使整个模型在给定的图形对象中可见。

调试方法

名称	描述
`debugLog`	将输入变元的值复制到 “调试日志”窗口。输入变元可以为字符串类型、双精度型、整型或布尔型的标量、一维数组或二维数组。

用于外部 C 库的方法

外部方法

名称	描述
`external`	返回由库特征的名称给定的外部 C （本机）库的接口。External 类使用 Java 本地接口 (JNI) 框架。有关更多信息，请参阅 Application Programming Guide。

外部方法返回的方法

外部方法通过以下方法返回 External 类型的对象：

名称	描述
`invoke`	使用提供的变元在库中调用一个命名的本机方法。
`invokeWideString`	使用提供的变元在库中调用指定的本机方法。
`close`	释放库和资源。如果您不调用此方法，系统会在不需要外部库时自动调用它。

进度方法

名称	描述
`setProgressInterval`	设置用于顶级进度的进度间隔，并在该级别显示消息。 调用此方法可以隐式重置先前通过调用 `setProgress()` 来设置的任何手动进度。
`setProgress`	为用户控制的进度级别设置值。
`resetProgress`	移除所有进度级别，并将进度重置为 `0`，以及将消息重置为空字符串。
`showIndeterminateProgress`	显示一个进度对话框，其中包含不确定的进度条、给定的消息和可选的 “取消” 按钮。
`showProgress`	显示一个进度对话框，其中包含可选的 “取消” 按钮和模型进度，以及一个或两个级别的进度信息。
`closeProgress`	关闭当前显示的进度对话框。
`startProgress`	将给定的进度条表单对象名称的值重置为 0。
`setProgressBar`	在 0-100 范围内设置给定的进度条表单对象名称的值，并设置关联的进度消息。

日期和时间方法

名称	描述
`currentDate`	以字符串 （其格式取决于服务器的默认设置）形式返回当前日期。
`currentTime`	以字符串 （不包含日期，并且其格式取决于服务器的默认设置）形式返回当前时间。
`formattedTime`	使用给定的格式返回格式化时间。格式可以为时间单位，也可以为描述较长格式的文本。
`sleep`	按指定的毫秒数休眠。
`timeStamp`	从 UTC 时间 1970 年 1 月 1 日午夜至当前时间的毫秒数
`getExpectedComputationTime`	返回用于描述 App 的近似计算时间的字符串。该字符串可通过方法 `setExpectedComputationTime` 进行更改。
`setLastComputationTime`	设置上次计算时间，以覆盖自动生成的时间。 您可以使用 `timeStamp` 方法记录时间差，然后设置以 ms 计的测量时间 （长整型）。
`getLastComputationTime`	以给定的格式返回上次计算时间。格式可以为时间单位，也可以为描述较长格式的文本。系统会将此格式本地化，并将输出转换为当前语言设置。

许可证方法

名称	描述
`checkoutLicense`	为每个指定产品签出一个许可证。
`checkoutLicenseForFile`	为打开 MPH 文件所需的每个产品签出一个许可证。
`checkoutLicenseForFileOnServer`	为打开 MPH 文件所需的每个产品签出一个许可证。
`getLicenseNumber`	返回包含当前会话的许可证编号的字符串。例如：`licensenumber=getLicenseNumber()`
`hasProduct`	如果 COMSOL 安装包含运行指定产品所需的软件组件，则返回 true。
`hasProductForFile`	如果 COMSOL 安装包含运行指定 MPH 文件所需的软件组件，则返回 true。
`hasProductForFileOnServer`	如果 COMSOL 安装包含运行指定 MPH 文件所需的软件组件，则返回 true。

转换方法

名称	描述
`toBoolean`	将字符串和字符串数组转换为布尔型。（‘`true`’返回 true，其他所有字符串则返回 false）。
`toDouble`	将字符串和字符串数组转换为双精度型。
`toInt`	将字符串和字符串数组转换为整型。
`toString`	将布尔型、整型和双精度型数据 （包括数组）转换为字符串。

数组方法

名称	描述
`getColumn`	为二维数组 （矩阵）中的指定列返回字符串、双精度、整型或布尔数组。例如，当已经从文件读取值并且只需在表格中显示特定列时，该方法非常有用。
`getSubMatrix`	为输入矩阵返回矩形子矩阵。可用于字符串、双精度、整型或布尔二维数组。
`insert`	在数组中插入一个或多个元素，并返回扩展的数组。可用于字符串、双精度、整型或布尔数组。
`append`	在数组末尾添加一个或多个元素，并返回扩展的数组。可用于字符串、双精度、整型或布尔数组。

名称	描述
remove	从数组移除一个或多个元素，并返回缩短的数组。可用于字符串、双精度、整型或布尔数组。
insertRow	将一个或多个行插入二维矩形数组，并返回扩展的数组。可用于字符串、双精度、整型或布尔数组。
appendRow	在二维矩形数组末尾添加一个或多个行，并返回扩展的数组。可用于字符串、双精度、整型或布尔数组。
removeRow	从二维数组移除一个或多个行，并返回简化的数组。可用于字符串、双精度、整型或布尔数组。
replaceRow	替换二维矩形数组中的一个或多个行，并返回该数组。可用于字符串、双精度、整型或布尔数组。
insertColumn	在二维矩形数组中添加一个或多个列，并返回扩展的数组。可用于字符串、双精度、整型或布尔数组。
appendColumn	在二维矩形数组末尾添加一个或多个列，并返回扩展的数组。可用于字符串、双精度、整型或布尔数组。
removeColumn	从二维矩形数组中移除一个或多个列，并返回较小的数组。可用于字符串、双精度、整型或布尔数组。
replaceColumn	替换二维矩形数组中的一个或多个列，并返回该数组。可用于字符串、双精度、整型或布尔数组。
matrixSize	将矩阵的行数和列数作为长度为 2 的整型数组返回。可用于字符串、双精度、整型或布尔数组。

字符串方法

名称	描述
`concat`	使用给定的分隔符将给定的字符串数组或矩阵连接到单个字符串。
`contains`	如果给定的字符串数组包含给定的字符串，则返回 true。
`find`	返回一个数组，其中包含指向字符串数组中字符串的出现总数的索引。
`findIn`	返回指向字符串数组中出现的第一个字符串的索引，或指向字符串中出现的第一个子字符串的索引。
`length`	返回字符串的长度。
`replace`	返回一个字符串，其中的字符串已替换为另一个字符串。
`split`	通过在给定分隔符处拆分给定的字符串，返回一个字符串数组。
`substring`	返回从给定位置开始的具有给定长度的子字符串。
`unique`	返回在给定字符串数组中包含唯一值的字符串数组。

集合方法

名称	描述
`copy`	返回给定数组或矩阵的副本。可用于字符串、双精度、整型或布尔数组。
`equals`	如果给定数组中的所有元素都相等且元素的数量相同，则返回 true。可用于字符串、双精度、整型或布尔数组。对于双精度，系统会使用相对容差进行比较。
`sort`	对给定的数组排序。注：数组按位置进行排序。可用于字符串、双精度或整型数组。如果是二维数组 （矩阵），则列按其行值从上到下进行排序。
`merge`	返回一个数组，其中包含从给定数组合并的所有元素。可用于字符串、双精度或整型数组。

With、 Get 和 Set 方法

名称	描述
`with`	可用于使代码更紧凑。
`endwith`	用于结束 with 语句。
`set`	设置布尔型、整型、双精度型或字符串属性值。支持标量、数组或矩阵属性。

名称	描述
setIndex	为给定索引处的矩阵或矢量设置字符串、双精度型或整型属性值。
getIntArray	获取整数矢量属性。
getIntMatrix	获取整型矩阵属性。
getBoolean	获取布尔属性。
getBooleanArray	获取布尔矢量属性。
getBooleanMatrix	获取布尔矩阵属性。
getDouble	获取双精度属性。
getString	获取字符串标量、矢量或矩阵属性。
getDoubleArray	获取双精度矢量属性或参数。
getDoubleMatrix	获取双精度矩阵属性或参数。
getStringArray	获取字符串矢量属性或参数。
getStringMatrix	获取字符串矩阵属性或参数。
getDblStringArray	将值作为字符串矩阵返回。
getInt	获取整型属性。
get	返回变量表达式。
descr	返回变量描述。

一般提示

- 在报告文件中包含输入数据和对应的输出数据。
- 保持简单直观。根据需要提供帮助、提示和文档。
- 防止错误操作：“安全输入 / 输出”、“重置为默认数据”等。
- 将缩略图随模型一起保存。
- 包含描述文本 （将显示在 COMSOL Server 库中）。
- 在要运行 App 的计算机平台上进行测试。
- 保持简约。从开发人员的角度来看，这样更易于确保逻辑正确，以及对 App 进行组织、调试、维护和进一步开发。从用户的角度来说，这样会使 App 更易于使用。简约方法需要开发人员在开发时倾注更多的精力，但后续维护工作会少得多，并支持分享给更多的用户。
- 如果 App 的大小易于控制，则将其嵌入模型中的库。
- 显示预计计算时间和计算后的实际计算时间。
- 在取消计算后，应清除先前计算产生的输出数据。
- 根据需要设置密码保护。（请记住：如果您忘记密码，您无法向任何人寻求帮助。）

方法

- 请勿创建不必要的方法。

  方法越少，您在进行查找时要浏览的方法列表就越短。较少的方法通常意味着您需要考虑的代码行也较少。

  - 如果您编写的多种方法在本质上具有相同的功能，请考虑将它们合并为一个方法，并通过输入变元来处理不同的情况。
  - 如果某个方法只从一个位置进行调用，则不要创建该方法。在相应的位置插入代码即可。
- 如果某个方法仅在表单中使用，或由表单事件或表单对象事件触发，则创建局部方法。

- 使用描述性名称为方法命名，以在按字母顺序排序时将相似的方法分组在一起。这样，您只需记住少量的信息，就能轻松找到所需的方法。长名称比难以理解的短名称更适用。

  方法命名示例：

  - 不输出任何内容的所有方法都以 `p` 开头 （`p` 表示过程）。
  - 提供输出的所有方法都以 `f` 开头 （`f` 表示函数）。
  - 所有菜单项方法都以 `m` 开头 （`m` 表示菜单）。
  - 您要经常访问的方法以 `a` 开头，以使其显示为列表中的第一项。
  - 所有绘图方法都以 `Plot` 开头 （`mPlotMesh`、 `mPlotResults` 表示菜单项方法）。
- 以上几点也适用于方法代码：保持简约，使用尽可能少的代码行和变量，为变量使用描述性名称，使用长名称代替难以理解的短名称，以及优化代码从而实现高效运行。
- 以上几点同样适用于声明：使用适当的名称，避免过多使用方法，以及在使用变量的位置 （在表单和方法或模型中）声明变量。

表单

- 请勿创建不必要的表单。
- 使用描述性名称为表单命名。理由与方法相同。
- 充分利用多种不同类型的表单对象。不同的表单对象有不同的用处。
- 请勿插入不必要的表单对象。输入数据的选项过多可能会造成 App 不易于使用。输出数据过多不便于查找重要信息。
- 插入文本框，供用户添加注释，以便在保存 App 时与用户的输入和输出数据集一同保存。
- 请考虑插入一个带方法的按钮，以重置为默认数据。
- 应用 “安全输入 / 输出”：
  - 当输入框中的输入数据超出限值时通知用户。您可以通过输入框的 “数据更改时”事件触发的通知来执行该操作，也可以在表单对象设置窗口中设置限制，系统随后会设置硬限制。在用于生成通知的方法中，您只需向用户发出警告，如果用户选择继续操作，则允许输入数据。
  - 在输出框中，给出合理的精度。如果当前结果与当前输入数据无关，则显示它。如果计算失败，则显示它。
- 包含工具提示、帮助、文档、提示和综合报告。
- 向用户提供有关在典型计算机上使用默认输入数据运行仿真所需时间的信息。根据 App 的不同，运行时间可能以秒、小时甚至天数来计算，用户在单击 “计算”按钮之前应该了解这一信息。请考虑在计算完成时，通过播放声音来提醒用户。用户在等待结果的同时可能正在处理其他事情。（如

果计算时间确实很长，则在计算完成时向用户或相关方发送一封附带报告的电子邮件，应该是一个更好的选择。）

- 精心安排表单布局。好看的表单外观会使 App 变得更易于使用且更有趣。
- 请考虑为按钮和菜单项设置键盘快捷键。

附录 G - 案例库示例

您可以在 “案例库” 中找到许多示例 App，它们展示了 “App 开发器” 的各种功能。这些 App 收集在名为 Applications 的文件夹中，并且可用于许多附加产品。您可以编辑这些 App，并以此为起点设计您自己的 App，也可以从中获取灵感。每个 App 都包含一个描述该 App 的文档 (PDF)，以及用于生成报告的选项。

下面列出一部分可用的 App 示例，其顺序与 “案例库” 树中的显示顺序一致。

名称	案例库
承受动载荷的梁	COMSOL Multiphysics
螺旋型静态混合器	COMSOL Multiphysics
传输线计算器	COMSOL Multiphysics
管式反应器	COMSOL Multiphysics
音叉	COMSOL Multiphysics
等效非线性磁曲线	AC/DC 模块
钢坯感应加热	AC/DC 模块
磁法勘探	AC/DC 模块
触摸屏模拟器	AC/DC 模块
吸收式消声器设计器	声学模块
声反射分析器	声学模块
家庭住宅声学分析仪	声学模块
管风琴音管设计	声学模块、管道流模块 [12]
小型音乐厅分析器	声学模块
循环伏安法	电化学模块、电镀模块、电池与燃料电池模块、腐蚀模块
电化学阻抗谱	电化学模块、电镀模块、电池与燃料电池模块、腐蚀模块
锂离子电池阻抗	电池与燃料电池模块 [1]
喷墨打印机	CFD 模块、微流体模块
NACA 机翼优化	CFD 模块 [8]
水处理池	CFD 模块
生物传感器设计	化学反应工程模块
液相色谱法	化学反应工程模块
膜透析	化学反应工程模块

名称	案例库
船体外加电流阴极保护	腐蚀模块
框架疲劳寿命	疲劳模块 [11]
参数化混凝土梁	岩土力学模块 [11]
同心管式换热器	传热模块 [2]
翅片式散热器	传热模块
周期性微结构的等效属性	传热模块
激光闪射法	传热模块
散热器强制空气冷却	传热模块
管线感应加热器	传热模块 [9]
阳伞和太阳辐射	传热模块
热电冷却器	传热模块
吸湿膨胀引起的 MEMS 压力传感器漂移	MEMS 模块、结构力学模块
微电阻梁	MEMS 模块
红细胞分离	微流体模块、粒子追踪模块 [3]
搅拌器	搅拌器模块 [4]
电荷交换池模拟器	分子流模块、粒子追踪模块 [10]
离子注入机计算器	分子流模块
离心调速器	多体动力学模块 [11]
卡车吊机分析器	多体动力学模块 [5,11]
压力容器应力分析	非线性结构力学模块 [11]
层流静态粒子混合器设计器	粒子追踪模块
地热泵	管道流模块
CCP 反应器	等离子体模块
分布式布拉格反射镜滤波器	射线光学模块
波纹状圆形喇叭天线	RF 模块
频率选择表面模拟器	RF 模块
微带贴片天线阵列合成器	RF 模块
表面等离激元线光栅	RF 模块、波动光学模块
波长可调 LED	半导体模块
梁截面计算器	结构力学模块 [6]
车架分析器	结构力学模块 [13]
过盈配合	结构力学模块

名称	案例库
桁架桥设计器	结构力学模块
桁架塔屈曲	结构力学模块
粘弹性结构阻尼器	结构力学模块
光纤模拟器	波动光学模块
表面等离激元线光栅	RF 模块、波动光学模块 [7]

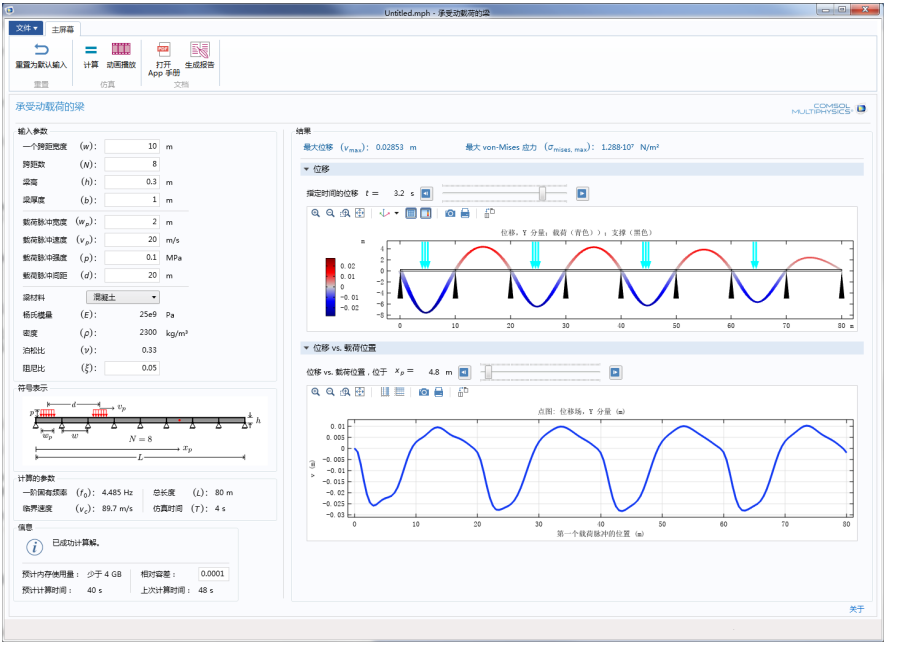
[1] 需要 “电池与燃料电池模块”和 “优化模块”。
[2] 运行时需结合使用 “CFD 模块”、“传热模块”、“微流体模块”或 “等离子体模块”。
[3] 在 “微流体模块”版本中，此 App 需要 “微流体模块”和 “粒子追踪模块”。在 “粒子追踪模块”版本中，此 App 需要 “粒子追踪模块”以及 “CFD 模块”、“微流体模块”或 “地下水流模块”。
[4] 需要 “CFD 模块”和 “搅拌器模块”。
[5] 需要 “结构力学模块”和 “多体动力学模块”。
[6] 提供此 App 的扩展版本，还需要 LiveLink™ for Excel® 产品。
[7] 在 “RF 模块”版本中，此 App 需要 “RF 模块”。在 “波动光学模块”中，此 App 需要 “波动光学模块”。
[8] 需要 “CFD 模块”和 “优化模块”。
[9] 需要 “传热模块”和 “AC/DC 模块”。
[10] 需要 “分子流模块”和 “粒子追踪模块”。
[11] 还需要 “结构力学模块”。
[12] 需要 “声学模块”和 “管道流模块”。
[13] 要获取完整的参数化功能，此 App 还需要 LiveLink™ for SOLIDWORKS®。

下面各节集中介绍了上表中列出的示例 App。这些着重介绍的 App 举例说明了 “App 开发器”中的各种重要特征，包括如何使用动画、电子邮件、优化功能、参数估计、表格以及如何导入实验数据。

承受动载荷的梁

此 App 模拟放置在多个等距支架上的梁在承受动载荷时的瞬态响应。其目的是分析当车辆通过桥梁时桥梁所产生的响应。据观察，在桥梁具有给定几何与材料属性的情况下，某些车速会在桥上引起共振，此时桥梁会发生高振幅振荡。在此 App 中，我们在假设二维平面应力近似的基础上对梁进行建模。梁由混凝土制成。

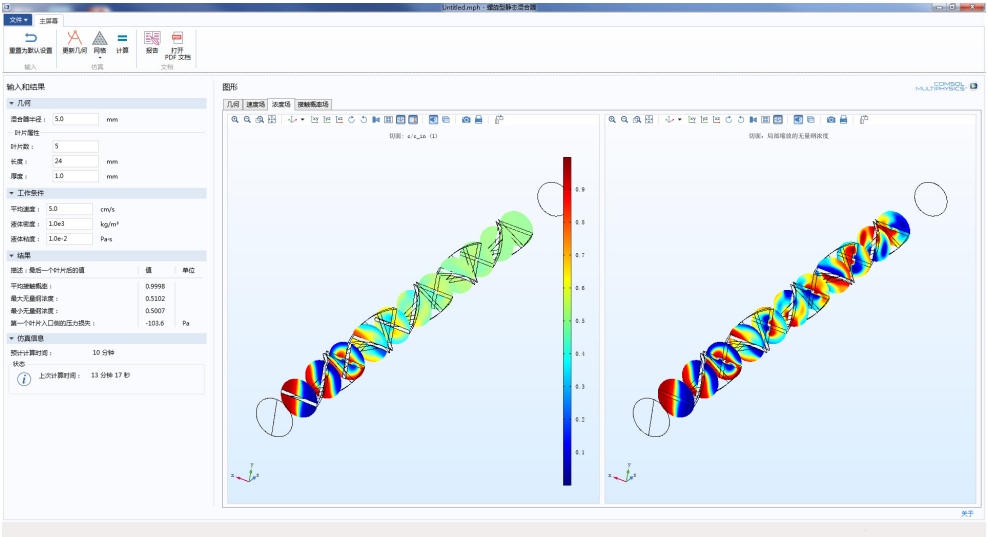
该 App 演示如何使用动画和滑块。第一个滑块显示变形相对于载荷位置的变化，第二个滑块显示位移随时间的变化情况。此 App 不需要任何附加产品。



螺旋型静态混合器

此 App 用于演示如何使用几何零件与参数化几何。此外，它还可用于估计包含一到五个螺旋叶片的系统中的混合程度，通常用于聚合反应中单体和引发剂的混合。该 App 仅限于牛顿液体，在混合器本身的聚合作用忽略不计的情况下，

可以得到很好的近似效果。该 App 演示如何使用**平铺或分页**类型的表单集合。此 App 不需要任何附加产品。



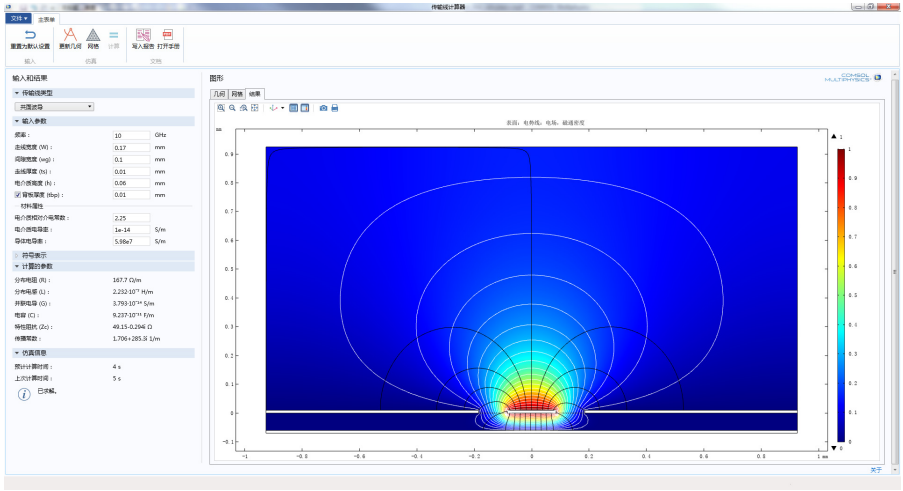
传输线计算器

传输线是一种用于在射频下引导交流电和电压波的波导结构。传输线的形式多种多样，并且大部分都易于加工，常用于印制电路板设计。它们是大多数现代电子设备中的关键元素，用于在尽可能降低损耗和失真的前提下，在同一个设备中或两个设备之间将信息从一个位置传输到另一个位置。

此 App 提供预定义的用户界面，用于计算传输线参数 R、 L、 G 和 C，以及某些常见传输线类型的参数化横截面的 γ 和 Z_0：

- 同轴线
- 双导线
- 微带线
- 共面波导 (CPW)

其中还提供几何、网格、电势、电场线和磁通量线的绘图。此 App 不需要任何附加产品。



管式反应器

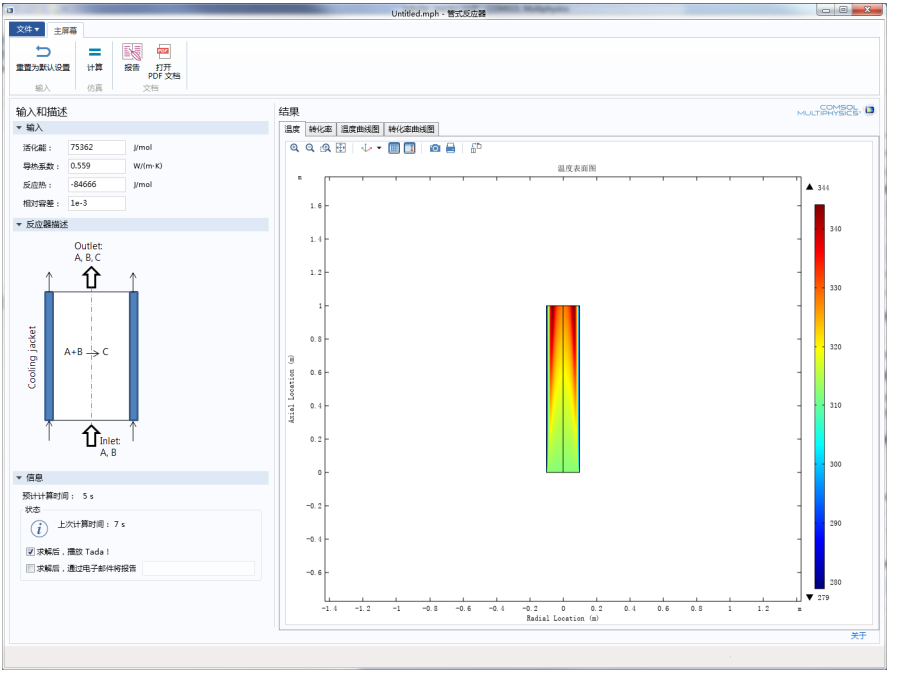
化学工程专业的学生可以借助此 App 对非理想管式反应器进行建模，包括温度和成分的径向和轴向变化，并研究不同工作条件所产生的影响。该 App 描述的是环氧丙烷与水发生放热反应形成丙二醇的过程。

其中还举例说明了教师如何构建定制的界面，来解决那些挑战学生想象力的问题。 App 中的模型和练习源自 Scott Fogler 编写的 *Elements of Chemical Reaction Engineering* 一书。

该数学模型包含轴对称坐标系中描述的能量平衡和物料平衡。学生可以更改反应器中的反应活化能、导热系数和反应热。得到的解呈现了反应器中的轴向和径向转换以及温度曲线。对于某些数据，仿真结果并不直观，这意味着对模型结果进行解释也成为了练习解决问题的途径。

请注意，在计算完成后，您还可以通过 App 发送电子邮件，只需选中“电子邮件”复选框并输入电子邮件地址即可。此时会发送一份包含当前设置和计算结果的报告。学生可以使用该功能将结果发送给导师。对于耗时较长的计算，此功能非常有用。例如，您可以在启动仿真后离开办公室或实验室，然后在计算

完成时从 App 获取完整报告，您可以在路上访问报告，也可以在能访问电子邮件的任何地方查看报告。此 App 不需要任何附加产品。



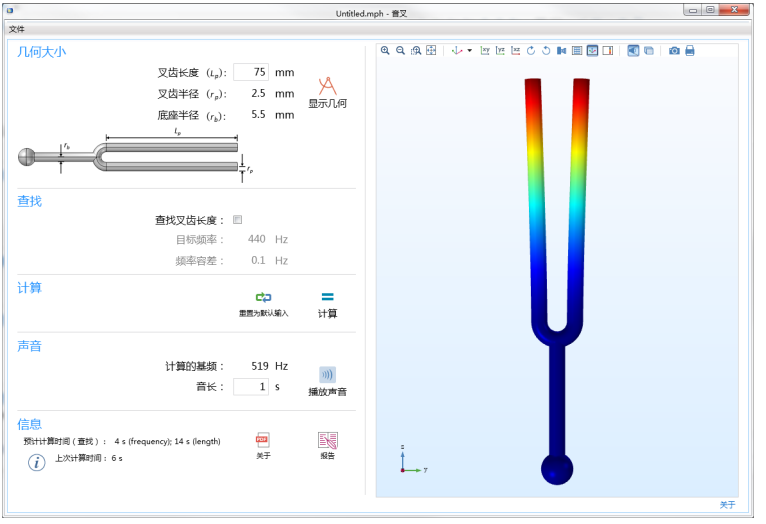
音叉

此 App 用于计算具有用户定义叉齿长度的音叉的谐振频率。或者，您也可以给出用户定义的目标频率， App 将找到相应的叉齿长度。叉齿和手柄半径取自一个商用音叉。

App 中嵌入的模型是使用 COMSOL Multiphysics 中包含的 “固体力学” 接口定义的，无需任何附加产品。叉齿长度搜索算法是一种正割法。

在计算结束时，会使用内置方法 `playSound` 基于计算频率产生正弦波声音。

有关更多背景理论知识，请参见 tuning_fork.mph 模型的 “案例库” 文档。

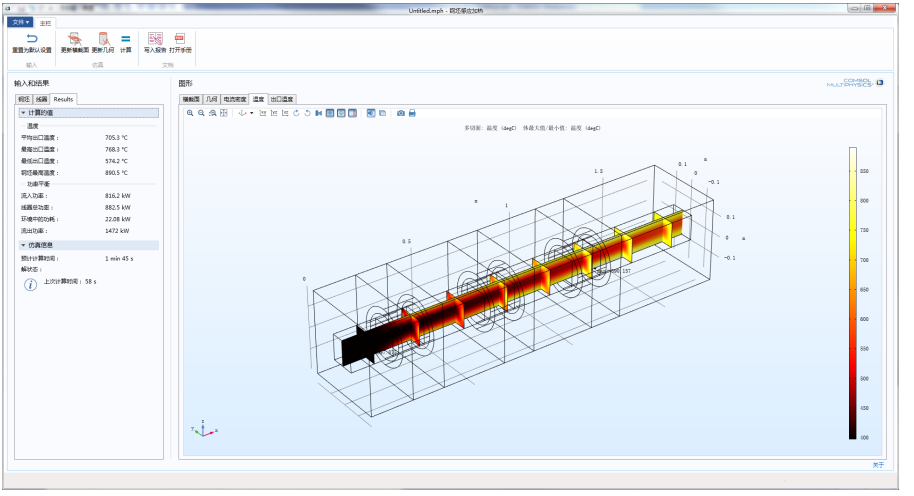


钢坯感应加热

此 App 可用于设计简单的钢坯感应加热系统，该系统由一个或多个电磁线圈组成，钢坯在其中做匀速运动。这些线圈通入交流电后会在金属坯中感应出涡流，从而通过焦耳热效应产生热量。

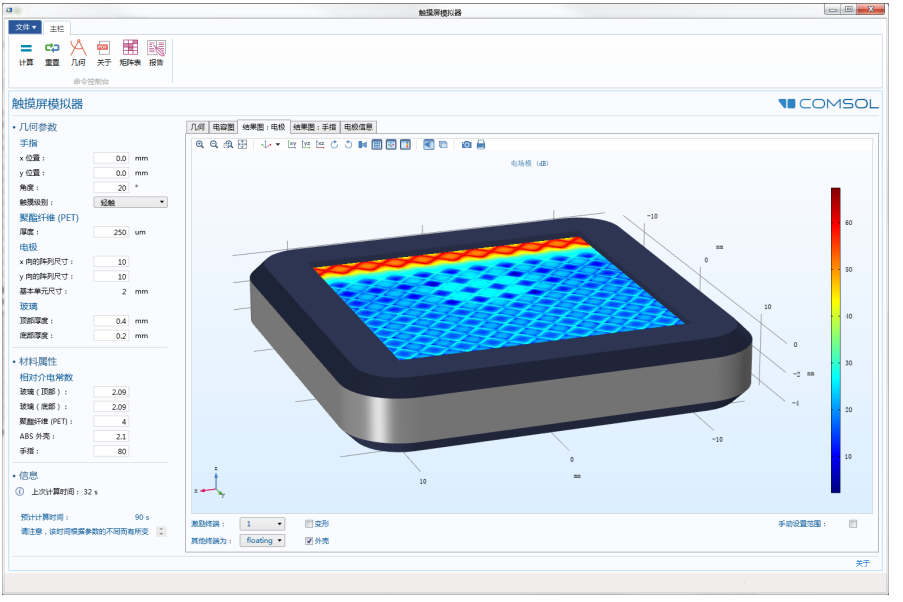
钢坯横截面、线圈数、位置、尺寸、初始温度和环境温度以及单个线圈电流都可以指定为输入。在计算出解之后，该 App 会显示钢坯加工过程中温度的三维

绘图、感应电流密度，以及出口横截面温度的二维绘图。最后， App 还会计算钢坯预期温度范围和系统功率平衡的数值数据。



触摸屏模拟器

此 App 用于计算当手指触摸一个小触摸屏时，该触摸屏的电容响应。电子电路可以使用此信息来推断手指的位置。在该 App 中，手指的位置和方向通过输入参数进行控制，并计算产生的电容矩阵作为输出。

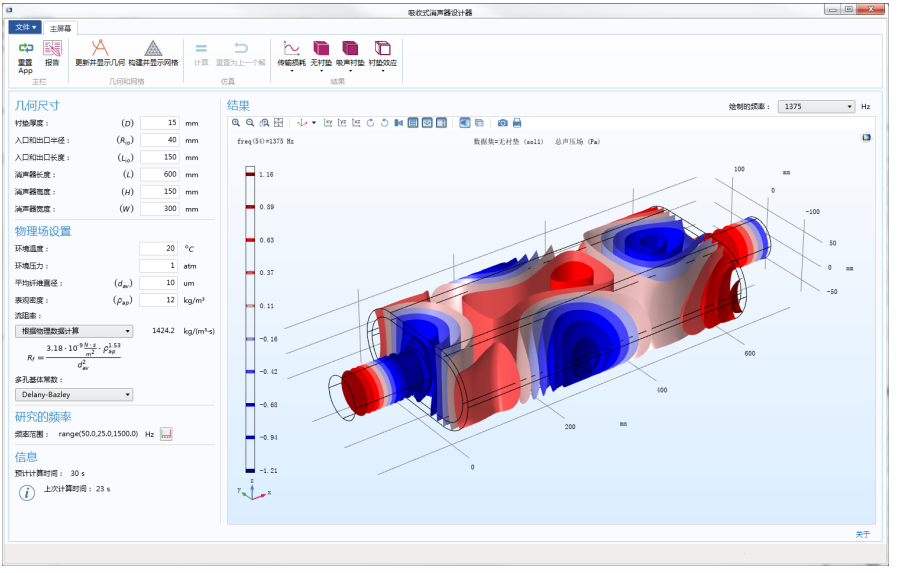


吸收式消声器设计器

此 App 用于研究和设计带多孔衬里的简单共振式消声器。消声器用于衰减内燃机或暖通空调系统等发出的噪声，并且通常在特定的频率范围内表现良好。衰减测量称为传输损耗 (TL)，这种测量会给出衰减 （以 dB 计）随频率的变化关系。传输损耗取决于消声器的几何形状以及可放置在系统中的多孔纤维材料的特性。此 App 用于研究修改消声器尺寸所产生的结果；环境工作条件以及多孔衬垫的材料属性 - 即各种变化因素对系统传输损耗的影响。

它是一个给定消声器模型的 “动态明细表” 示例。销售工程师可以将此类 App 带到客户面前，向他们展示为他们量身定制的消声器的性能。例如，可以设计

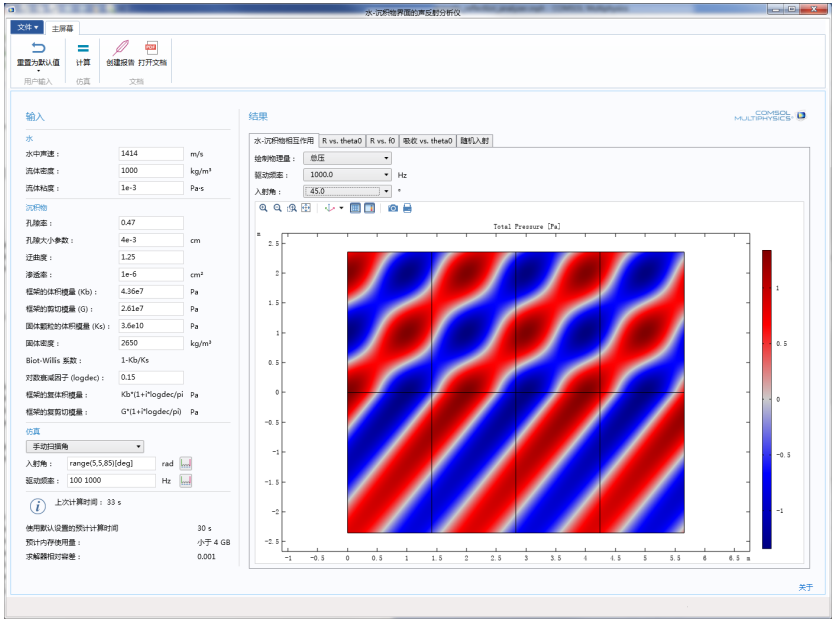
一种消声器用于安装在受空间约束的车辆内。在这种情况下，您可以与客户一起直观地体验它的性能，并审查各种不同的选项。



声反射分析器

此 App 用于分析水 - 沉积物界面上的平面波反射。反射和吸收系数作为入射角和频率的函数来确定。此外，随机入射吸声系数或扩散场吸声系数基于模拟数

据进行计算。流体 （本例中为水）的材料属性和多孔介质 （此处为半无限沉积层）的属性均可修改。



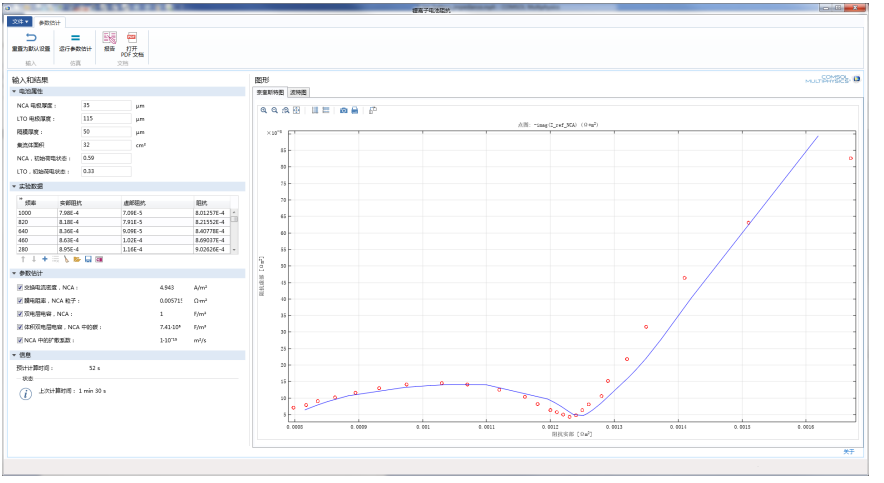
锂离子电池阻抗

此 App 用于解释实验电化学阻抗谱 (EIS) 测量方法，并演示如何使用模型和测量数据来评估锂离子电池的性能。

其中采用 EIS 测量得到的实验数据作为输入，并对这些测量结果进行模拟，然后基于实验数据运行参数估计。

控制参数包括交换电流密度、粒子电阻层的电阻率、NCA 的双电层电容以及正极中碳载体的双电层电容。对测得的正极阻抗进行拟合时采用的频率范围是 10 mHz 到 1 kHz。

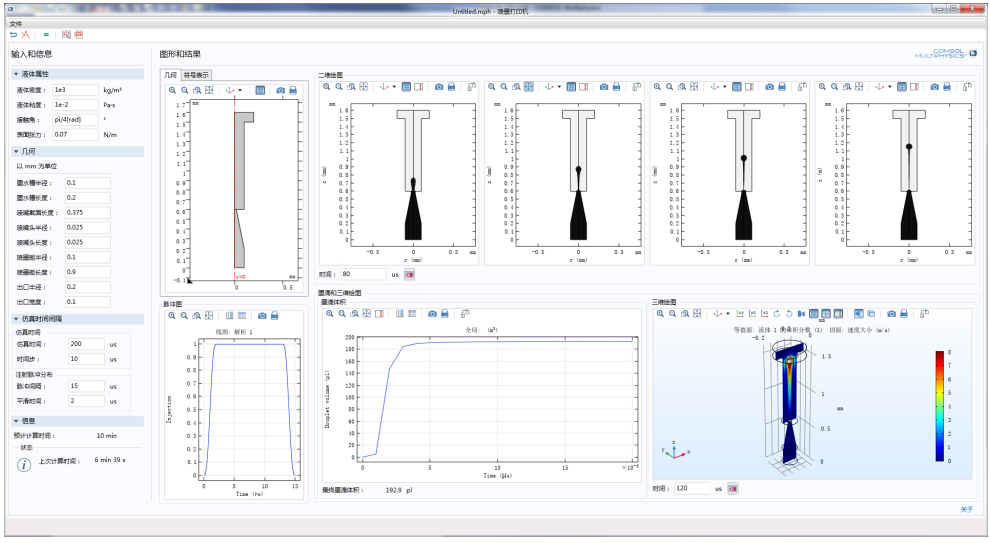
该 App 演示如何加载逗号分隔值 (CSV) 文件格式的实验数据，并利用“优化模块”进行参数估计。



喷墨打印机

此 App 演示如何调整喷墨打印机喷嘴的形状和操作以达到所需的墨滴大小，具体取决于注入液体的接触角度、表面张力、粘度和密度。

结果还表明，注入的体积是否会先分解成多个墨滴，然后在基板上进行合并形成最终墨滴。

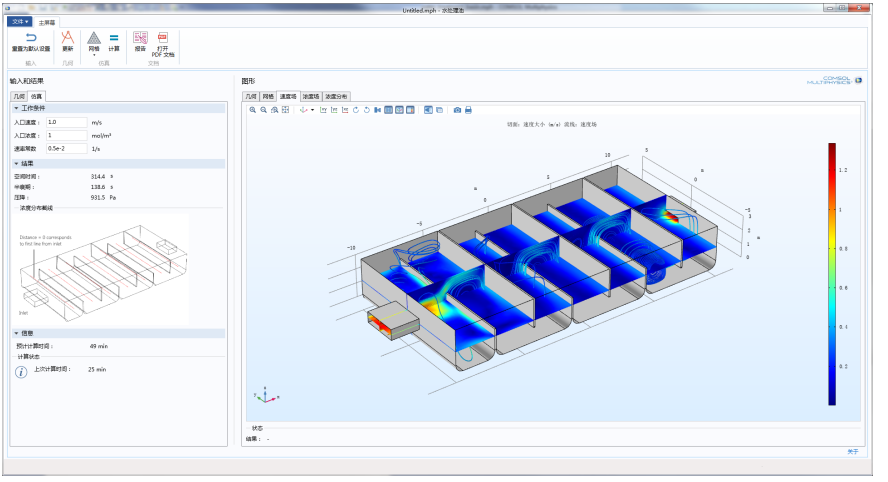


水处理池

“水处理池” App 举例说明如何利用此类 App 在三维模式下为液体中的湍流建模。非常有趣的一点是，它还分析了溶液中溶质的物质平衡。溶质反应是一级反应，这是一种常见的反应类型，用来描述高度稀释的化学物质的减少过程。该 App 还显示如何使用完全参数化的几何与累积选择为湍流建模。

您可以基于它来设计自己的 App，用于模拟液体中的稳定湍流以及高度稀释溶质的反应过程。

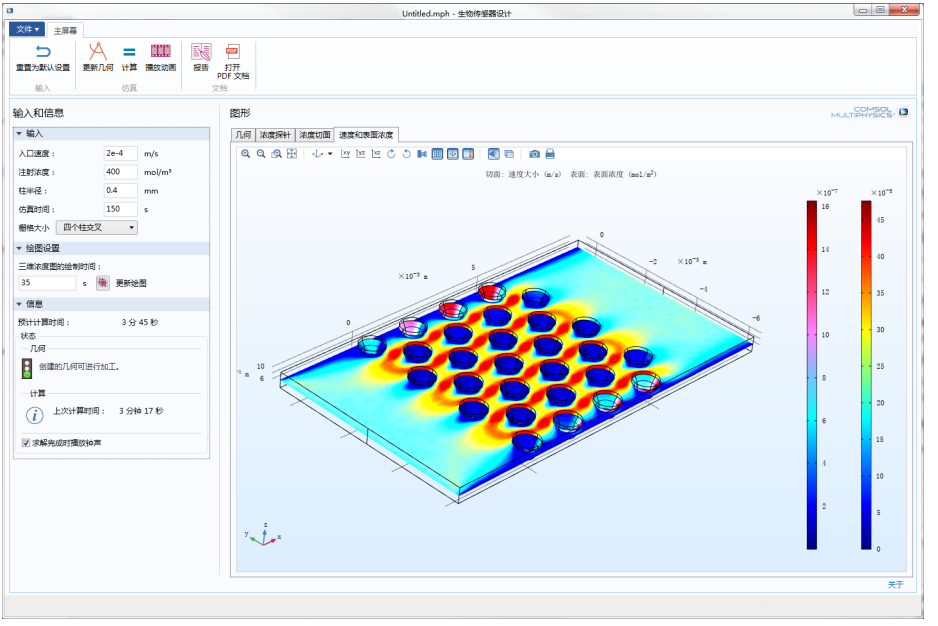
示例系统是水处理工艺中的加氯消毒池。



生物传感器设计

生物传感器中的流通池包含用于检测生物分子的微柱阵列。这些微柱上涂有一种活性物质，可在样品流中选择性地吸附生物分子。这些生物分子随后会在表面发生反应。此 App 支持用户通过改变柱径、栅格间距和入口速度等参数来更改传感器的设计，研究设计对检测结果的影响。几何结构与工作条件对信号强度和扩散有很大的影响。

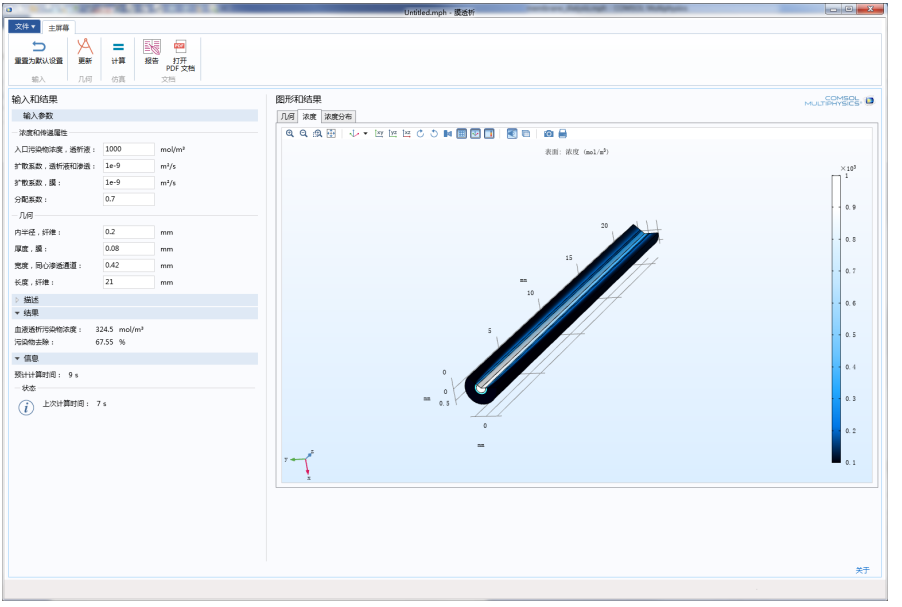
此外， App 还会报告由微柱的最小间距设置的制造约束。



膜透析

此 App 模拟在膜透析装置中净化的血液中的污染物浓度。模拟的透析装置由中空纤维透析器制成，中空纤维壁用作去除污染物的透析膜。透析液在纤维内流

动，而渗透液则在外面流过。通过改变输入参数， App 可以检测出在装置内最大程度地去除污染物的方法。

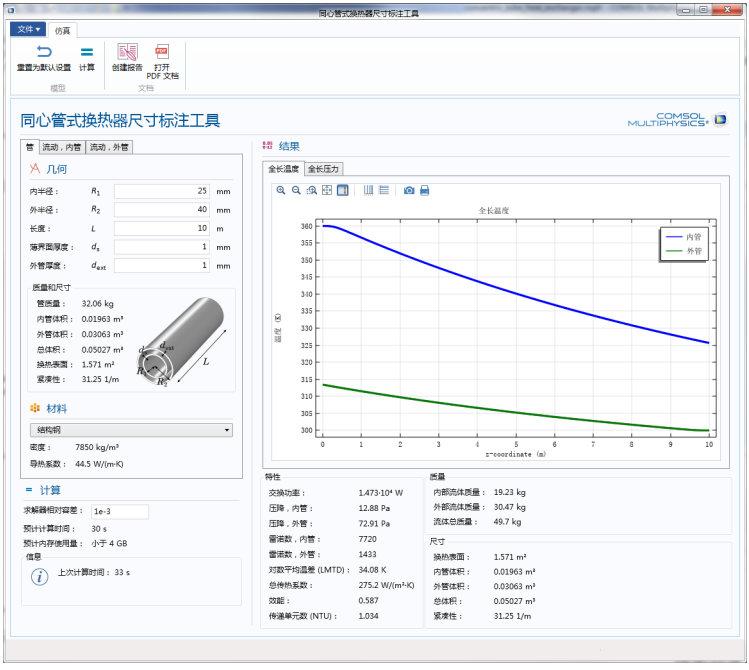


同心管式换热器

尺寸物理量是换热器性能的首要指标。此 App 用于针对给定的配置计算这些物理量。

本例研究使用两个同心管分离两种不同流体的情况。这两种流体可以反向流动，也可以平行流动。管和流体都可以通过用户界面进行定制。

计算完成后，可以显示温度曲线图和多个物理量。其他输入还包括流体属性，例如，可用的体积和质量、紧密度（交换表面与换热器体积之比）和材料属性。

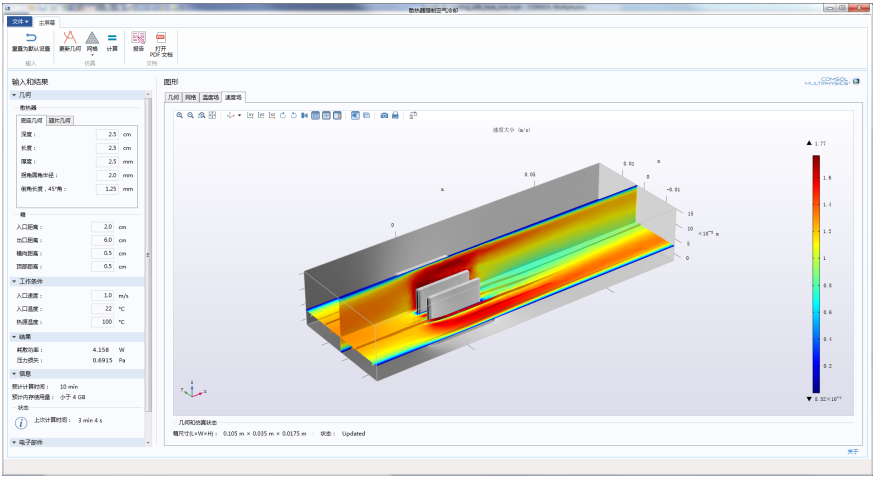


翅片式散热器

散热器通常以针对给定风扇曲线的散热能力作为性能评估基准。要进行这类实验，一个可行的方式是将散热器放置在带绝热壁的矩形通道中。然后测量通道入口和出口的温度和压力，以及将散热器底座保持在给定温度所需的功率。在这些条件下，可以估计散热器的散热量与通道压力损失。

该 App 的目的是通过建模和仿真对基准实验进行研究。例如，散热量会随着翅片数的增加而增大，直到翅片多到对气体流动产生很大的阻碍，造成流动逐渐

缓慢，从而降低散热量。这意味着在给定通道总压力损失的情况下，可以设置最佳的尺寸和翅片数，使冷却功率达到最高。此 App 可供您执行此类研究。



周期性微结构的等效属性

我们经常可以在碳纤维和蜂窝结构等复合材料中发现周期性微结构。它们可以表示为沿三个传播方向重复的基本单元。为了降低计算成本，仿真可以将复合材料的所有细节用具有等效属性的均匀域来代替。

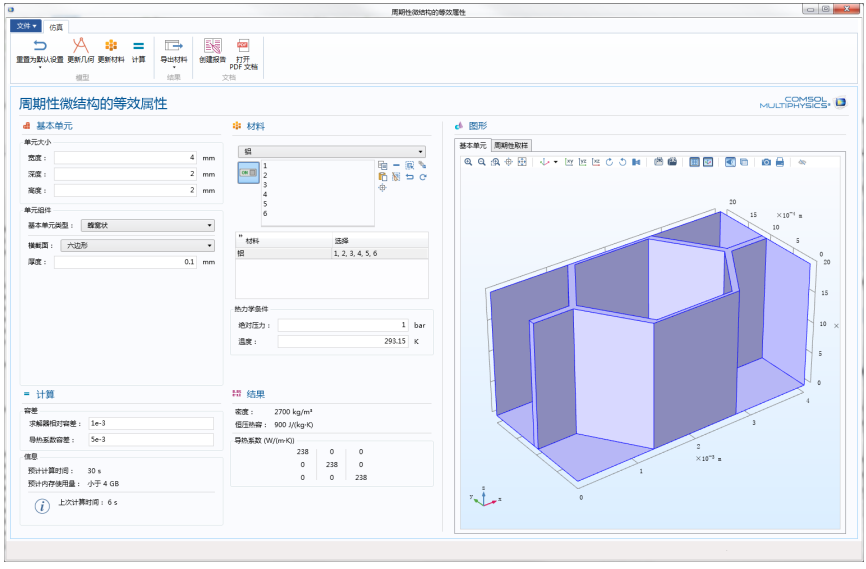
此 App 计算基本单元的几何构型和材料属性的等效属性。其中提供九个可参数化的基本单元和包含 13 个预定义材料的列表可供选择。您可以在现有基础上轻松扩展该 App，在其中添加平行六面体基本单元或其他材料。

在热方程等扩散方程中，等效扩散系数采用张量的一般形式。

在此 App 中，基于基本单元的不同区域中具有给定材料的选定基本单元形状计算了以下材料属性：

- 密度
- 恒压热容
- 热导率

此 App 中内置的基本单元库包含多种广泛使用的单元类型，例如平行叠层、纤维增强复合材料或蜂窝结构。一旦完成几何设置，物理场便会包含单元相反边界的周期性热条件。

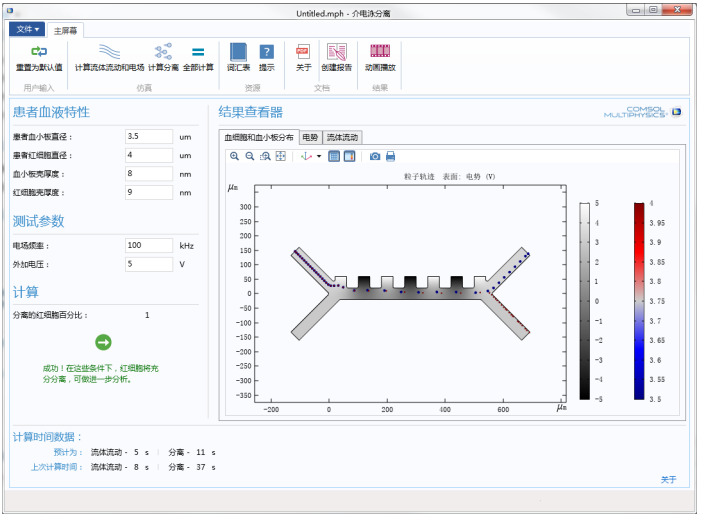


红细胞分离

介电泳 (DEP) 是介电粒子在非均匀电场中受力的现象。粒子中的电场会产生极化，使粒子受到与电势梯度成正比的介电泳力。

介电泳力对粒子的大小、形状和介电特性很敏感，使得介电泳可以用来分离不同种类的粒子。这一过程可以应用到生物工程领域，用来分离混合物中不同种类的细胞。此 App 示例显示如何从血样中选择性地筛选红细胞，从而将红细胞

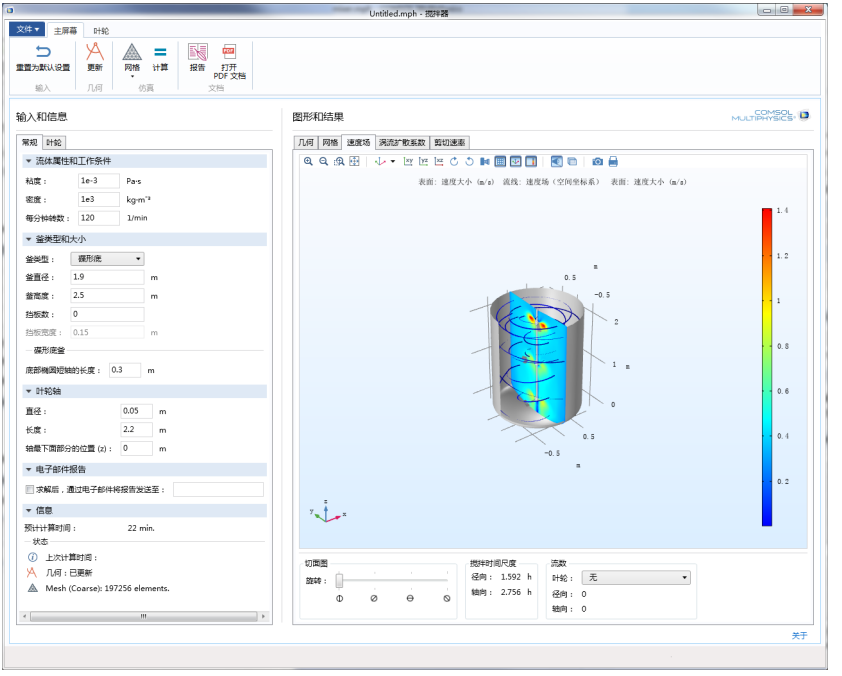
与血小板分离。这一过程非常有用，因为血小板会引起血液凝结，而一旦形成血凝块，就可能导致血小板污染的样本不适用于后续测试。



搅拌器

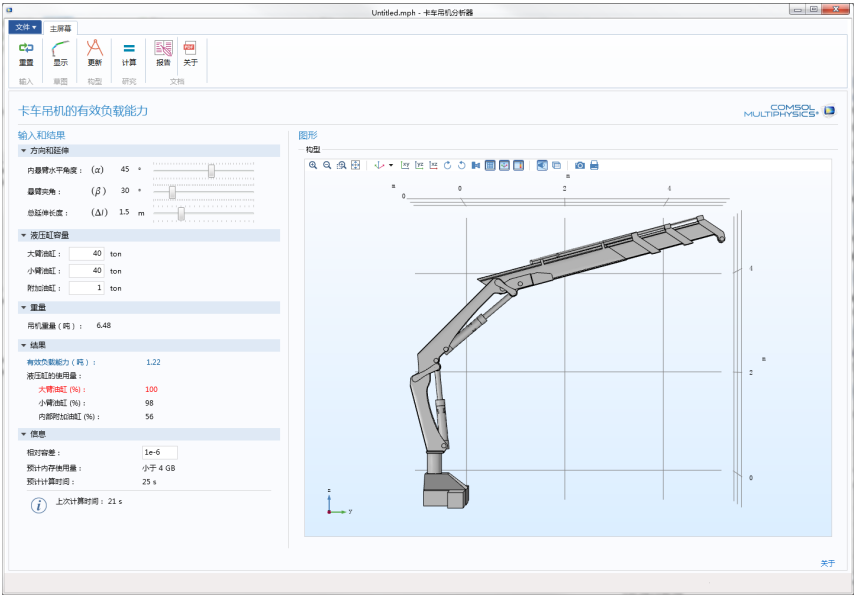
“搅拌器” App 的目的是提供一个用户友好的界面，供科研人员、流程设计者和工艺工程师研究容器、叶轮和工作条件对驱动叶轮所需的混合效率和功率的影响。您可以使用该 App 来理解在给定流体的情况下，搅拌器的设计和操作，并进行优化。但最重要的是，您可以在此基础上构建您自己的 App，从而对搅拌器和反应器进行建模和仿真。

该 App 演示如何使用零件和累积选择来自动设置嵌入模型中的域和边界。即使 App 用户所做的选择是用于创建各种不同的几何，也可以自动创建这些设置。



卡车吊机分析器

许多卡车都装有吊机，用来装卸货物。这种吊机有许多液压缸，用来控制吊机的运动和多个机构。

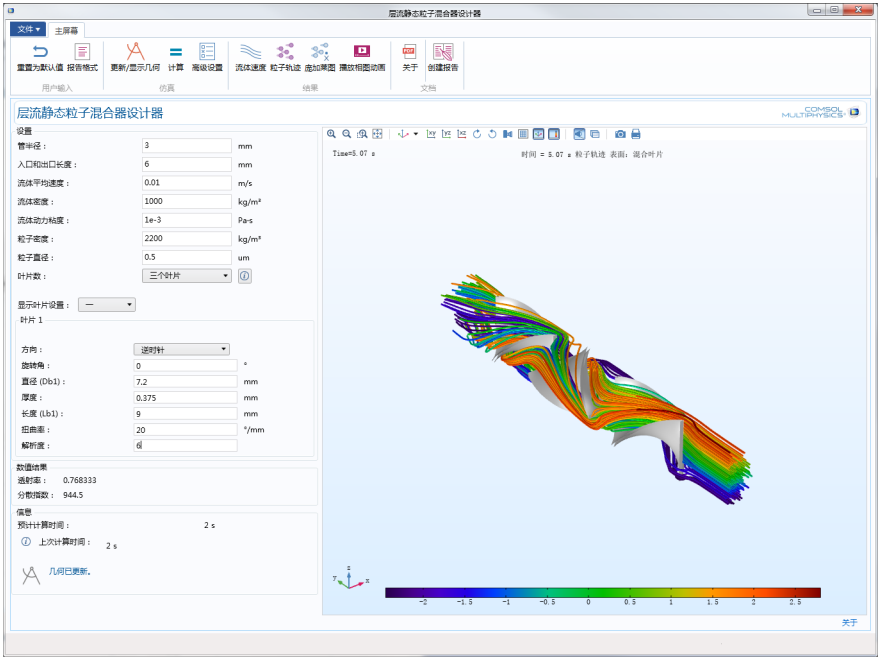


在此 App 中，通过对吊机执行刚体分析，来确定吊机指定方向和范围的有效负载能力。其中还提供液压缸的使用率，并突出显示达到使用极限的液压缸。您可以修改液压缸的容量，以提高其有效负载能力和使用率。

层流静态粒子混合器设计器

静态混合器也称为静止混合器或管道混合器，其原理是将流体泵入一个含固定叶片的管道以达到混合的目的。在层流流态下，这种混合技术产生的压力损失非常小，因此特别适用于层流混合。此 App 研究扭曲叶片静态混合器中的流体流动，通过计算悬浮颗粒经过混合器的轨迹来评估混合性能。该 App 计算了在

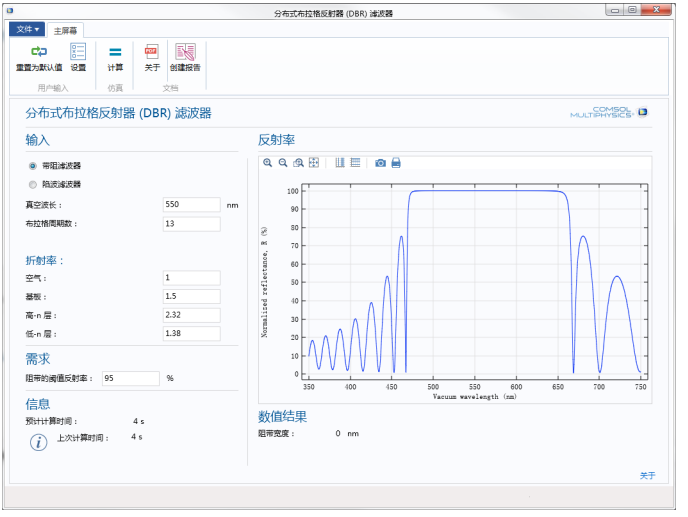
室温条件下，一种物质在溶剂中溶解的静态混合效果。您可以研究流体和颗粒属性以及固定叶片结构对颗粒混合产生的影响。



分布式布拉格反射镜滤波器

分布式布拉格反射镜 (DBR) 由两种材料的交替层组成。每种材料都有不同的折射率，导致在与 DBR 层的垂直方向出现高、低折射率的重复模式。当光通过这种结构传播时，会在每个层间界面上发生反射。多次反射波之间的干涉效应会导致 DBR 的反射率与波长高度相关。与普通金属镜相比， DBR 的主要优势在于它可以设计为在所选波长上具有定制的反射率。

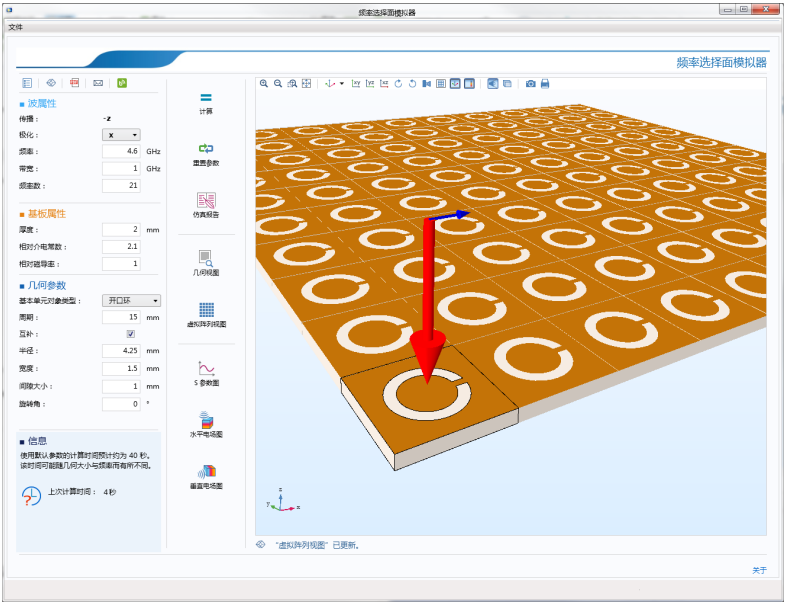
此 App 可用于基于两个分布式布拉格反射镜之间的腔体来研究简单光学陷波滤波器的反射率。



频率选择表面模拟器

频率选择表面 (FSS) 是一种产生带通或带阻频率响应的周期性结构。此 App 模拟从内置基本单元类型中选择的用户指定的周期性结构。其中提供五种常见的 FSS 基本单元类型，它们具有两个预定义的极化方向，并沿法向入射方向传播。

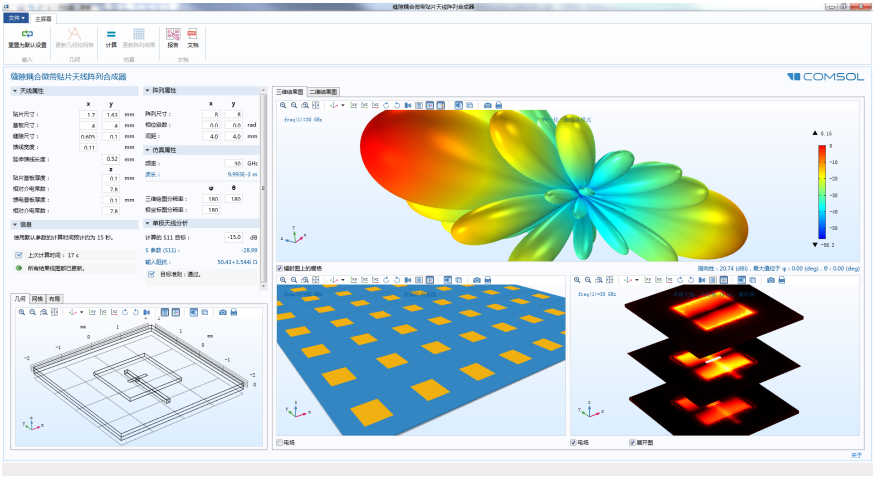
分析内容包括反射谱和透射谱、基本单元顶面的电场模，以及基本单元域中垂直截面上显示的 dB 尺度的电场模。



微带贴片天线阵列合成器

此 App 模拟一个在多层低温共烧陶瓷 (LTCC) 基板上制作的缝隙耦合微带贴片天线。仿真结果包括天线阵列的远场辐射方向图及其方向性。远场辐射方向图是通过将阵列因子与单个天线辐射方向图相乘来近似得到的，从而在无需模拟复杂的全阵列模型的情况下执行有效的远场分析。使用默认的输入频率 (30

GHz) 可以轻松评估 5G 移动网络的相控天线阵原型。该 App 还演示通过相机在天线周围移动生成的动画。



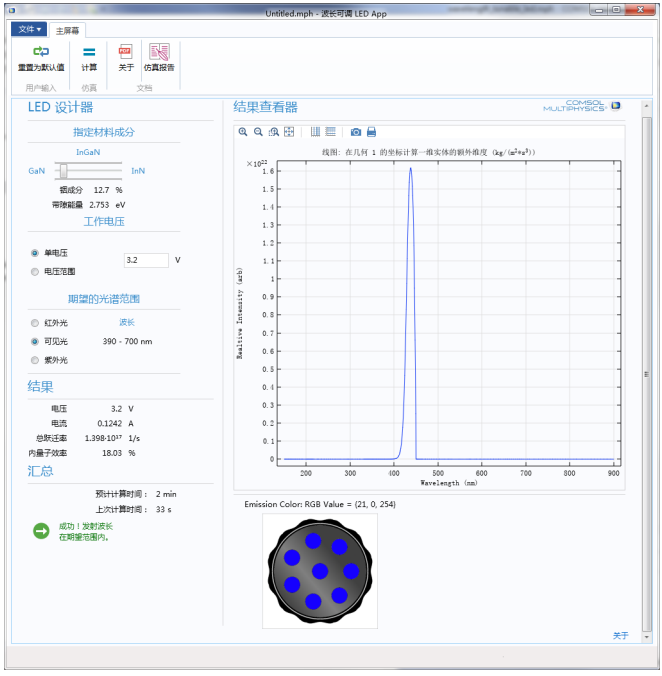
波长可调 LED

近年来，蓝光 LED 备受关注，其原因在于它们能促进现代高效照明技术的发展。由于氮化镓的带隙能量大，因此被广泛用于产生蓝光。此 App 模拟一个基于氮化镓的发光二极管的发光特性。

这种设备的活性区使用的材料是 InxGa1−xN，其中含有镓和铟混合物，铟的分数由 x 给出。该光学活性区的带隙可以通过改变铟分数从而改变材料的成分来进行控制。由于纯 InN 和 GaN 分别在光谱范围的红外和紫外部分发射，因此可以利用这一技术在整个可见光谱范围内调整 InxGa1−xN 的发射能量。

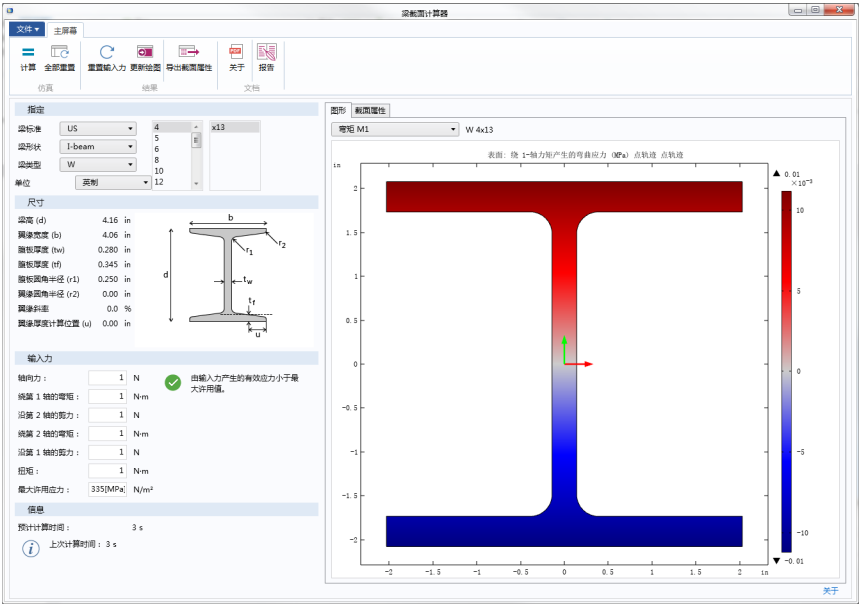
此 App 支持对设备的铟分数和工作电压进行控制。从而计算其中的电流、发射强度、电致发光光谱和内量子效率。您可以输入单工作电压，也可以输入一系

列电压。如果输入一个电压范围，还可以计算电流电压曲线，从而确定设备的接通电压。此 App 使用了多种不同的方法。



梁截面计算器

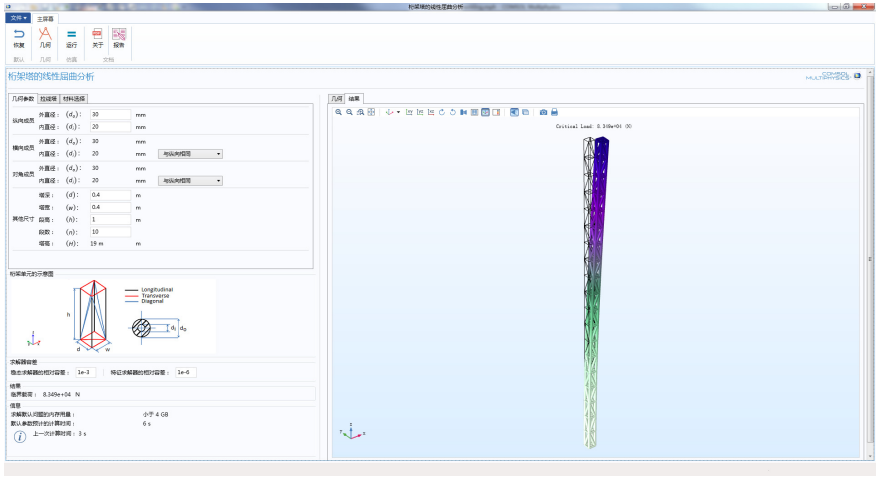
此 App 计算特定钢梁截面的截面属性。它还可用于计算当横截面受到一组作用力和力矩时，其中的详细应力分布。其中提供大量的美国和欧洲标准梁。如果您拥有 LiveLink™ *for* Excel® 产品许可证，则所有输入和结果数据都会显示在一个表格中，您可以将其导出至 Excel® 文件。您可以编辑包含梁尺寸数据的 Excel® 工作簿，并将这些数据重新导入 App。



桁架塔屈曲

屈曲分析用于确定临界压缩载荷，一旦超出这一限值，结构就会变得不稳定。此 App 可以模拟桁架塔在垂直压缩载荷作用下的屈曲。这一桁架塔可以选择性地通过拉线支撑。该 App 主要用于计算和分析塔在不同几何条件下的屈曲载

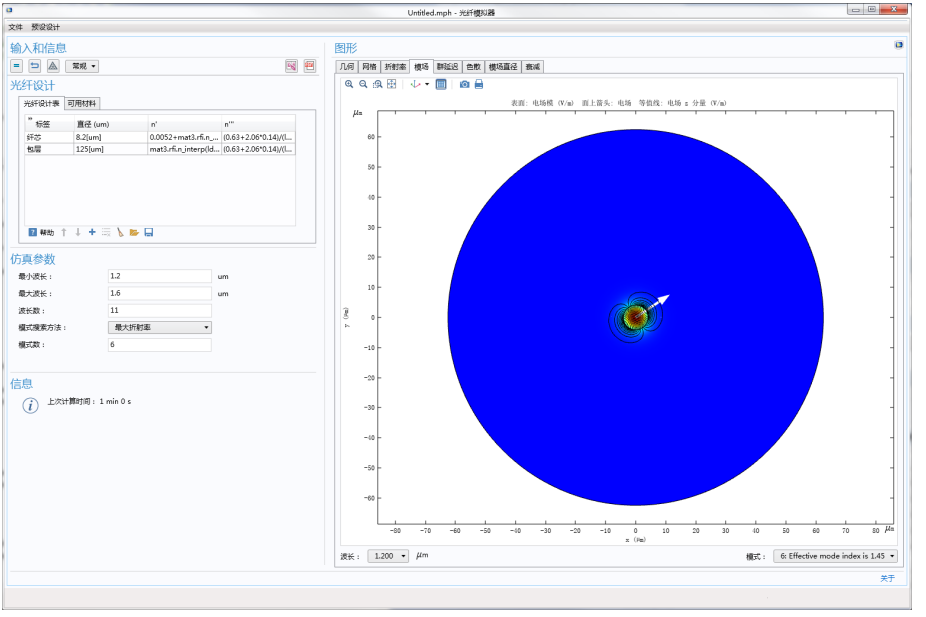
荷，即各种塔高、横截面积以及不同的材料。该 App 在计算过程中加入了恒载效应（桁架和支撑拉线的自重及其预张力）。

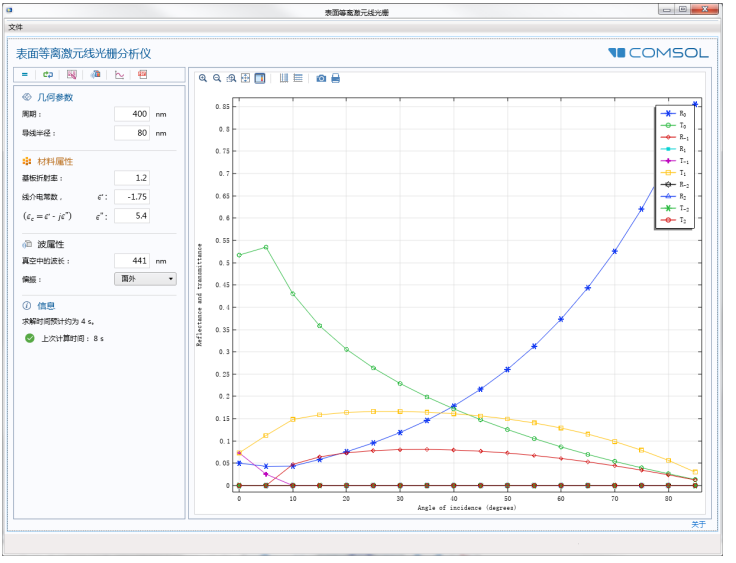


光纤模拟器

该 App 用于对同心圆介电层结构执行模态分析。每层都通过外径以及折射率的实部和虚部进行描述，折射率表达式可以包含对波长和径向距离的依赖性。因

此，该模拟器可用于分析阶跃折射率光纤和梯度折射率光纤。这些光纤可以具有任意数量的同心圆层。计算结果包含群延迟和色散系数。



表面等离激元线光栅

此 App 计算透射波和反射波的衍射效率 (m = 0) 以及第一和第二衍射级 （m = ±1 和 ±2）与介质基片上线光栅入射角的函数关系。平面波的入射角从正入射扫描至掠入射。该 App 还显示某个选定入射角的多个光栅周期的电场模图。