

App 开发器

简介

5.3

COMSOL

App 开发器简介

© 1998–2017 COMSOL 版权所有

受 cn.comsol.com/patents 中列出的美国专利和美国专利 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 和 9,454,625 的保护。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和复制。

COMSOL、 COMSOL 徽标、 COMSOL Multiphysics、 Capture the Concept、 COMSOL Desktop、 LiveLink 和 COMSOL Server 为 COMSOL AB 公司的注册商标或商标。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，亦不为其正式认可、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 5.3

联系信息

请访问 “联系 COMSOL” 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及电话号码。您还可以访问全球销售办事处页面 cn.comsol.com/contact/offices，以获取地址和联系信息。

如需联系技术支持，可以访问 COMSOL Access 页面并填写在线申请表，位于：cn.comsol.com/support/case。其他有用链接还包括：

- 技术支持中心：cn.comsol.com/support
- 产品下载：cn.comsol.com/support/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 7

App 开发器桌面环境 10

App 开发器和模型开发器 17

参数、变量和范围 18

运行 App 20

在 COMSOL Multiphysics 中运行 App 20

在 COMSOL Server 中运行 App 28

发布 COMSOL App 33

App 开发器入门 35

表单编辑器 40

表单设置窗口 40

单独的表单设置窗口 41

表单编辑器首选项 43

表单对象 44

表单编辑器中的编辑器工具 49

按钮 51

图形 63

输入框 79

单位 85

文本标签 85

数据显示 86

表单编辑器中的数据访问 89

草图和栅格布局 95

App 之间进行复制 109

主窗口 111

菜单条和工具栏 113
功能区 117
事件 118
启动和关闭时的事件 119
全局事件 120
表单和表单对象事件 123
使用局部方法 124
声明 125
标量 128
一维数组 131
二维数组 133
选择列表 135
文件 136
单位集 137
快捷方式 142
图形数据 144
方法编辑器 148
转换命令序列为方法 148
语言单元窗口 153
方法编辑器中的编辑器工具 153
方法编辑器中的数据访问 156
录制代码 158
检查语法 161
查找和替换 162
模型表达式窗口 163
使用快捷方式 164
语法高亮、代码折叠和缩进 165
方法编辑器首选项 167

Ctrl+Space 和 Tab 实现代码完备 168
创建局部变量 170
局部方法 171
模型方法 173
带输入输出变元的方法 177
调试 179
停止方法 181
模型对象 181
语言单元示例 181
库. 186
图像. 186
声音 189
文件 191
附录 A— 表单对象 192
所有表单对象列表 192
切换按钮 193
复选框 196
组合框 200
方程 220
线 221
网页 222
图像 223
视频 223
进度条 225
日志 228
信息日志 229
结果表 230
表单 233

表单集合 234
卡片堆 236
文件导入 240
信息卡片堆 244
数组输入 247
单选按钮 251
选择输入 253
文本 257
列表框 258
表格 262
滑块 267
超链接 269
工具栏 270
定距片 271
附录 B — App 之间进行复制 273
附录 C — 文件处理和文件方案语法 275
COMSOL Server 文件处理 275
文件方案语法 278
文件导入 280
文件导出 288
附录 D — 键盘快捷键 295
附录 E — 内建方法库 297
附录 F— 开发 App 的准则 314
附录 G—App 库示例 317

前言

仿真软件的用户通常是那些拥有硕士或博士学位、具有多年建模与仿真经验且接受过全面培训的专业人士。他们通常是来自大型机构研发部门的科学家，或者是学术研究者。这是因为仿真理论非常复杂，并且典型的仿真软件都提供许多选项，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，一小部分专家在为一个很大的群体服务，这个群体可能是产品开发、生产领域的工作人员，也可能是学习物理场效应的学生。仿真模型通常非常复杂，能够实现建模的只有那些可以安全提供输入数据来获得有用输出结果的人。因此，使用计算机进行建模和仿真是产品开发、生产和教育中的瓶颈。

为了使一小群人能够为很大的用户群体服务， App 开发器提供了解决方案。借助 App 开发器，仿真专家可以创建适用于自己的直观且特定的用户界面，或者通用的计算模型，它是一个现成可用的 App。通用模型可以作为几个不同 App 的起点，每个 App 向用户提供只针对手头特定任务的输入和输出选项。 App 包含用户文档、“在允许范围内”检查以及单击按钮时生成预定义报告。

创建 App 通常需要以下领域专家的协同合作：物理场、数值分析、编程、用户界面设计以及图形设计。

在一定程度上， COMSOL 的技术支持团队可以为您的 App 物理场和数值分析设置提供建议。此外， COMSOL 文档和在线资源也能够提供很大帮助。对于编程和设计，技术支持团队能够提供的帮助很有限。对这些领域来说，您自己的开发工作至关重要。

App 开发器便于团队创建技术良好的 App，能够避免意外的用户输入错误，同时专注于相关的输出详情。

COMSOL 确信这是在世界范围内传播成功使用仿真的方式，我们全力致力于实现这一目标。

简介

COMSOL® App 通过高度定制的用户自定义界面与 COMSOL Multiphysics® 模型以直观和有效的方式进行交互，本书给出了 App 开发器桌面环境的快速概览，其中附有显示如何使用表单编辑器和方法编辑器的示例。本书还提供参考材料，说明可供使用的内置方法和函数的详细列表。如果您希望了解如何使用模型开发器的信息，请查阅手册 《COMSOL Multiphysics 简介》。



如果您希望在阅读本书前先查看 App 示例，可以从 App 库的任意一个 Applications 文件夹中打开和运行其中的 App。当阅读本书的同时保持打开状态，从而可以进行尝试。只有 Applications 文件夹中包含带有用户界面的 App，其他文件夹放置的是不包含用户界面的教学模型。

App 开发器仅为 Windows® 平台版本的 COMSOL Multiphysics 提供，可以通过 COMSOL Desktop® 环境调用。 COMSOL Multiphysics 及其附加产品用于创建 App，当从 COMSOL Multiphysics 或 COMSOL Server TM 运行它时需要拥有与开发者相同的附加模块。

更多资源，包括视频教程，可以在线查阅：cn.comsol.com。

通过 COMSOL MULTIPHYSICS 运行 App

当拥有 COMSOL Multiphysics 许可证时，可以通过 Windows®、 macOS 以及 Linux® 平台上的 COMSOL Desktop 来运行 App。

通过 COMSOL SERVER 运行 App

当拥有 COMSOL Server 许可证时，可以通过在 Windows®、 macOS、 iOS、 Linux® 以及 Android™ 平台上的主流网页浏览器来运行网页形式的 App。在 Windows® 上，您还可以通过一个免费安装的 COMSOL Client 连接到 COMSOL Server 后运行 App，请从 cn.comsol.com 下载安装程序。COMSOL Server 不包含 COMSOL Desktop 中提供的 App 开发器、物理场开发器或模型开发器工具。

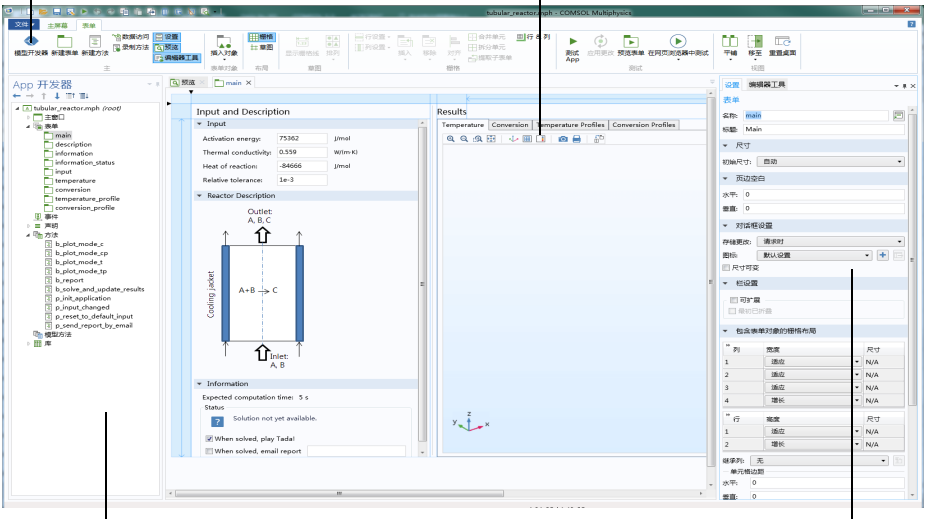
开发 App 的准则

如果您并不擅长开发图形化用户界面或编程，您可能需要阅读第 302 页的 “附录 F— 开发 App 的准则”。

App 开发器桌面环境

模型开发器和 App 开发器 - 通过单击这个按钮在模型开发器和 App 开发器之间进行切换。

COMSOL DESKTOP 环境 - COMSOL Desktop 环境提供了 App 开发器的使用环境，包括表单和方法编辑器，以及模型开发器。



App 开发器窗口 - 包含 App 树的 App 开发器窗口。

设置窗口 - 单击 App 树中任意一个节点，包括表单编辑器和方法编辑器，可以看到对应的设置窗口。

上面的截屏表示当您使用 App 开发器时将看到的操作界面。 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 37 页）
- 方法编辑器 （参见第 132 页）

App 树

App 树包含以下这些节点：

- **主窗口**
- **表单**
- **事件**
- **声明**
- **方法**
- **模型方法**
- **库**



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

表单节点，包含表单子节点。其中每个表单可能包含一些表单对象，例如，输入框、图形对象，以及按钮等。

事件节点包含全局事件的子节点。这些子节点为修改各种不同的数据实体 （例如，全局参数或者字符串变量等）时触发的所有事件。全局事件还可以与 App 的启动和关闭操作关联起来。

声明节点用来声明全局变量，可以用作模型中定义的全局参数和变量。

方法节点包含 App 方法子节点。其中包含模型开发器的模型树节点的运行命令之外的操作代码。例如，方法可以执行循环、处理输入和输出，以及向 App 用户发送信息和警告。 App 方法修改的是正在运行的 App 的模型对象，而不是当前会话中由 “模型开发器” 表示的模型对象。

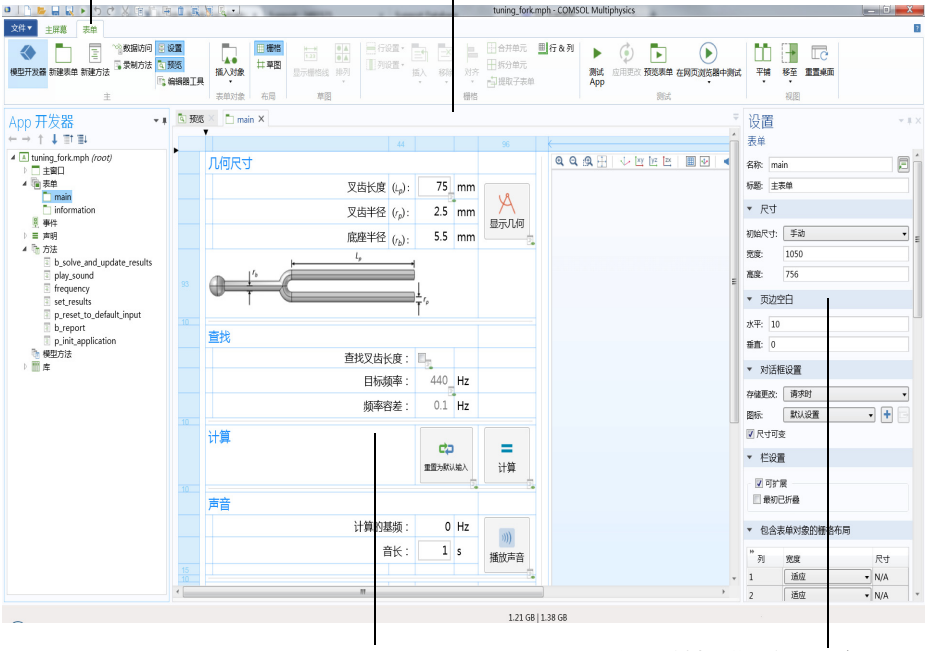
模型方法节点类似于**方法**节点。**方法**节点包含的 App 方法修改正在运行的 App 的模型对象，而**模型方法** 节点包含的模型方法直接修改当前会话中由 “模型开发器” 表示的模型对象。模型方法可用于将由多个手动步骤组成的建模任务自动化。

库节点包含嵌入到 MPH 文件的图片、声音，以及文件，因此您不需要将这些文件另外随 App 单独分发。此外，库节点还包含 Java® 实用类节点，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以快速地访问表单编辑器。

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曳来移动对象，单击对象来编辑其设置。



表单对象 - 每个表单包含表单对象，例如输入框、复选框、图片、按钮，以及其他更多控件。

设置窗口 - 单击 App 树中的节点或表单对象，可以看到对应的设置窗口。

使用表单编辑器来开发用户界面布局，通过表单对象来创建表单，例如，输入框、图片、按钮，等。

表单编辑器的主要组成部分：

- 表单功能区选项卡
- 包含 App 树的 App 开发器窗口
- 表单窗口
- 编辑器工具窗口
- 设置窗口

创建新表单

如果要创建一个新表单，请右键单击 App 树的**表单**节点，选择**新建**表单。您还可以单击功能区中的**新建表单**。创建新表单将自动打开**新建表单**向导。

如果您的 App 中已经有了一个表单，例如 form1，然后您想要编辑它，可以通过以下两种方法来打开表单编辑器：

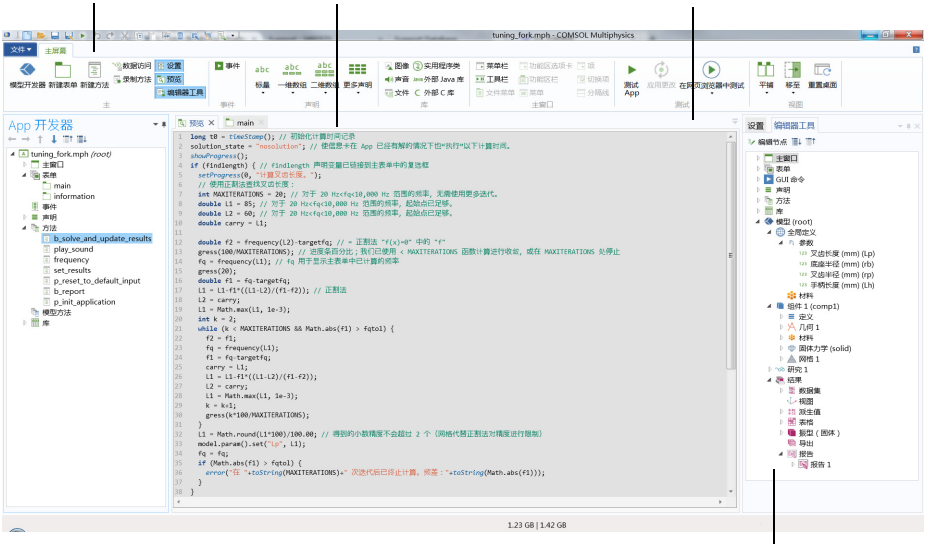
- 在 App 树中，双击 **form1** 节点。
- 在 App 树中，右键单击 **form1** 节点，然后选择**编辑**。

方法编辑器

方法选项卡 - 功能区中的方法选项卡可以快速访问易用的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之间进行切换。

设置窗口 - 单击任意一个 App 树节点可以看到对应的设置窗口。



模型表达式、语言单元，以及模型代码窗口 - 这些窗口显示了编写代码的工具，模型表达式窗口显示所有的模型中可以使用的常数、参数、变量以及函数。语言单元窗口用来插入内置的方法模板。模型代码窗口用来编辑和运行模型树节点的提取代码。

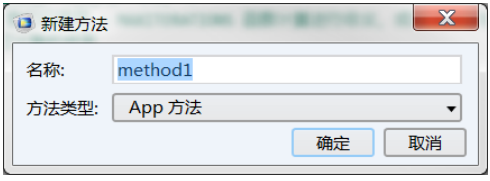
使用方法编辑器来编写没有被标准的模型树节点用法所涵盖的操作的方法，它相当于其他的编程语言中的子程序、函数、或者过程。

方法编辑器的主要组成部分：

- 方法功能区选项卡
- 包含 App 树的 App 开发器窗口
- 方法窗口
- 模型表达式、语言单元、编辑器工具和设置窗口 （这些窗口在上图中堆叠在一起）

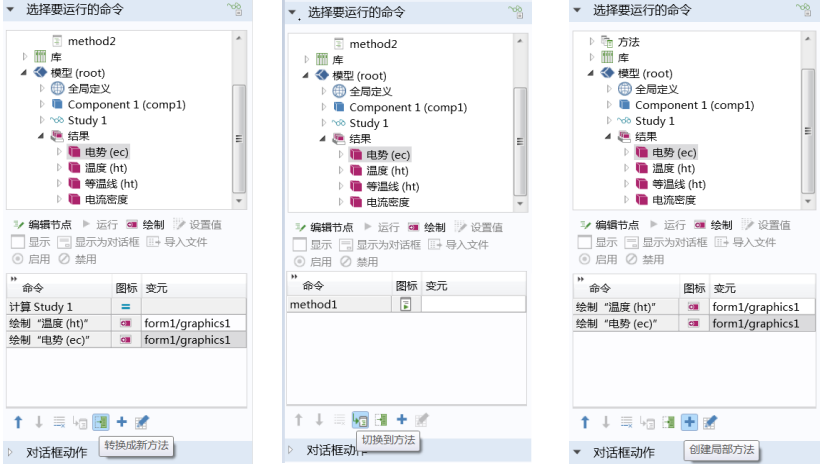
创建新方法

如果想要创建新方法，请右键单击 App 树中的**方法**节点，然后选择**新建方法**。您还可以单击功能区中的**新建方法**。在**新建方法**对话框中，将**方法类型**设为**App 方法**或**模型方法**。



创建一个新方法将自动打开方法编辑器。以这种方式创建的 App 方法属于全局性的方法，可以从所有 App 方法和表单对象中调用。模型方法是全局性的方法，可以从其他所有方法和 “模型开发器” 功能区的 “开发工具” 选项卡中调用。

与一个按钮或菜单关联的命令序列可以自动地通过单击**转换成新方法**来转换成一个新的 App 方法，然后通过单击**切换到方法**来打开新建的方法。您还可以通过单击**创建局部方法**来创建一个表单对象的局部方法。以上这些操作见下列这些截图。



如果方法已经存在，例如 `method1`，您可以使用以下这些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，然后选择**编辑**。
- 在表单对象或事件的设置窗口的命令序列下单击**切换到方法**。

App 开发器首选项

要访问 **App 开发器**的**首选项**，请从**文件**菜单中选择**首选项**，然后选择 **App 开发器**页面。



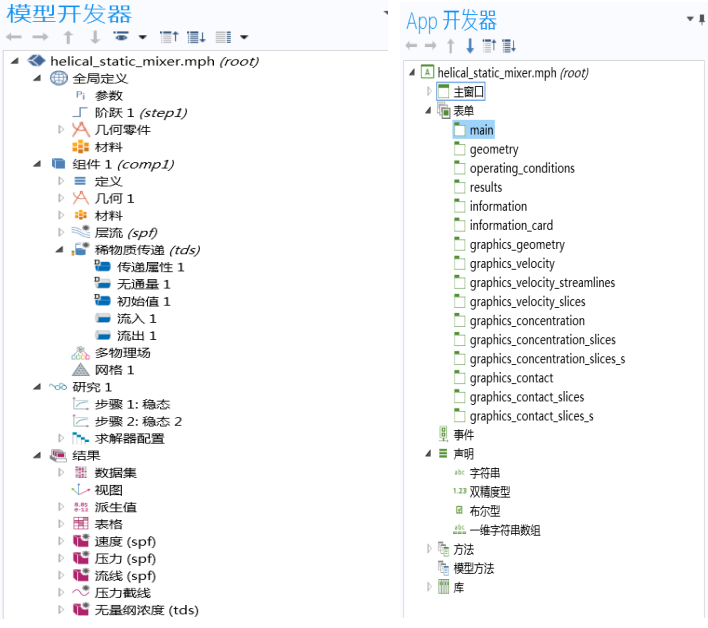
您可以配置 COMSOL 桌面环境，使 App 开发器显示在单独的桌面窗口中。选中 **App 开发器使用单独的桌面窗口**复选框。

请注意，您可以分别使用键盘快捷键 Ctrl+Alt+M 和 Ctrl+Alt+A 在模型开发器和 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**关闭前的最大编辑者数**复选框并编辑数字 （默认值为 15）。设为较低数值可加快包含大量表单的 App 的加载。

App 开发器和模型开发器

使用 App 开发器，基于模型开发器开发的模型来创建 App。 App 开发器提供了两个重要的工具用于创建 App：表单编辑器和方法编辑器。此外， App 可以包含菜单栏或功能区。表单编辑器包含拖放用户界面零件的功能，例如，输入框、图形对象，以及按钮等。方法编辑器是一个编程环境，允许您修改表征不同模型部件的数据结构。下图显示了模型开发器和 App 开发器窗口。



当创建一个 App 时，您通常需要从一个已有的模型出发。然而，您也可以同时开发 App 用户界面和底层模型。您可以在任意时刻很方便地在模型开发器和 App 开发器之间切换。 App 的模型部分，以模型树形式表征 ，被称做内嵌模型。

App 开发器中的工具可以通过多种方式访问和执行内嵌模型的设置。例如：

- 如果模型使用参数和变量，您可以通过新建表单向导或编辑器工具将它们直接链接到输入框。以这种方式， App 的用户可以直接编辑影响模型的参数和变量。请查阅第 45 和 67 页，了解更多信息。

- 通过使用新建表单向导或编辑器工具，您可以在 App 中包含运行研究节点的按钮，用来启动求解器。此外，您可以使用这个向导引入图形、数值输出、复选框，以及组合框，等等。请查阅第 31 和 45 页，了解更多信息。
- 数据访问工具和编辑器工具窗口可以用来使用表单对象或方法来直接访问模型中的底层设置。请查阅第 45、 77 和 128 页，了解更多信息。
- 通过使用记录代码工具，您可以将您在模型树及其节点中执行操作时的命令记录在方法中，后续可以进一步编辑。请查阅第 132 页，了解更多信息。

参数、变量和范围

模型树可能同时包含用于控制模型设置的参数和变量。下图显示了一个 App 的模型树，其中包含**参数**和**变量**节点。



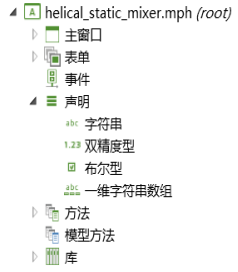
参数是在模型树的**全局定义**节点下定义，一般是用户定义的常数，可以用于整个模型开发器。也就是说，它们是 “全局”性质。重要的应用包括：

- 参数化几何尺寸
- 指定网格单元尺寸
- 定义参数化扫描

变量既可以在**全局定义**节点下定义，也可以在任意模型**组件**节点的**定义**子节点中定义。一个全局定义的变量可以用于整个模型，而模型组件中定义的变量只能用于该组件。变量可以使用随空间或时间变化的表达式，可以包括正在求解的因变量。

在模型开发器中，一个参数或变量是一个字符串，其额外的限制是它的值必须是一个有效的模型表达式。如果要了解更多关于模型中的参数和变量的用法相关的信息，请查阅 《COMSOL Multiphysics 简介》。

一个 App 可能需要更多额外的变量用于表单编辑器和方法编辑器，这些变量通常在 App 开发器的 App 树的**声明**节点下声明。下图显示了一个包含多条声明的 App 树。



App 开发器中的声明变量通常分为几类，包括标量、数组、布尔变量、字符串、整数，以及双精度数。在使用变量前，您必须声明它的类型。

事实上，这些分类的变量意味着它们可以直接在方法中使用，无需事先使用内建方法进行转换。比起表征模型表达式的参数和变量，这样就更容易书写代码。此外，在 App 开发器中也提供了一些转换这些不同类型变量的工具，详情可以查阅第 111 和 259 页。有关分类变量的更多信息，请参阅 *Application Programming Guide*。

运行 App

当拥有 COMSOL Multiphysics 许可证时，可以从 COMSOL Desktop 环境运行 App。当拥有 COMSOL Server 许可证时，可以从各种操作系统和硬件平台的主流网页浏览器中运行 App。此外，您可以使用 Windows® 系统上的 COMSOL Client 连接到服务器后运行 App。

下面的两个章节解释如何从 COMSOL Multiphysics 和 COMSOL Server 环境运行 App。第 32 页的第三部分 “发布 COMSOL App” 描述了您发布 App 的权限。

在 COMSOL Multiphysics 中运行 App

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

- 单击功能区或快速访问工具栏中的**测试 App**。
- 选择**文件**菜单或快速访问工具栏中的**运行 App**。
- 双击 Windows® 桌面上的 MPH 文件图标。
- 选择功能区中的**在网页浏览器中测试**。

测试 App

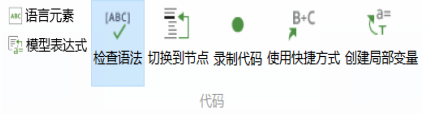
测试 App 用来快速地进行测试，它将打开一个单独的显示 App 用户界面的窗口，并保持 App 开发器桌面环境的运行。



当测试 App 时，您可以通过单击**应用修改**按钮实时地修改表单、方法以及内嵌的模型。不过并不是所有的修改都可以实时应用，在这种情况下，您会得到一个提示，要求您关闭 App 并重新单击**测试 App**。

如果要预览表单的布局且不需要运行 App，可以单击功能区中的**预览表单**。

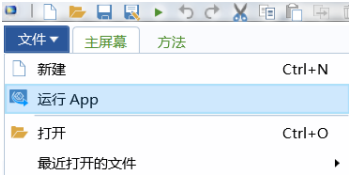
当使用**测试 App** 时，所有的方法都会使用内置的 Java® 编译器进行编译，只要有语法错误，就会生成错误信息，并且停止测试 App 进程。如果希望在测试 App 之前检查语法错误，可以单击**方法**选项卡中的**检查语法**按钮。



检查语法会使用内置的 Java® 编译器编译方法，查找语法错误。所有的语法错误将会显示在方法编辑器的**错误和警告**窗口中。请查阅第 105 页 “方法编辑器”来了解更多信息。

运行 App

运行 App 会在 COMSOL Desktop 环境中运行 App，选择**运行 App** 来观察目标 App 的运行情况。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，但是不限制您的运行。



双击 mph 文件

当您双击 Windows® 桌面上的 MPH 文件图标时，将在 COMSOL Multiphysics 中打开 App，表示 MPH 文件扩展名关联到 COMSOL Multiphysics。App 既可以打开进行编辑，也可以进行运行，您可以在 App 树的根节点控制其行为。这个节点的**设置**窗口有一个栏称为 App，您可以从中选择**编辑 App** 或者**运行 App**。对这个设置进行的修改将在保存 MPH 文件时得到应用。



选项**编辑 App** 将在 **App 开发器**中打开 App。

选项**运行 App** 将以实时运行模式打开 App，实现生产目的。这个选项相当于在**文件**菜单中选择**运行 App**，区别在于双击 MPH 文件将打开一个新的 COMSOL Multiphysics 会话。

如果您安装了 Windows® 系统的 COMSOL Client，那么 MPH 文件扩展名可能关联到了 COMSOL Client，这种情况下双击 MPH 文件将提示您登录到一个 COMSOL Server。

忽略许可证错误

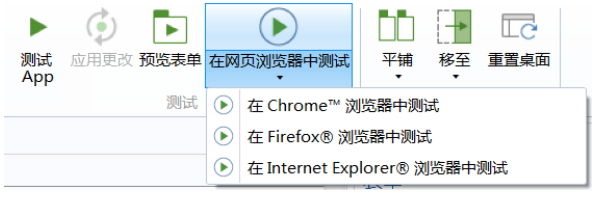
在 App 树根节点的**设置**窗口中，复选框**启动时忽略许可证错误**用于控制运行 App 时许可的产品的特征。



选中该复选框后，即使所有必需的许可证都不可用，也可以启动 App，但是无法使用未提供许可证的产品的功能。然而，可以编写方法来创建 App，使用到的功能动态地适应可用的许可证类型。

在网页浏览器中测试 App

在网页浏览器中测试用于在网页浏览器中测试 App，这个功能可以很方便地测试从网页浏览器连接到 COMSOL Server 上访问 App 时的外观和感觉。



您可以选择在哪一个已安装的网页浏览器中运行 App。**在网页浏览器中测试**将打开一个单独的浏览器窗口，其中包含 App 用户界面，与此同时 App 开发器桌面环境保持运行。

测试 App VS. 在网页浏览器中测试

测试 App 基于 Microsoft® .NET Framework 组件来启动 App，而**在网页浏览器中测试**则基于 HTML5 组件来启动带用户界面的 App。**测试 App** 将显示当使用

COMSOL Multiphysics 或 Windows® 系统的 COMSOL Client 连接到 COMSOL Server 时会看到的用户界面的外观。而**在网页浏览器中测试**则显示当通过网页浏览器连接到 COMSOL Server 时会显示的用户界面外观。

如果需要在 macOS、 iOS、 Linux®，以及 Android™ 平台上的网页浏览器中测试用户界面的外观和功能，需要安装 COMSOL Server。

下表总结了运行 App 的不同选项。

SERVER	CLIENT
COMSOL Multiphysics	测试 App
COMSOL Multiphysics	在网页浏览器中测试
COMSOL Multiphysics	运行 App
COMSOL Server	Windows® 平台上的 COMSOL 客户端
COMSOL Server	网页浏览器

Server 列表示执行繁重的 CPU 计算的软件组件， Client 列表示用来表现 App 的用户界面的软件组件。

保存运行中的 App

当您测试 App 时，会赋予它文件名称 Untitled.mph，它是原始 MPH 文件的一个副本，这不同于运行 App 的情况。

默认情况下，当退出 App 时，用户不会得到保存变化内容的提示。您可以从 App 树的根节点来控制这个行为。这个节点的**设置**窗口包含一个称为 App 的栏，您可以在其中选择**当关闭时询问保存 App** 复选框，见下图，



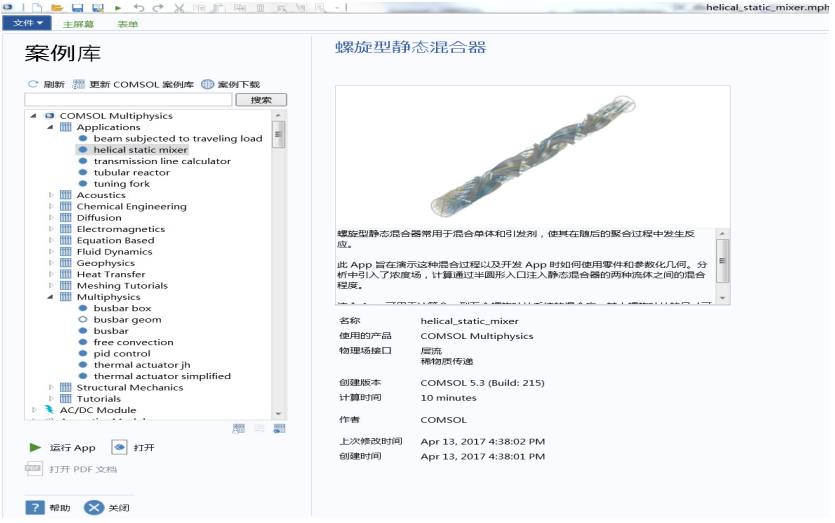
另一种方案是，添加一个包含保存 App 命令的按钮或者菜单。更多信息请查阅第 84 页。

案例库

从**文件**菜单，选择**案例库**来运行和浏览安装 COMSOL 时自带的示例 App。本书中的很多截屏都来自于这些示例。



您可以通过单击**案例库**树下方对应的按钮来运行 App，或者打开它来进行编辑。



只包含模型，但没有额外的表单或方法的 App 不能直接运行，只能打开用于编辑。包含表单和方法的 App 收录在名为 **Applications** 的文件夹。

案例库中的 App 将会持续改进和更新，您可以通过单击**更新 COMSOL 案例库**来进行更新。

还有一些不属于**案例库**的 App，可以从 COMSOL 官网的**案例下载**中找到它们。请单击**案例下载**按钮来查找，这将打开一个浏览器，其中显示案例下载网页。

每个 App 包含一个相关的预览图片，显示在**案例库**中。在 COMSOL Server 网页界面，预览图片显示在**案例库**页面。

以下操作可用于设置预览图片，单击 App 树的根节点，**设置**窗口包含两个选项：**从图形窗口设置**和**从文件加载**。

通过**从文件加载**选项，您可以加载 PNG 或 JPG 格式的图片文件，请选用图片大小为从 280X210 到 1024X768 像素的图片，确保图片可以正确地在 COMSOL Multiphysics 和 COMSOL Server 中显示为预览图片。

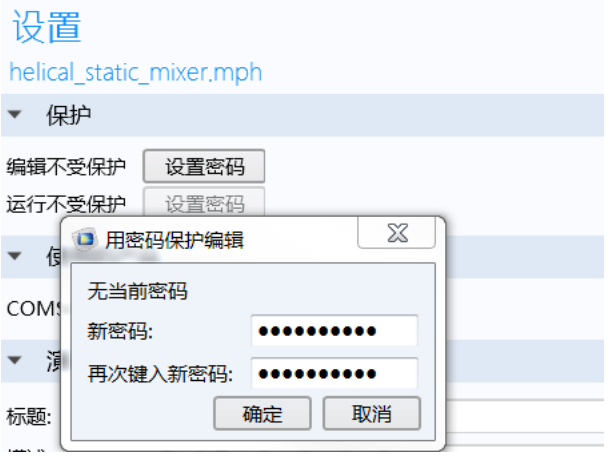


从图形窗口设置选项自动地创建两个预览图片：

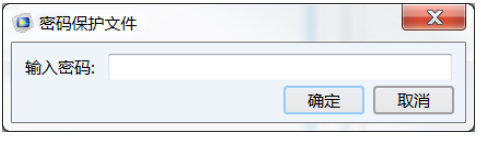
- 一个显示在 App 树的根节点和“案例库”中**设置**窗口的大小为 280X210 像素的图片。
- 一个用于报告的默认标题页和 COMSOL Server 的 “案例库”中的大小为 1024X768 像素的图片。

密码保护

App 可以进行密码保护，管理相关许可。您可以在**设置**窗口分别为编辑和运行设置密码，请单击 App 开发器窗口中 App 树的根节点。您必须拥有编辑 App 的权限来创建运行保护的密码。



当打开受密码保护的 MPH 文件进行编辑或运行时，会跳出一个对话框要求您输入密码：



如果需要删除密码保护，创建一个空密码即可。

密码保护用来加密所有模型和 App 设置，包括方法。然而二进制数据不会被加密，例如，包括内嵌 CAD 文件的定型几何，网格数据，以及解数据等。

安全性设置

当通过 App 开发器创建 App 时，重要的是考虑托管 App 的计算机上的安全性。COMSOL Multiphysics 和 COMSOL Server 提供很相近的安全性设置，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 中的安全性选项可以在**文件**菜单的**首选项**窗口中的**安全性**页面找到。在 COMSOL Server 中，如果您是以管理员身份登录，它们位于

COMSOL Server 网页界面的首选项页面。如果您不确定使用哪种安全性设置，请联系您的系统管理员。

在 COMSOL Server 中运行 App

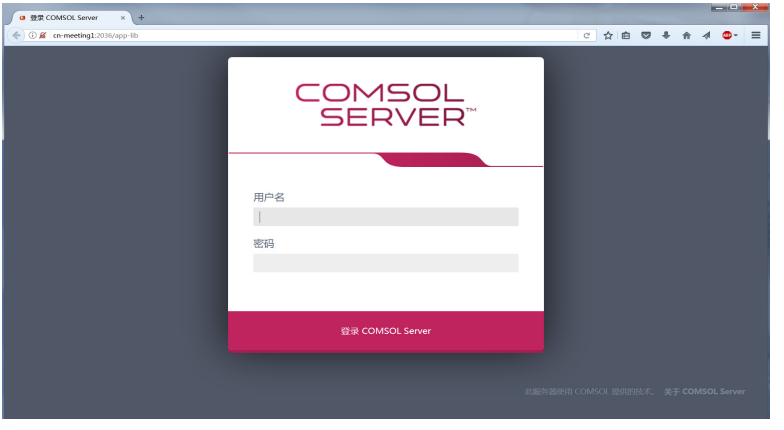
您可以通过网页浏览器或基于 Windows® 的 COMSOL Client 连接到 COMSOL Server 后运行 App。基于 Windows® 的 COMSOL Client 还允许用户运行需要 CAD 产品的 LiveLink™ 的 App，参见 “在 COMSOL Client 中运行 App”。

在网页浏览器中运行 App 不需要安装其他软件和任何网页浏览器插件，这种方式支持交互式一维、二维和三维图像。在网页浏览器中，三维图形渲染基于 WebGL™ 技术来实现，这是所有主流网页浏览器都支持的一种技术。

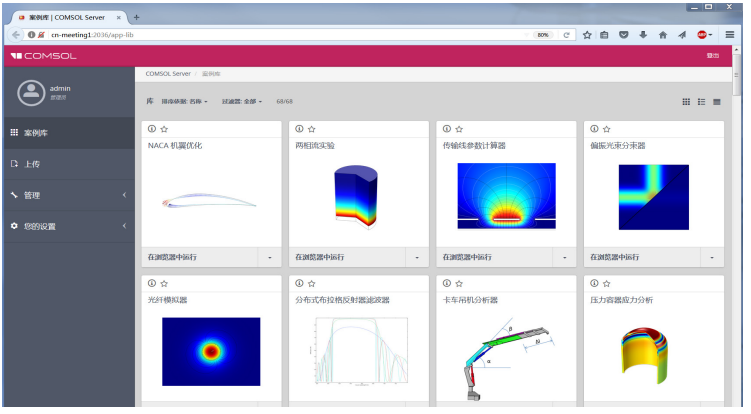
在网页浏览器中运行 App

使用网页浏览器，您可以直接指向 COMSOL Server 网页界面所在的计算机名和端口 — 例如，

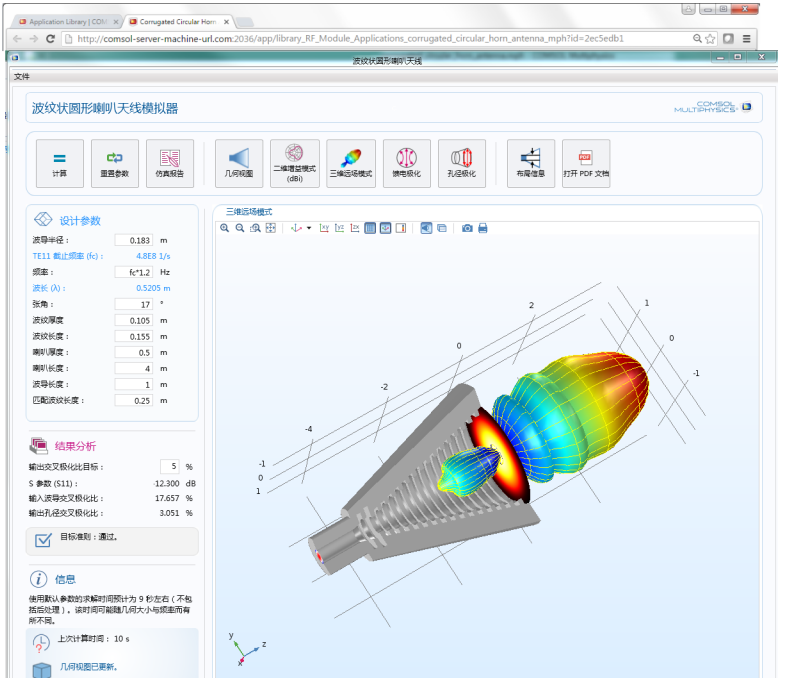
`http://cn.comsol-server-machine-url.com:2036` （假设端口号 2036 ）是安装 COMSOL Server 时指定的端口号。您需要提供用户名和密码来登录。



当登录进去后，**案例库**页面显示可供运行的 App 列表。



单击**在浏览器中运行**可以运行 App，它将在浏览器的另一个单独的选项卡中运行。



在网页浏览器中运行 App 的局限

当您创建一个用于网页浏览器中运行的 App 时，必须确保在 App 开发器中使用栅格布局模式，参见第 83 页的 “草图和栅格布局”。这样可以保证用户界面的布局可以自适应网页浏览器窗口的大小和长宽比。对于低分辨率的显示器，必须确保在目标平台中测试用户界面的布局，检查所有的表单对象都可显示出来。包含可调尺寸的图像表单的 App 可能无法匹配低分辨率显示器。在这种情况下，使用固定宽度和高度的图像，确保所有的表单对象与目标浏览器窗口匹配。依据网页浏览器和显卡的不同类型，可能会对能够在一个 App 使用多少个图形对象有限制。您可以通过使用多图形对象，以及切换源复用相同的图形对象来绕过这个限制。

当在网页浏览器中运行 App 时，不支持 CAD 软件的 LiveLink™ 产品。

当在智能手机和一些平板电脑的网页浏览器中运行 COMSOL App 时，可能不支持全部功能。典型的限制包括播放声音，或者打开文档。此外，也许不能支持文件的上传和下载。

如果 App 允许用户进行选择，例如，在边界上单击以便设置边界条件，在网页浏览器中运行与 COMSOL Multiphysics 或者基于 Windows® 的 COMSOL Client 有明显差异。在网页浏览器中，当在边界上悬停时，这些边界不会自动高亮，需要单击一次来高亮边界，第二次单击完成选择。继续单击第三次将高亮用于取消选择的边界，而第四次单击则完成取消选择。这个过程同样适用于域、边，以及点。

请注意，文件浏览功能略有不同，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，这可能会影响用户体验。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 浏览器可能还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。

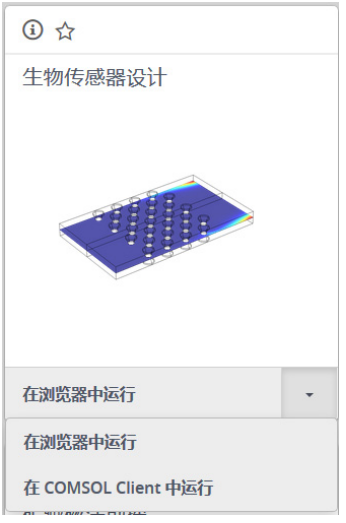
在 COMSOL CLIENT 中运行 App

除了上述提到的使用网页浏览器运行 App，在 Windows® 操作系统中可以使用基于 Windows® 的 COMSOL Client 连接到 COMSOL Server 来运行原生 App。这种方式通常得到更好的图形性能，并支持更多一维、二维和三维的复杂图形渲染。此外，基于 Windows® 的 COMSOL Client 允许运行需要 CAD 软件的 LiveLink™ 产品的 App，这种情况下需要您连接到的 COMSOL Server 上拥有相

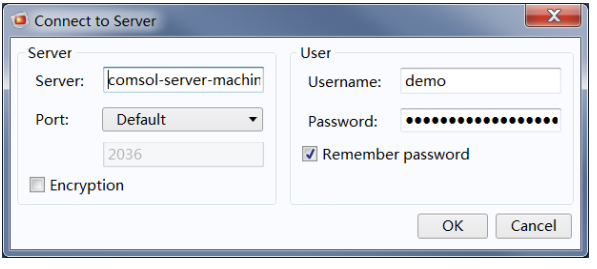
应的许可证。您可以通过两种方式在 Windows® 平台上的 COMSOL Client 中启动 App：

- COMSOL Server 网页界面将允许您选择在网页浏览器中或者使用 Windows® 上的 COMSOL Client 运行 App。

  如果您用这种方式在 COMSOL Client 中运行 App，但还没有安装，您将被要求下载并安装。

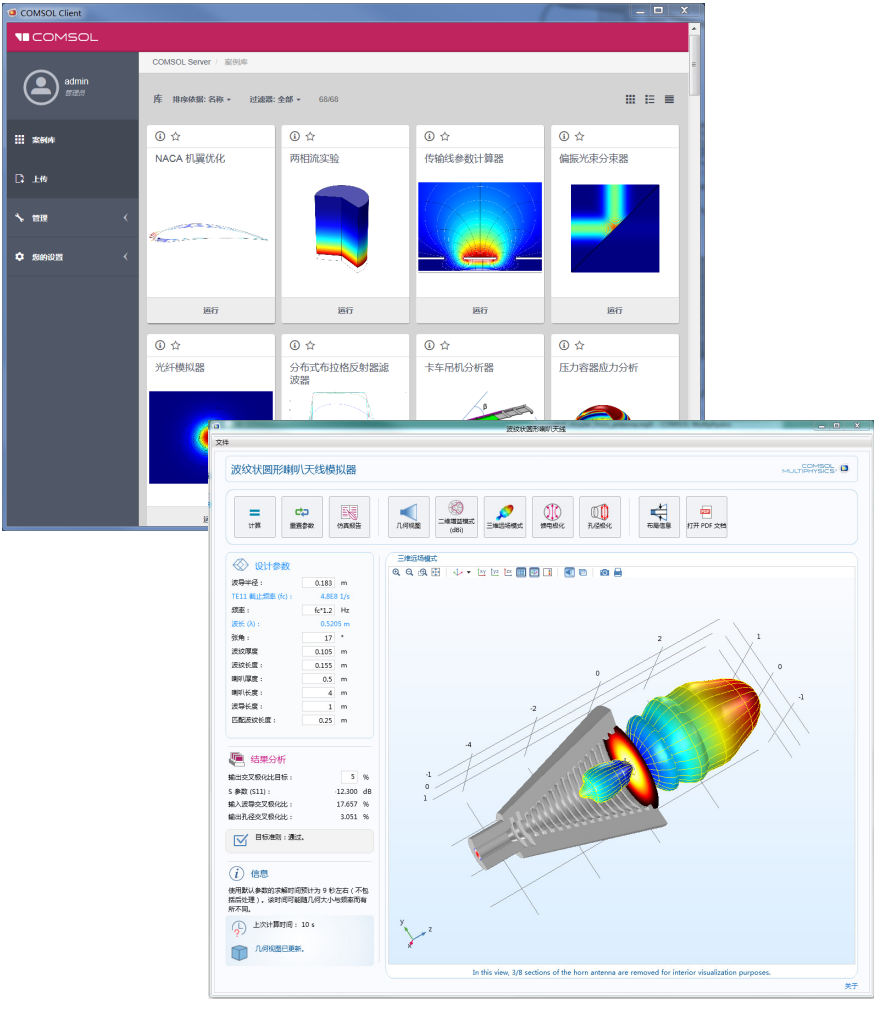


- 如果已经安装了 Windows® 上的 COMSOL Client，将在桌面上出现一个快捷方式。您可以双击它，在您能够使用 COMSOL Client 运行 App 之前，系统将提示您使用有效的用户名和密码登录 COMSOL Server。登录之后， COMSOL Client 中会显示您从网页浏览器登录时看到的相同的 COMSOL Server 网页界面。



使用 COMSOL Client， App 作为原生的 Windows® 应用程序运行于一个单独的窗口。例如， COMSOL Client 可能具有 Windows® 的功能区选项卡。当在网页浏览器中运行时，功能区显示为工具栏。

下图中， COMSOL Server 网页界面上显示 （顶部）已经有一个 App 在 Windows® 的 COMSOL Client 中启动 （底部）。



在多台计算机或集群上运行 COMSOL Server

COMSOL App 主要通过以下两种不同方式在多台计算机或集群上运行：

- 通过主要和次级服务器安装 COMSOL Server。
- 在模型开发器中为一个特性集群配置一个研究节点。

主要和次级实例

与单个计算机实例（或安装）相比，在多台计算机上通过主要和次级实例运行 COMSOL Server，允许更多并发用户和 App。主要的 COMSOL Server 实例称为主实例，其他实例称为次级实例。主实例用于所有传入连接，例如，显示网页界面或者在网页浏览器中或通过 COMSOL Client 运行 App。实际计算会分流到次级服务器计算机。这种类型安装的主要优势在于无需为特定集群定制 App。负载平衡自动由主服务器管理，并在次级服务器之间分配工作负载。

COMSOL Server 安装可以由多个主服务器和辅助服务器安装组成，没有额外的许可证要求。可以使用 COMSOL Server Web 界面执行管理任务，无需校验运行 App 的用户的许可证密钥。仅在运行 App 时校验许可证密钥。

为集群扫描或集群计算配置研究节点

如果您想将集群用于需要较大参数化扫描或高性能计算能力的 App，则可以使用**集群扫描**和**集群计算**选项配置该 App 的**模型开发器**研究节点。请注意，构建这类 App 时，需要浮动网络许可证。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*。运行这类已启用集群的 App 时，可以使用 COMSOL Server 或浮动网络许可证类型的 COMSOL Multiphysics。可以从 COMSOL Server Web 界面进行集群系统设置。

有关 COMSOL Server 的更多信息，请在安装 COMSOL Server 时参阅 *COMSOL Server Manual*，也可以从以下位置进行访问：
http://cn.comsol.com/documentation/COMSOL_ServerManual.pdf。

发布 COMSOL App

COMSOL 软件许可协议 (SLA) 允许您将自己的 COMSOL App 发布给他人进行使用，包括商业用途，但有一定限制，如 SLA 中所述：cn.comsol.com/sla。通过这个许可，您可以与他人共享您的 App 并向他们收取费用。

使用您所提供的 App 的用户需要具有 COMSOL Multiphysics 或 COMSOL Server 的访问权限，以及必要的插件产品。如果是通过 COMSOL Multiphysics 使用 App，用户需要从属于购买 COMSOL Multiphysics 许可证的组织。为了更加灵活，您可以在安装 COMSOL Server 时进行设置，让世界各地的用户都可以访问您的 App。只需向他们提供您的 COMSOL Server 安装用户名和密码。此外，用户也可以购买他们自己的 COMSOL Server 许可证。

通过 COMSOL App 许可证（也可从 cn.comsol.com/sla 获取），您可以进一步修改 “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，但有一定限制，如 App 许可证中所述。例如，借助许可证，您可以通过添加或移除您自己的功能将 “案例库” 中的一个 App 用作您自己的 App 的起点。

如果您想将 App 许可证应用于您创建的 App， App 许可证包含有关如何操作的说明，还可以解决如何使用您选择用于对 “案例库”中的 App 所做的修改的项，同时这些 App 的原始部分在具有 App 许可证的情况下仍可用。

如果您使用 COMSOL Server 托管和运行 App，在遵守特定限制的情况下，SLA 还允许您腾出时间处理适用于组织外部人员的 COMSOL Server 许可证 (CSL)，以托管和运行您发布给他人的 App。

从 COMSOL Multiphysics 模型出发

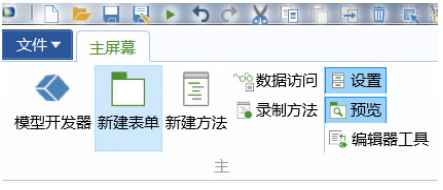
如果您还没有将模型加载到 COMSOL Desktop 环境，可以通过**文件** > **打开**从您的文件系统中选择一个 MPH 文件或者从 App 库选择一个文件。注意，在 **Applications** 文件夹中的文件是已经准备好的 App。 App 库的其他文件包含模型和文档，但是不包含 App 的用户界面。

当模型加载后，单击**主屏幕**选项卡中的 **App 开发器**按钮，切换到 App 开发器的桌面环境。



创建一个新表单

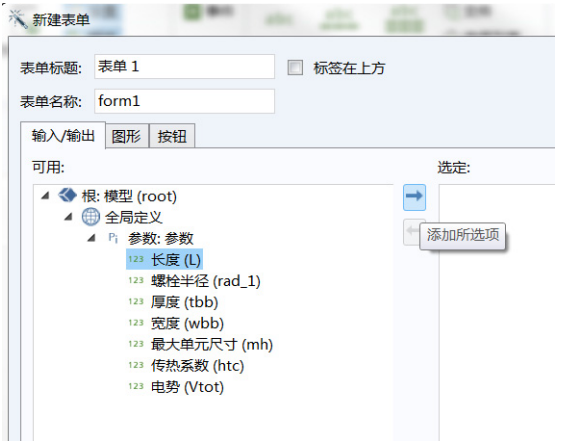
单击**主屏幕**选项卡上的**新建表单**按钮，开始开发一个用户界面布局。这时将启动**新建表单**的向导。



新建表单向导将帮助您添加最常用的用户界面控件，即所谓的表单对象。其中包含三个选项卡：

- **输入 / 输出**

- **图形**
- **按钮**



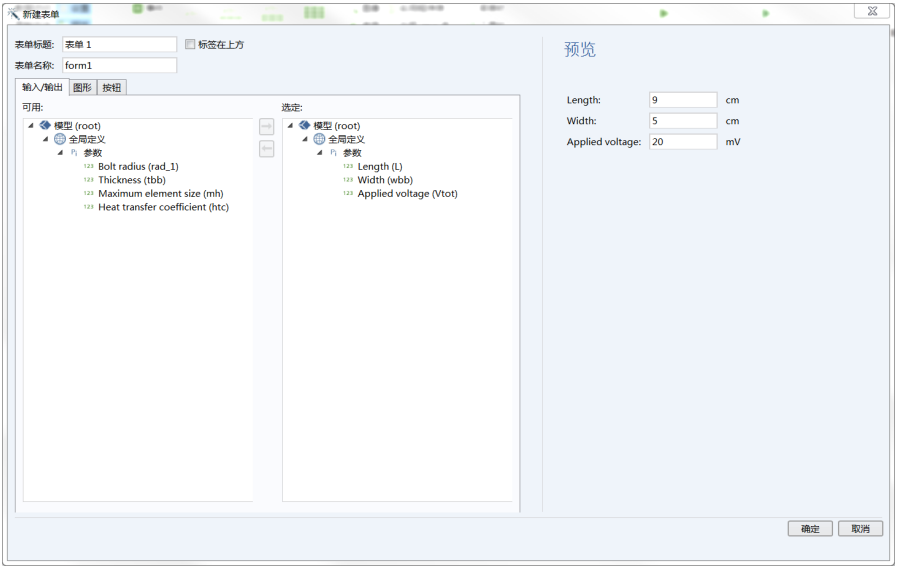
双击节点或单击**添加选定** 按钮，将节点从**可用**栏移动到**选定**栏。选定节点将成为 App 中的表单对象，在右边的**预览**区将显示表单的预览图。表单对象的尺寸以及其他一些设置可以在退出向导后进行编辑。在向导窗口的顶部，您可以修改表单的名称和标题。详情参见第 37 页的 “单独的表单设置窗口”。

您还可以通过单击**完成**按钮在这个阶段选择退出**新建表单**，后续可以手动添加表单对象。

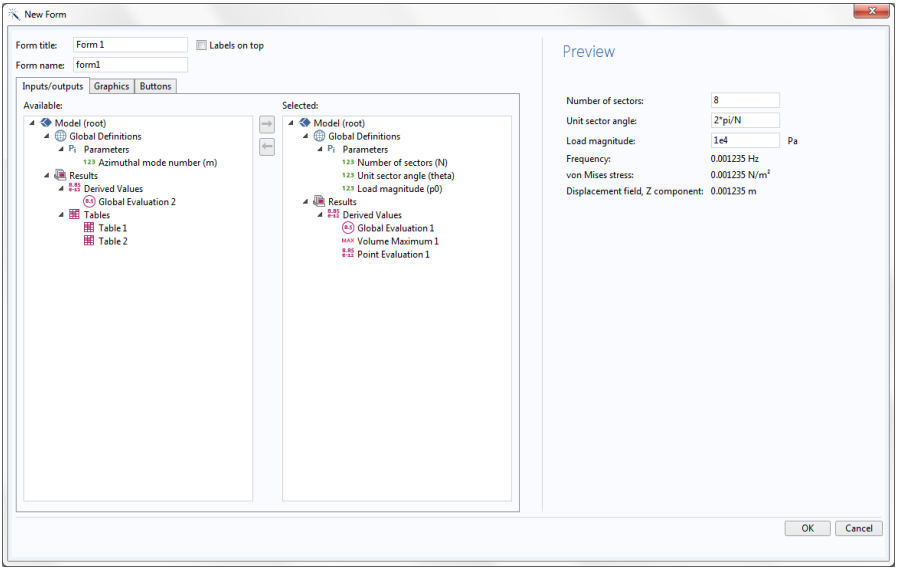
输入 / 输出选项卡

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。您可以通过使用**数据访问** （参见第 77 页）来使得模型的其他部分可以用作输入和输出。例如，复选框和组合框只能通过这种方式添加。例如，您可以通过启用**数据访问**，使得**网格**节点下预定义的**单元尺寸**组合框可以出现在向导中。

在下图中，选定了三个参数作为输入框，包括长度、宽度和应用电势：



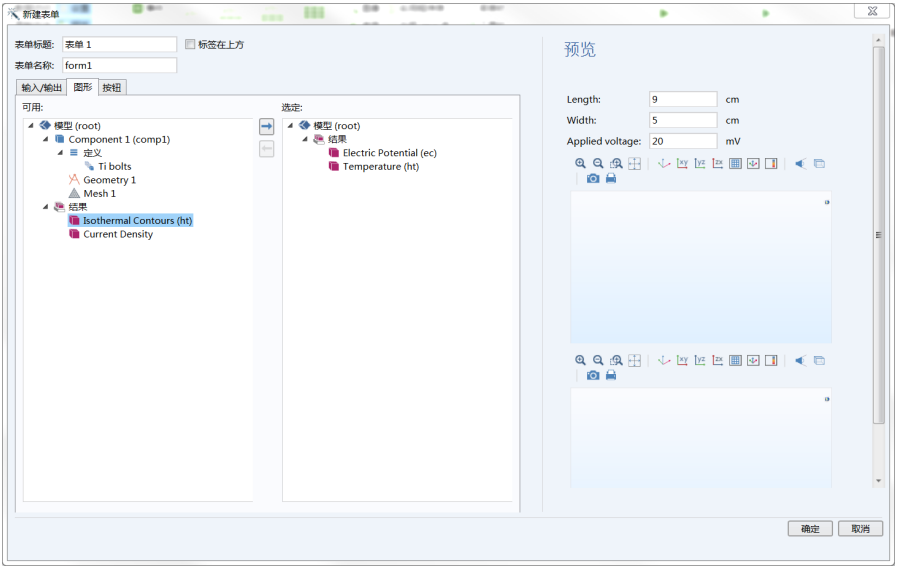
下图中，对应于另一个模型，选定了三个**派生值**节点作为数据显示对象：



当退出向导后，您可以进一步编辑输入框和数据显示对象的尺寸和字体颜色，以及其他设置。

按钮选项卡

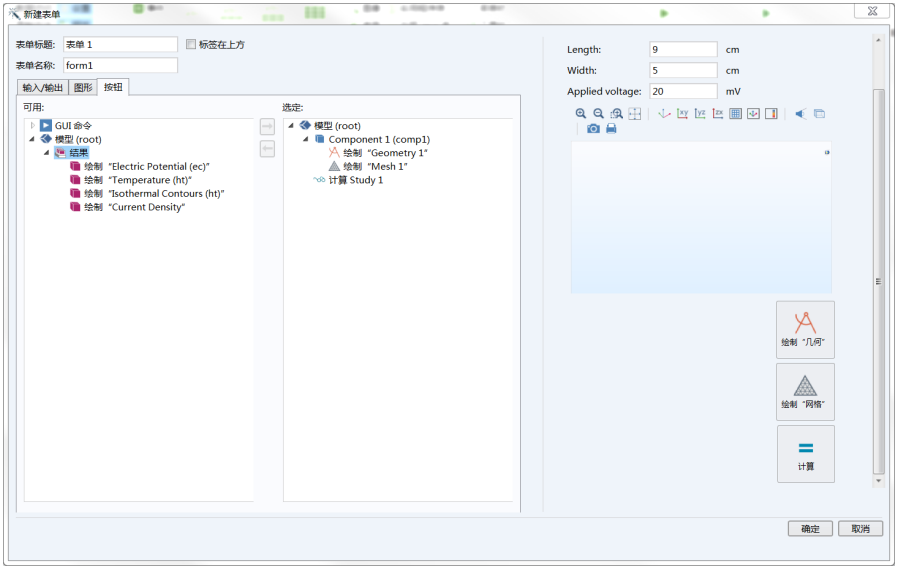
在模型树节点下的**图形**选项卡下可以另存为图形实体：**几何**，**选择**，**网格**和 **结果**。在下图中，就选择了两个这样的节点。



按钮选项卡

按钮选项卡显示可以通过单击 App 用户界面中的按钮进行运行的模型和 App 树节点。这样的树节点包括**绘制几何**、**绘制网格**、**计算研究**，以及**结果**下的各个绘图组。此外，您还可以添加用于 **GUI 命令**、**表单**，以及**方法**的按钮。

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



使用表单编辑器，您可以添加运行您自己的定制命令序列或方法的按钮。

退出向导

单击**确定**退出向导，这将自动导向到表单编辑器。

保存 App

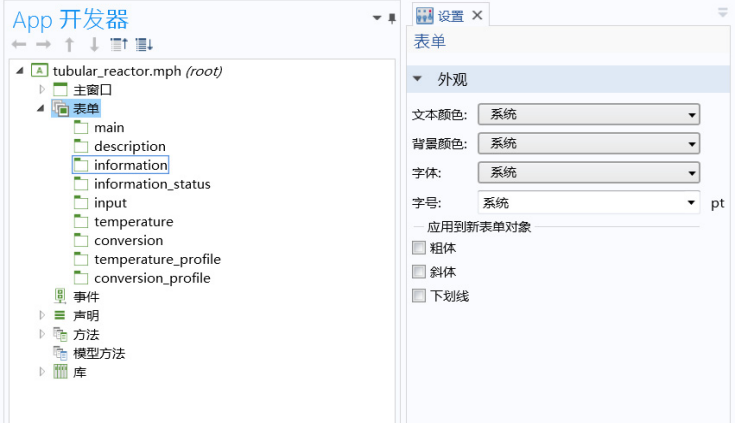
如果需要保存 App，请从文件菜单选择**文件** > **另存为**。浏览到您具有写权限的文件夹，将文件保存为 MPH 格式，这个 MPH 文件中包含所有的 App 的信息，包括由模型开发器创建的内嵌模型。

表单编辑器

使用表单编辑器来开发用户界面布局，创建包含表单控件的表单，例如，输入框、图形、按钮，等等。

表单设置窗口

当单击 App 树的**表单**节点时，会显示表单的设置窗口。它可供您改变表单的外观，可以修改设置项**文本颜色**、**背景颜色**、**字体**、**字体大小**、**粗体**、**斜体**，以及**下划线**。

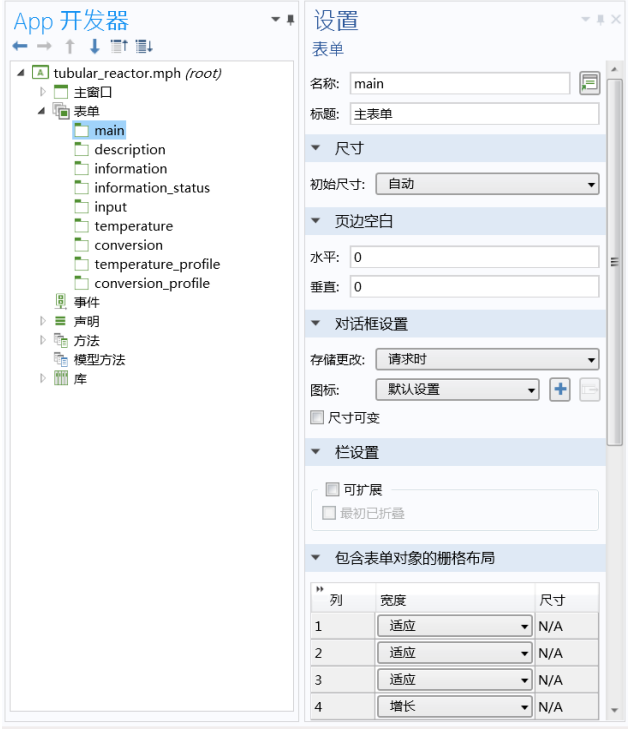


默认设置将会在可能使用的情况下被所有新建表单和新建表单对象继承。

上图中，以及下面的一些图中，**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

下图显示了一个表单的设置窗口。

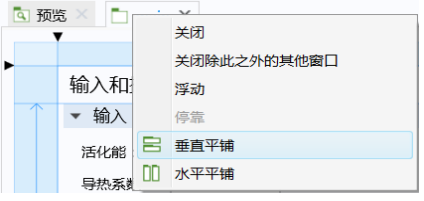


每个表单具有各自的**设置**窗口，包含以下设置：

- 用于在其他表单对象和方法中引用表单的**名称。**
- 用于多个表单的 App 中的表单**标题。**
- 当用作对话框或者当**主窗口**设置为由表单决定其尺寸时，表单的**初始尺寸。**
- 到左上角 （**水平**和**垂直**）的**页边距。**
- 何时保存对话框中的修改 （**保存修改**）的选项，参见第 56 页 “表单显示为对话框”。

- 对话框左上角显示的**图标**。
- 当作用对话框时，表单是否**尺寸可变**的选项。
- 查看的栏是否**可扩展**，以及它们是否**初始折叠起来** （**栏设置**）的选项。
- 表单中所有行和列的格式列表 （**包含表单对象的栅格布局**）。
- **外观**中包含**文本颜色**、**背景色**，以及**背景图片**。
- **事件**可以在加载或关闭表单时触发 （**加载时执行**或**关闭时执行**）。

双击一个表单节点，在表单编辑器中打开其窗口。或者，您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可以查看上下文菜单，其中包含的选项有：关闭、浮动，排列表单窗口等。



草图和栅格布局模式

App 开发器默认使用草图布局模式，在这种情况下，您使用的是固定的对象位置和尺寸。除非有其他说明，“表单编辑器”节的指南假定表单编辑器为草图布局模式。关于栅格布局模式的详细信息，请查阅第 64 页的 “草图和栅格布局”。

表单的初始尺寸

表单的初始尺寸有两种选项：

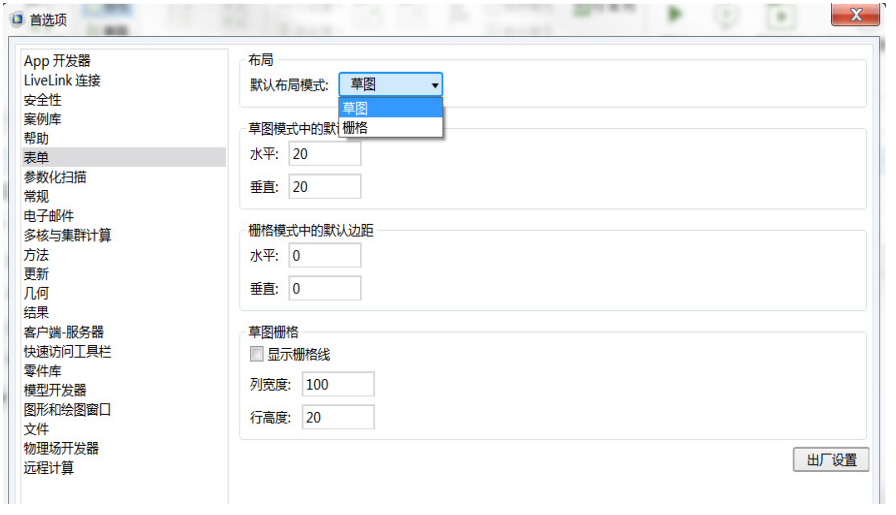
- 手动，让您输入宽度和高度的像素值。
- 自动，基于表单包含的表单对象来自动确定尺寸。如果您使用栅格模式，其中的列或行设置为**增长**，则尺寸不由表单对象决定。在这种情况下，使用表单编辑器的栅格尺寸作为基准点 （通常略大）。您可以通过拖动栅格的右边

或底边来修改栅格的尺寸。参见第 85 页 “栅格布局”，了解更多关于栅格布局模式的信息。



表单编辑器首选项

通过以下操作可以访问表单编辑器的**首选项**，从**文件**菜单的**首选项**对话框中选择**表单**页面。

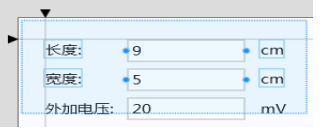


表单栏包含修改以下默认项目的设置：**布局模式**、**页边空白**和**草图格点**。

定位表单对象

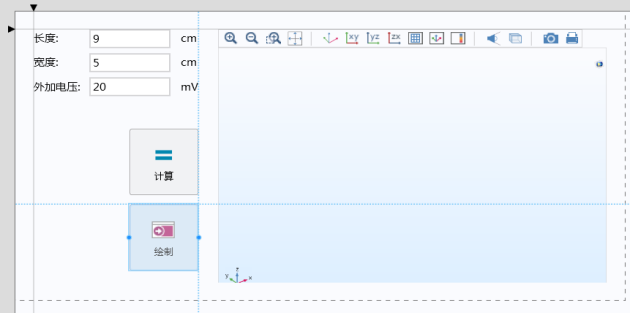
您可以使用以下方法之一很方便地修改表单对象的定位，例如，输入框、图形对象和按钮等：

- 单击对象来选中它，选定的对象将会蓝色边框高亮显示。
- 如果要选中多个对象，可以使用 Ctrl+ 单击。您还可以在表单窗口中单击和拖动，创建一个选择框来选中框内的所有对象。
- 按住并拖动到新位置，蓝色引导线将有助于相对于其他对象进行定位。



- 在草图布局模式，您还可以使用键盘上的箭头键来移动对象。使用 Ctrl+ 箭头键还可以进行微调定位。

下图中，一个**绘图**按钮从它的原始位置移动，其中蓝色引导线显示出单位对象和**计算**按钮的对齐情况。



改变表单对象的尺寸

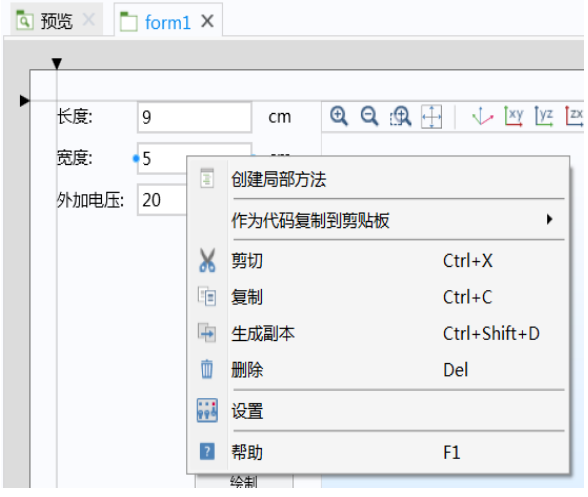
以下操作可修改对象的尺寸：

- 单击对象来选中它。
- 按住并拖动其中一个控制点，它在高亮的蓝色边框中以蓝色点显示。如果没有控制点，说明不能修改这种类型的表单对象的尺寸。

复制、粘贴、创建副本，以及删除对象

如果要删除一个对象，单击选中它，然后按下键盘上的删除键。您还可以单击快速访问工具栏上的删除按钮。

您可以使用 Ctrl+C 和 Ctrl+V 来复制 - 粘贴对象，或者您还可以右键单击对象，从菜单中选择**复制**、**创建副本**、**删除**，以及其他操作。



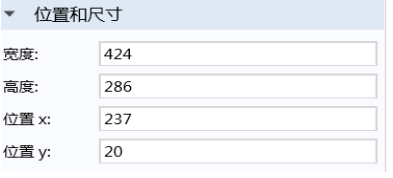
如果要粘贴一个已经复制的对象，在表单的空白区域再次右键单击，当已有复制的对象时，将显示**粘贴**菜单项。下图中之前复制了一个输入框，因此这里显示的是**粘贴输入框**。



通过像素值调整位置和尺寸

当您在草图布局模式，可以通过在**设置**窗口的**位置和尺寸**栏中键入像素值来调整位置和尺寸：

- 单击对象来选中它，确保显示它的**设置**窗口。如果没有，可双击对象或者在**表单**选项卡中单击**设置**按钮。
- 在**位置和尺寸**栏的编辑数值。

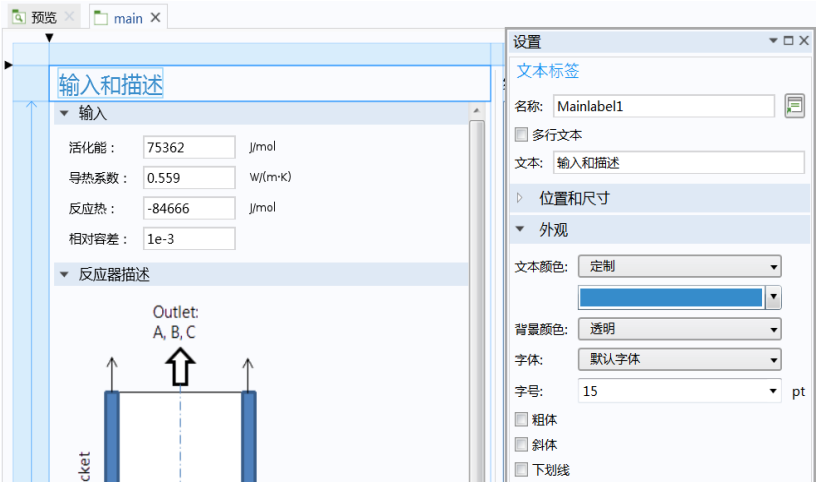


根据表单对象的类型，**位置和尺寸**栏会有不同的选项。对于栅格布局模式，对行和列分别有对象的位置的附加选项，详情参见第 83 页 “草图和栅格布局”。

修改显示文本的外观

对于显示文本的表单对象，**设置**窗口的**外观**栏可供您修改显示文本的属性，例如，字体、字体颜色，以及字体尺寸。对于诸如按钮类型的表单对象，对象的尺寸将适配于字符串的长度。

下图中的设置窗口，显示修改了字体大小和颜色的文本标签对象。



通过使用格点布局模式 （参见 93 页的 “草图和网格布局”），您可以进一步控制表单对象的大小，例如设置一个按钮的任意大小。

选择多个表单对象

如果您选择了不止一个表单对象，例如，使用 Ctrl+ 单击，则**设置**窗口将包含一组可以在选定对象之间共享的属性。共享的属性总是来自于**外观**栏，**位置和尺寸**栏以及**事件**栏。

表单对象的名称

表单对象通常都有名称，通常是一个不包含空格的字符串，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 等。**名称**字符串可以用于在其他表单对象或方法中引用对象。当在**设置**窗口的**名称**框上悬停时，对象的路径显示为工具提示。

插入表单对象

您可以在那些**新建表单**向导创建的表单中再插入一些表单对象。在**表单**功能区标签，选择**插入对象**菜单可以看到所有可以使用的对象的列表。



表单编辑器章节剩下的部分只描述了由**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能包括：

- **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**

但是，使用**数据访问**(请参见 88 页)时，可能也会添加以下表单对象：

- **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 180 页的 “附录 A —— 表单对象”。

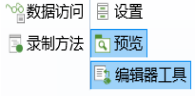
与表单对象关联的事件和操作

您可以将对象关联到由事件触发的操作，这些对象包括按钮、菜单项、功能区按钮、表单，以及表单对象。一个操作可以是一个命令序列，其中包含全局或局部方法。局部方法不能在定义对象之外的地方调用或可见。可以与对象进行关联的事件依据对象类型有所不同，包括：单击按钮、键盘快捷方式、加载表单 （加载时执行）、关闭表单 （关闭时执行），以及修改变量值 （修改数据时执行）等。

在表单对象上使用 Ctrl+Alt+ 单击可以打开**方法**编辑器中的任何相关方法。如果没有与表单对象相关联的方法，将创建一个新的局部方法与表单对象进行关联，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，则该序列将转换为代码并插入到局部方法中。

表单编辑器中的编辑器工具

编辑器工具窗口是**新建表单向导**的一个重要的补充，可以通过插入对象菜单快速地创建复合表单对象。单击**表单**选项卡上方的**主**组中相应的按钮，可以打开**编辑器工具**窗口。



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



当选定一个节点，在编辑器树下的工具栏中会显示可供使用的选项，用于插入对象。还可以右键单击查看包含这些选项的列表。

根据节点的不同，可能出现以下选项：

- **输入**

  插入下列**输入框**、**复选框**、**组合框**或**文件导入**对象：

  - 使用选定节点作为**源**插入**输入框**，其中可能包含**文本标签**和**单位**对象。
  - 使用选定节点作为**源**插入**复选框**。
  - 使用选定节点作为**源**插入**组合框**，会自动创建一个选择列表，对应于节点的列表。本选项只有当使用**数据访问** （参见第 77 页）使得对应的节点可以在编辑器树中使用时才有效。
  - 使用选定节点作为**文件目标**插入**文件导入**对象。

- **输出**
  - 当可以使用时插入**数据显示**对象，其中可能包含**文本标签**。
  - 当选定节点为**表格**时插入结果**表格对象**。
- **按钮**
  - 插入一个**按钮**对象，并包含运行选定节点的命令序列。
- **图形**
  - 使用选定节点作为**初始图形内容的源**插入**图形**对象。
- **编辑节点**
  - 给出对应模型树节点的**设置**窗口。

当使用方法编辑器时，编辑器工具窗口也是一个重要的工具。在方法编辑器中，它用来生成与编辑器树节点关联的代码。参见第 128 页 “方法编辑器中的编辑器工具”，了解更多信息。

按钮

单击按钮，相当于触发由命令序列定义的操作的事件，按钮的设置窗口中的主栏允许您：

- 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**来取代默认的按钮的渲染方式。
- 将按钮的**尺寸**从**大**改为**小**。

- 添加**工具提示**，当在按钮上悬停时可以显示出来。
- 通过单击输入框并在键盘上使用 Shift、Ctrl 和 Alt 等修正键与另一个键组合形成**键盘快捷键**。其中 Alt 必须与其他至少一个修正键进行组合。



选择运行的命令

选择运行的命令栏让您控制与单击按钮事件相关联的操作。下图的**设置**窗口显示一个触发四条命令序列的按钮。



在菜单、功能区或工具栏等的设置窗口中，也提供了**选择运行的命令**栏，其功能见本节的描述。要了解更多使用菜单、功能区和工具栏的信息，请查阅第 63 页的 “图形工具栏”、第 99 页的 " 主窗口”、第 224 页的 “表格”，以及第 231 页的 “工具栏”。

如果需要定义命令序列，可以在**选择运行的命令**栏中选择编辑器树中的节点，然后单击树下的某个高亮按钮或右键单击并选择命令。下图中选定了**几何**节点，然后**运行**和**绘图**命令高亮显示。单击**运行**向命令序列中添加创建几何命令。单击**绘制**添加先构建几何后绘制几何的命令。**编辑节点**选项将跳转到模型树或 App 树中的对应节点。



无需在**绘图几何**命令之前执行**构建几何**命令 （可通过单击**运行**执行）。**绘图几何**命令将先构建几何，然后绘制几何。类似地，**绘图网格**命令将先构建网格，然后绘制网格。

高亮显示可供选择的命令图标是指那些可应用于特定树节点的命令。下面的列表是可供使用的命令图标，依据节点而变：

- **运行**
- **绘图**
- **设置值**
- **显示**
- **显示为对话框**
- **导入文件**

- **启用**
- **禁用**

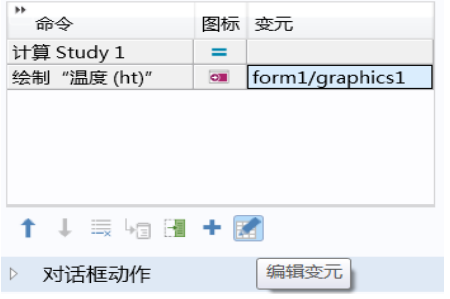
一些命令需要变元，例如各种绘图命令，它们的变元用于定义在哪个图形对象上渲染绘图。

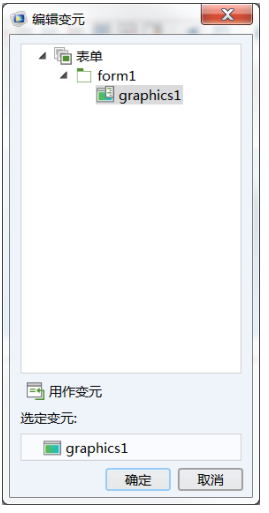
下图显示由**新建表单**向导创建的**计算**按钮的**设置**窗口和命令序列。这个按钮包含一个有两条命令的序列：**计算研究 1** 和**绘制温度**。



其中**绘制温度**命令有一个变元 graphics1。

如果要添加或修改输入变元，可以单击命令序列下方的**编辑变元**按钮，如下图所示。



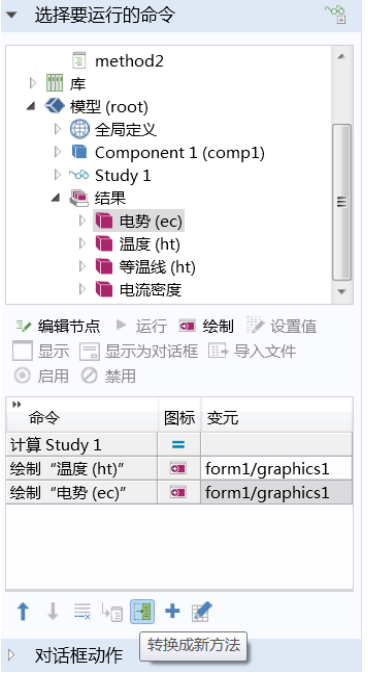


为了在特定的表单中引用图形对象，需要使用以下语法：/form1/graphics2、/form3/graphics1，等等。如果没有指定特定的表单，例如， graphics1，则使用按钮所在的表单。

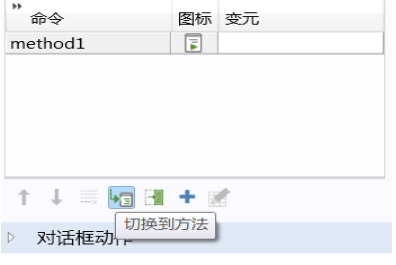
可以使用命令序列表格下方的一些按钮来控制命令的顺序和内容，包括：**上移**、**下移**，以及**删除**等按钮。

将命令序列转换为方法

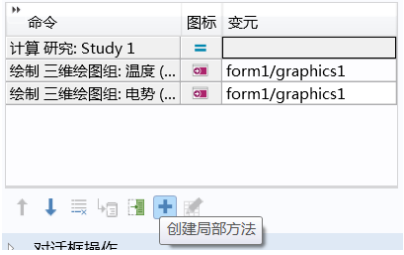
通过单击**转换为新方法**，一个命令序列可以自动地转换为一个新方法，并进一步在方法编辑器中进行编辑。



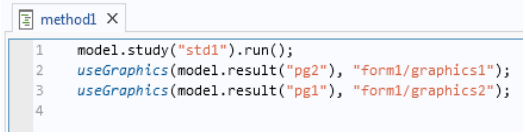
通过单击**切换到方法**打开新方法。



您还可以通过单击**创建局部方法**来创建一个表单对象的局部方法。下图显示了这些选项。



方法中包含调用对应于命令序列中的各条命令的内建方法，如下图所示。



在这个例子中，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，除非用户进行修改）的模型树节点。第二和第三行：

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

分别使用内建方法 useGraphics 显示对应于绘图组 pg1 和 pg2 的绘图。在这个例子中，这些绘图分别显示在两个不同的图形对象中，即 graphics1 和 graphics2。

参见第 123 页 “方法编辑器”，了解更多信息。

设置参数和变量值

设置值命令可以用来设置参数和变量的值，它们常用于**参数**、**变量**，以及**声明**节点。此外，**设置值**还可以用来设置由**数据访问 （参见第 77 页）**功能开放的

可以访问的属性。下图显示了用来初始化一组参数的命令序列，以及字符串变量。



如果要了解如何从方法执行相同命令序列的操作，可以单击命令表下方的**转换成新方法**按钮。

修改表单的可见性

表单上有个按钮还可以用于显示新表单，有两种实现方法。第一种是使用**显示**命令，这将使用新表单替换原始表单。第二种是使用**显示为对话框**命令，新表单将以对话框形式弹出在当前表单上，而且通常需要用户输入。

在**选择运行的命令**栏，您可以选择**显示**命令，下图显示的按钮带有**显示 form3**命令的命令序列。



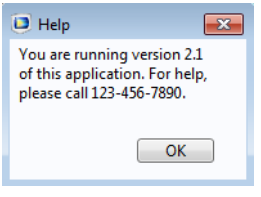
这个命令将保留表单与按钮关联，并使得指定的表单可见。

表单显示为对话框

为了使用**显示为对话框**命令，从**选择运行的命令**栏开始，然后选择需要显示的表单。下图显示了一个带有**作为对话框显示 form2** 命令的设置窗口示例。



通过这些设置，在 App 中单击按钮将启动下面这个对应于 **form2** 的对话框：



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，**对话框操作**栏有两个复选框：

- 关闭对话框
- 保存修改

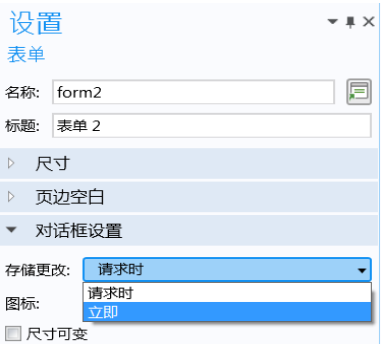
在上例中，选定了**关闭对话框**复选框。这确保当单击**确定**按钮时将关闭 **form2** 窗口。因为 **form2** 不包含任何用户输入，因此不需要选择**保存修改**复选框。

常用的对话框按钮及其相关对话框操作有：

按钮	对话框操作
确定	关闭对话框并保存修改
取消	关闭对话框
应用	保存修改

对话框屏蔽 App 中的其他任何用户交互，直到它被关闭。

为了控制输入对话框的数据是否被保存下来，在表单的设置窗口的对话框设置栏，您可以选择当发生修改时是否**在应求时执行**或**立即**保存修改，如下图所示。



外观

在按钮的**设置**窗口中，**外观**栏包含字体设置以及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**启用**，由**状态**子栏下的复选框控制。大多数表单对象的**外观**栏设置类似，但部分表单对象具有其他选项，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 的用户界面中。已清除**启用**复选框的表单对象将被禁用，或 “灰显”，但仍可见。表单对象的状态还可以通过内置方法来控制。例如，假定一个已启用或已禁用的布尔变量用于决定名为**按钮 3** 的按钮的启用 / 禁用状态。在这种情况下，您可以用以下方式控制按钮的状态：

```
setFormObjectEnabled("button3", enabled_or_disabled);
```

类似地，以下调用

```
setFormObjectVisible("button3", visible_or_not);
```

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 294 页的 “GUI 相关的方法”*和 Application Programming Guide*。

图形

在图形对象的设置窗口，使用**初始化图形内容的源**来设置默认显示的绘图组。您可以单击**作为源使用**，或者单击树中的节点。如果已经存在绘图组的解，当打开 App 时会显示出对应的解。下图显示了以**温度**绘图作为源的图形对象的**设置**窗口。

选择初始图形内容的源

在图形对象的**设置**窗口，使用**初始化图形内容的源**来设置默认显示的绘图组或动画。您可以单击**作为源使用**，或者单击树中的节点。如果已经存在绘图组的

解，当**打开 App 时会显示出对应的解。下图显示了以温度绘图作为源的图形对象的设置窗口。**



除了**结果**绘图节点，您还可以使用**动画**、**选择**、**几何**以及 **网格**节点作为**选定的源**。

选中**将第一个绘图缩放到窗口大小**复选框可确保出现在图形画布中的第一个绘图显示完整的模型 （缩放到窗口大小）。图形内容首次发送到图形对象时将触发该操作。

选中**数据采集**复选框可使图形对象交互，这样您可以 （以此为例）单击特定点处的绘图，并检索该坐标处的温度数值。有关更多信息，请参阅 77 页的 “数据采集”。

外观

对于图形对象，**设置**窗口的**外观**栏有以下选项：

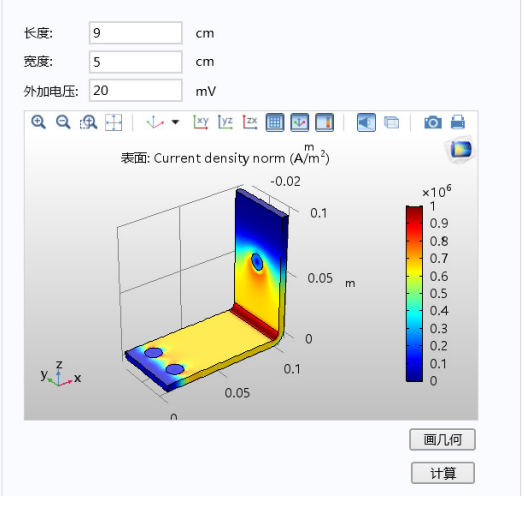
- 包含右上角商标的**图标**。

- 设置二维绘图的背景**颜色**。
- 通过选择**顶部颜色**和**底部颜色**来调整的均匀或渐变背景颜色。



此外，子栏**状态** （上图中未显示）包含图形对象可见和已启用状态的设置。有关更多信息，请参见 61 页的 “更改表单对象的启用和可见状态”。

下图显示的 App 中，背景颜色的**顶部颜色**设置为白色，**底部颜色**设置为灰色。此外，其中没有包含标准的绘图工具栏。



图形命令

在按钮等命令序列中使用的编辑器树中，**图形命令**文件夹包含处理或修改图形对象的命令。下图显示了含有一个打印图形对象内容的命令的命令序列。



有以下这些**图形命令**可供使用：

- **缩放到窗口大小**
  - 使得整个模型可见。
- **重置当前视图**
  - 将当前激活的视图重置为启动 App 时的状态， 参见第 66 页 “视图”。
- **背景光**
  - 触发背景光 （开或关）。
- **透明度**
  - 触发透明度设置 （开或关）。
- **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。

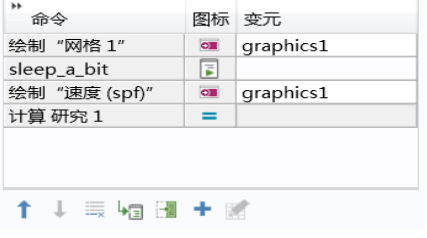
注意，在标准的图形工具栏上有这些命令的工具拦按钮，包括**缩放到窗口大小、重置当前视图、背景光、透明度**，以及**打印**等，参见下一节 “图形工具栏”。

边求解边绘图

您可以在计算过程中绘制结果，让用户监视模型的收敛性。在这个示例中，假设启用了**求解过程中的结果**的**绘图**选项。这个选项可以在模型树**研究**节点的**设置**窗口中选中，如下图所示。



您可以加入一种调用内置 sleep 法的方法，在切换显示其他类型的图形前简略显示图形信息。将其插入到命令序列中的绘图命令之后，如下图所示。



在该示例中， `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 300 页的 “sleep”。

在上面的命令序列中，**计算研究**命令之前有一个**绘制速度**命令，确保图形对象可以一边求解一边绘制速度。

使用多个图形对象

由于在可能运行您的 App 的平台上可能受到显卡硬件资源的限制，您应当力争使用最小数量的图形对象，这样可以确保最大可能地运行 App。此外，如果您试图在网页浏览器中运行 App，可能还会有一些额外的可以使用多少个图形对象的限制。不同组合的硬件、操作系统，以及网页浏览器有不同的限制。

在这种情况下，在不同的表单中的两个相同名称的图形对象相当于是两个不同的图形对象。例如， form1/graphics1 和 form2/graphics2 代表两个不同的图形对象。此外，如果图形对象用于子表单 （参见第 194 页 “表单”），那么每使用一个子表单，就相当于一个不同的图形对象。

为了在 App 中显示很多不同的绘图，您可以创建按钮、切换按钮，或者单选键等，在不使用子表单的表单中绘制相同的图形对象。

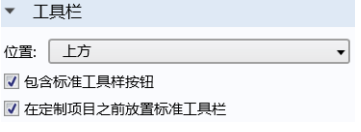
如果您需要使用方法来修改绘图，可以使用 useGraphics 命令。下面的代码示例基于一个布尔变量的值，通过重用相同的图形对象来切换不同的绘图组。

```
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;
}
```

图形工具栏

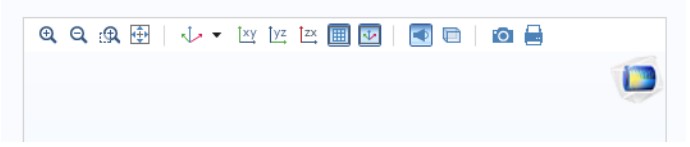
用于**初始化图形内容的源**的树节点类型决定显示的工具栏类型。根据空间维度的不同，以及引用的源是否为**几何**、**网格**、**选择**或**绘图组**节点，工具栏会有不同。例如，**绘图组**节点会额外显示一个**显示图标符号**按钮。

在图形对象的**设置**窗口的**工具栏**设置栏中，您可以控制是否引用图形工具栏，以及它的位置 （**下方**、**上方**、**左侧**、**右侧**）。



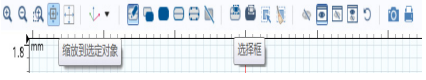
几何和网格的图形工具栏

下图显示标准图形工具栏在三维模型的**几何**或**网格**节点用作**初始化图形内容的源**时的外观。



选定对象的图形工具栏

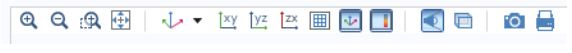
当**初始化图形内容的源**设置为**显式**选定对象时，图形工具栏将包含其他项：**缩放到选定对象**、**选择框**和**取消选择框**。如下图所示 （**取消选择框**位于**选择框**的右侧）。



有关更多信息，请参见 73 页的 “选定对象”。

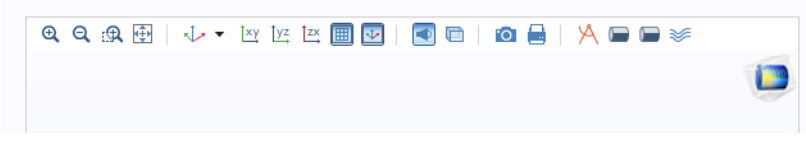
绘图组的图形工具栏

下图显示标准图形工具栏在**三维绘图组**节点用作**初始化图形内容的源**时的外观。



定制图形工具栏按钮

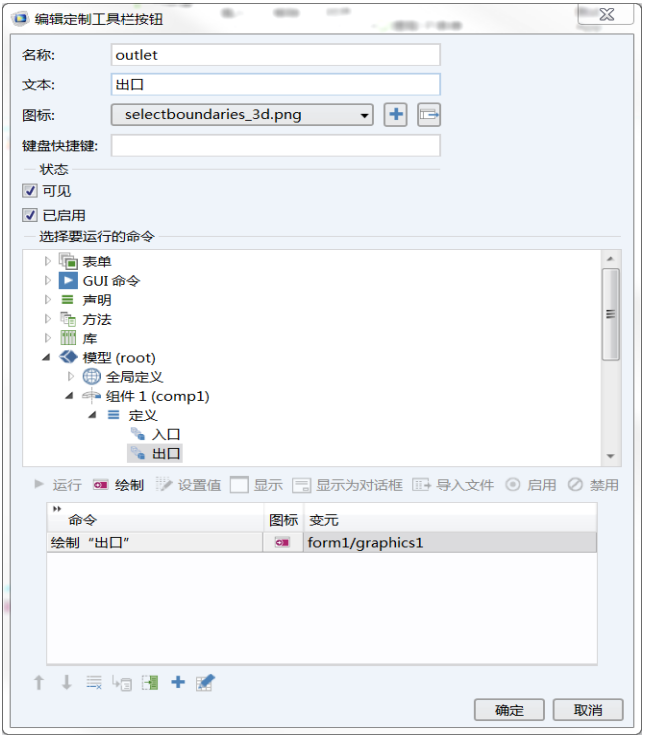
在工具栏设置栏中，您还可以在图形工具栏中添加定制的按钮。使用表格下的按钮来添加或删除定制工具栏按钮 （项）。您还可以将这些按钮上移或下移、添加**分隔线**，以及**编辑**按钮等。下图显示在一个标准的图形工具栏右侧添加了四个按钮。



下图显示对应的图形工具栏项的表格。



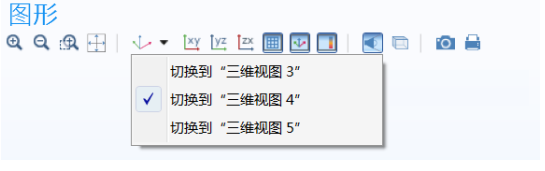
如果想要编辑工具栏项的命令序列，可以单击**编辑**按钮，打开**编辑定制工具栏按钮**对话框。



这个对话框中是与那些按钮类似的设置，详情参见第 47 页 “按钮”。

视图

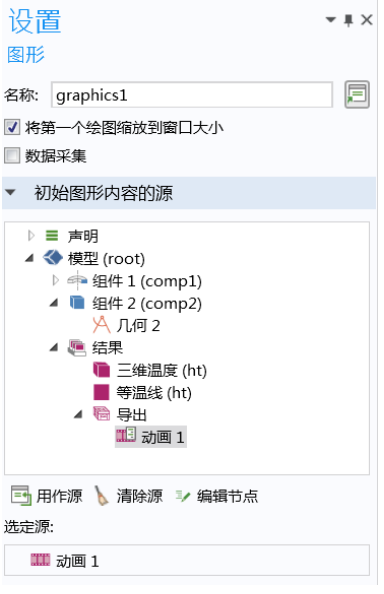
在图形工具栏中，**切换到默认三维视图**按钮 （只适用于三维显示）将显示包含所有可以使用的视图的菜单，当前激活的视图前会有一个勾选标记。



除了所有视图的列表，还有一个选项为**重置当前视图**，用来将当前激活的视图重置为启动 App 时的状态。

动画

可以通过使用**结果** > **动画**节点作为 **初始图形内容的源**，在 App 中显示 动画。



要运行动画，请使用 “新建表单” 向导或 “编辑器工具” 窗口从 （以此为例）运行**结果** > **动画** 节点的按钮创建一个命令。



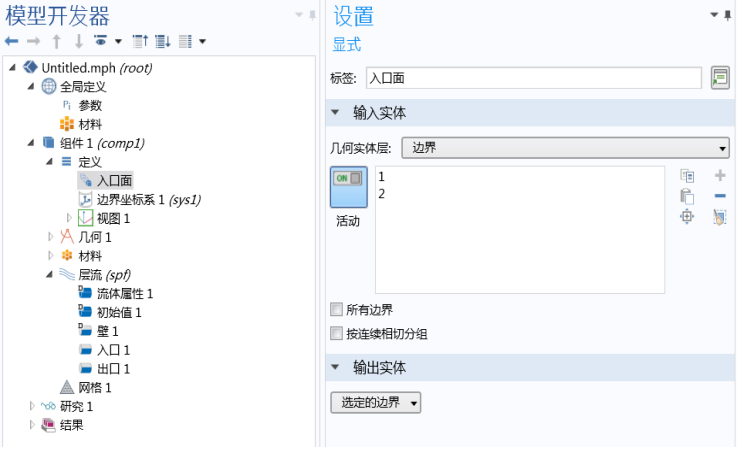
使用 “新建表单” 向导或 “编辑器工具” 时，动画按钮默认外观如下所示：



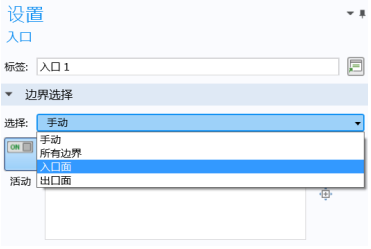
选择

模型开发器中的选择

在模型开发器中，指定名称的 选择可以让您分组域、边界、边，或者点，用于分配材料属性、边界条件，以及其他模型设置。您可以在**组件** > **定义**节点下通过添加子节点来创建各种类型的选择。这些可以在整个模型组件中重用。作为一个有关如何使用选择的示例，我们来考虑边界条件的选择。在选择哪些边界与特定边界条件相关联时，可直接在 COMSOL Desktop 环境图形窗口中的这些边界上单击。这是称为**手动**选择的默认选项 （如下图所示）。然后，这些边界将添加到与该边界条件关联的选择。通过指定名称的选择，您可以定义可重用于几个不同类型边界条件的全局选择，只需从下拉列表进行选择即可。下图显示带有两个关联边界 （22 和 88）的给定名称为**入口**的**显式**选择。



下图显示使用了**入口**选择的**入口**边界条件**设置**窗口。在此示例中，还存在一个出口表面选择。

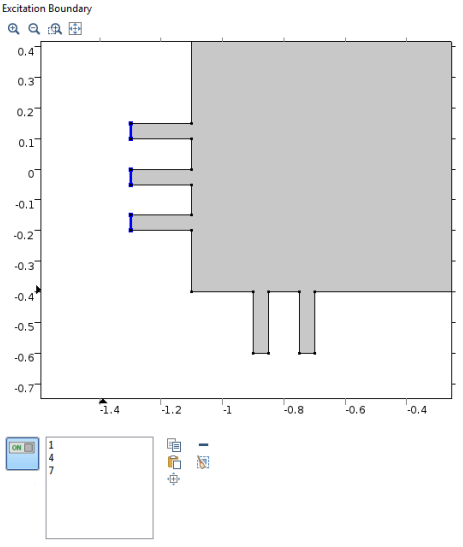


方便起见，除了**手动**选项，还存在一个**所有边界**的快捷方式。

App 开发器中的选择

显式选择类型可以让您基于实体序号分组域、边界、边或点，也是最适合于 App 开发器配合使用的是选择类型。在 App 开发器中，您可以允许 App 的用户交互式地通过**选择输入**或**图形**对象来修改属于**显式**选择的实体。在下面的示例中，内嵌模型包含通过**显式**选择定义的边界条件。**选择输入**和**图形**对象用来让

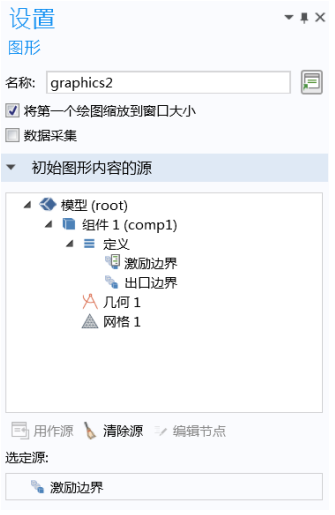
用户选择入射波激励的边界。



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

为了可以通过单击直接选择图形对象，您可以链接一个边界到用于分组边界的显式选择，如下图所示。选择这个显式选择，然后单击**作为源使用**。在下图

中，有两个显式选择，分别是激励边界和出口边界。并且图形对象 graphics2 链接到激励边界。



当一个图形对象按照这种方式直接链接到一个显式选择时，图形对象显示几何，并且用户可以通过在边界上单击形成交互。选中的边界将添加 （或删除）到对应的显式选择。

为了让它能够通过数字进行选择，您可以链接一个**选择输入**对象到显式选择。有关更多信息，请参见 240 页的 “选择输入”。

编辑器工具窗口提供快速方法来添加链接至**显式**选择的**图形**对象或**选择输入**。要获取这些选项，请右键单击编辑器树中的**显式**选择节点。

您可以让全局**事件**通过**显式**选择触发。这允许命令序列或方法在用户单击几何对象、域、面、边或点时运行。有关使用全局事件的更多信息，请参见 116 页的 “事件”和 119 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**复选框以使图形对象交互，这样就可以 （以此为例）单击特定点处的绘图并检索该坐标处的温度数值。在下图的

数据采集的目标栏中，选中了标量双精度变量 Tvalue。该变量在**声明**节点下声明。在运行的 App 中，指针位置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处的 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，请参阅 144 页的 “图形数据”。

输入框

输入框允许用户修改参数或变量的值。在新建表单向导中，当选中参数或变量时，创建了三个表单对象：

- **文本标签**对象，用于参数或变量的描述
- **输入框**对象，用于输入值
- **单位**对象 （如果适用），给出输入值的单位

通过使用**编辑器工具**窗口选择参数或变量，创建相同的三个表单对象。
假设您不使用编辑器工具窗：为了再插入一个输入框，可使用功能区中的**插入对象**菜单并选择**输入框**。在表单编辑器中，将输入框链接至特定参数或变量，

方法是从**源**栏的树中将其选中，然后单击**用作源**。在**设置**窗口的**源**栏中，还可设置**初始值**。下图显示一个输入框的**设置**窗口。

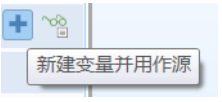


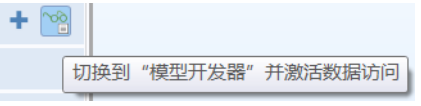
除了参数和变量，输入框还可以使用**信息**节点作为**源**。

初始值的默认设置是**来自源数据**，这意味着如果源是参数，输入框中显示的初始值就是在模型开发器的**参数**节点中指定的参数值。其他的**初始值**选项是**定制值**，允许使用与源不同的初始值。如果清除了**可编辑**复选框，则 App 中显示的**初始值**不允许被修改。

您可以添加带文本的**工具提示**，当您的鼠标在输入框悬停时将显示出来。

源项栏的题头包含两个按钮，用于方便地访问两个工具，它们用于创建更多属性和变量作为输入框的源。





创建新变量并用作源按钮可以用来在**声明**节点下添加新变量，请查阅第 92 页的 “声明” 了解更多信息。**切换到模型开发器并激活数据访问**按钮用来访问如下节所述的底层模型属性。参见第 77 页 “表单编辑器中的模型数据访问”，了解更多**数据访问**的信息。

数据确认

输入框的**设置**窗口中的**数据确认**栏允许您分别对用户输入的单位和值进行验证。



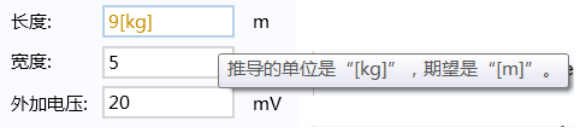
当在新建表单向导中创建输入框时，如果适用就可以使用**添加单位到数**设置项。这个设置项假设用户在输入框中输入数值，但是它也可以处理后面通过 COMSOL 方括号 [] 单位语法实现的带有单位的数。例如，如果**单位表达式**是 mm，则可以使用 1[mm]，以及其他长度单位，例如 0.1[cm]。不兼容的单位将显示**错误信息**。表达式 1.23[mm] 的参数，如果作为源使用将得到单位 mm，并在编辑框中显示初始值 1.23。

单位尺寸检查列表有以下选项：

- **无**
- **与物理量兼容**
- **与单位表达式兼容**

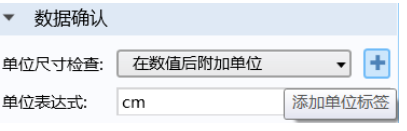
- **在数值后添加单位** （默认）
- **附加来自单位集的单位**

当 App 的用户输入不兼容的单位时 （单位不能转换为**数据确认**设置中指定的单位），表达式的值可以高亮为桔色作为警告。可以通过选择**与物理量兼容**或**与单位表达式兼容**选项来启用这个特征。此外，用户还会看到一个工具提示解释单位不匹配，如下图所示。



如果出现单位不匹配，并且 App 没有执行更进一步的错误控制，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**单位尺寸检查**列表右侧有一个按钮**添加单位标签**。

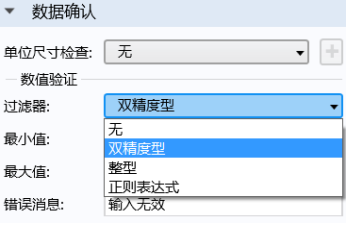


如果输入框的右侧还没有放置单位标签，单击这个按钮将在这里添加一个单位标签。

选项**无**将不提供单位确认功能。

数值验证

选项**在数值后添加单位**、**附加来自单位集的单位**和**无**允许您使用过滤器对输入的数字进行数值验证。



选项**无**的**过滤器**列表有以下选项：

- **无**

- **双精度**
- **整型**
- **正则表达式**

选项**在数值后添加单位**、**附加来自单位集的单位**的**过滤器**列表只允许使用**双精度**和**整型**选项。

双精度和**整型**选项基于**最小**和**最大**值来进行过滤，如果输入值超出了这些值的范围，将显示**错误信息**。您可以使用这些框中的全局参数。如果使用了全局参数，可以定义带单位或不带单位的此类参数。如果使用的是不带单位的全局参数，那么当它们用作**最小**和**最大**值时只考虑这些参数的数值。例如，考虑单位为 cm 的长度参数 L 的输入框的数据验证。此外，假设全局参数 `Lmax` 用作**最大**值。如果您希望 L 的最大值为 15 cm，那么参数 `Lmax` 的以下值将为：15 （不带单位）、`15[cm]`、`0.15[m]`、`150[mm]` 等。

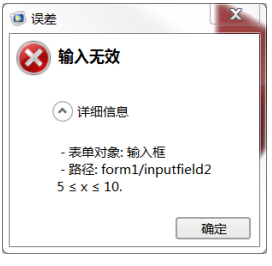
对于**附加来自单位集的单位**选项，**最大**和**最小**值始终与 单位集设置的**初始值**有关。有关单位集的更多信息，请参见 135 页的 “单位集”。

正则表达式选项 （在**单位尺寸检查**设为**无**时可用）允许您使用正则表达式来匹配输入字符串。请参考动态帮助来了解更多关于正则表达式的信息。单击窗口的右上角的帮助图标，并搜索 “正则表达式”。

有关更多高级要求，请注意，输入框内容的几乎所有验证都可以通过调用方法来实现，方法是使用输入框**设置**窗口中的**事件**栏。

错误信息

您可以定制由**错误信息**显示的文字，在开发和调试 App 的过程中，有时候很难判断这些错误由何而来。因此，当使用**测试 App** 按钮时，会显示如下图所示的额外的调试信息。

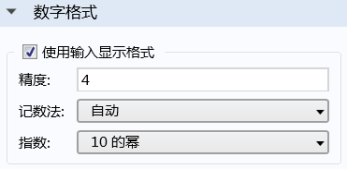


调试信息通常包含表单对象的类型、表单对象的路径，以及错误的原因，例如 5<=x<=10。

当使用运行 App 或 COMSOL Server 启动 App 时将不会显示这些额外的信息。

数字格式

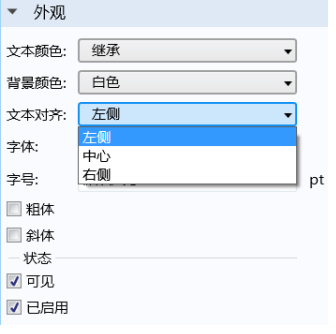
数字格式栏包含**使用输入显示格式**复选框，当选中它时，将启用显示格式的相同类型作为**数据显示**对象。



请参考第 70 页的 “数据显示”，了解更多信息。

外观

除了颜色和字体设置，输入框的**外观**栏还包含**文本对齐**设置，允许文本靠**左**、**中**或**右**排列。



输入框为**可见**或者**已启用**由状态子栏下的复选框控制。有关更多信息，请参见 61 页的 “更改表单对象的已启用和可见状态”。

单位

在**单位**对象的**设置**窗口，您可以设置单位为一个固定的字符串，或者链接到一个输入框。单击单位**标签**列表右侧的**切换到源**按钮，显示表单中链接到的输入框对象。下图显示了一个单位对象的**设置**窗口。



当使用新建表单向导添加输入框时，如果适用则会自动添加单位对象。默认情况下，单位使用 Unicode 渲染。另一种方式是使用 LaTeX 渲染，可通过选中**LaTeX 标记**复选框来实现。然后单位的显示将不再依赖于选定的字体。

文本标签

文本标签对象用于显示表单中的文本。当使用新建表单向导添加输入框时，将自动地添加与参数或变量关联的描述文本作为**文本标签**。其中有一个复选框允

许实现**多行文本**。如果选中这个选项，会启用一个**换行**复选框。下图显示了**文本标签**对象的设置窗口。



可以使用功能区中的**插入对象**菜单并选择**文本标签**来插入更多**文本标签**。

数据显示

数据显示对象用来显示标量和数组的数值，如果有关联单位，将会作为**数据显示对象**的一部分显示出来。

源

在数据显示对象**设置**窗口的源栏，选择一个模型树中的节点，然后单击如下所示的 **作为源使用**按钮。有效的参数、变量，以及属性包括：

- 来自**派生值**节点的输出，例如 **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**节点声明的变量。
- 通过**数据访问**工具开放出来的属性，参见第 77 页“在表单编辑器中的模型数据访问”。
- 下面这些**信息**节点变量，它们位于根节点和每个研究节点：
  - **期望的计算时间**

    这是您在根节点设置窗口的期望编辑框中输入的值。
  - **上一次计算时间**（在根节点下）

    这是上一次计算过的研究所需的计算时间。
  - **上一次计算时间**（在研究节点下）

    这是本研究在上一次计算时耗费的计算时间。

  当第一次启动一个 App 时，将重置上一次计算时间，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，只有**派生值**节点才能生成**数据显示**对象。**声明**的变量和**数据访问**开放出来的常数只能作为**输入框**对象。

当选中**派生值**节点时，基于对应的**派生值**节点变量将创建两个表单对象：

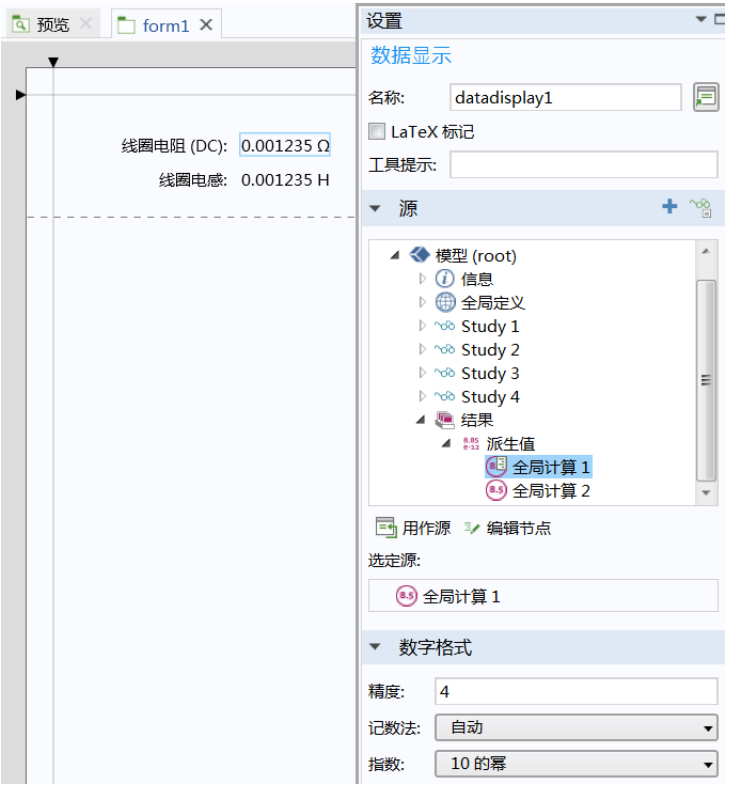
- 变量**描述**的**文本标签**
- 变量值的**数据显示**对象

这些表单对象的设置随后可以被编辑。还可以使用功能区的**插入对象**菜单并选择**数据显示**来插入更多**数据显示**对象。

数字格式

数字格式栏供您设置**精度**、**计数法**，以及**指数**。

下图显示了变量线圈阻抗和线圈导纳的数据显示对象示例，当适用时自动添加格式化的单位标签显示为对象的一部分。



渲染方法

默认情况下，数据显示对象的单位使用 Unicode 渲染。另外，您还可以使用 LaTeX 渲染，通过选择 **LaTeX 标记**复选框来实现。然后数据显示不再依赖于选定的字体。

数组和矩阵的格式化显示只支持使用 LaTeX 渲染，下图显示了一个使用选定的 LaTeX 标记显示的数据显示对象的二维双精度数数组 （参见第 119 页）。

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

您可以添加工具栏提示，当鼠标悬停在数据显示对象上时显示出来。

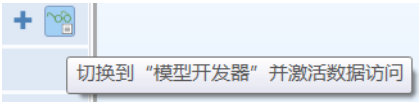
表单编辑器中的数据访问

很多类型表单对象的**设置**窗口中有一个栏允许您选择包含模型树或部分模型树以及部分 App 树的树结构中的节点，或者部分模型树。例如，输入框的**源**栏，或者按钮的**选择要运行的命令**栏。默认情况下，模型树和 App 树中有很多属性不可用，因为一个模型中可能包含数百个甚至上千个属性，显示如此之多的完整列表很不实用。然而，这些 “隐藏的” 属性可以通过**数据访问**技术用于您的 App。

本节以下部分通过示例介绍将**数据访问**用于输入框和按钮的用法。

输入框的数据访问

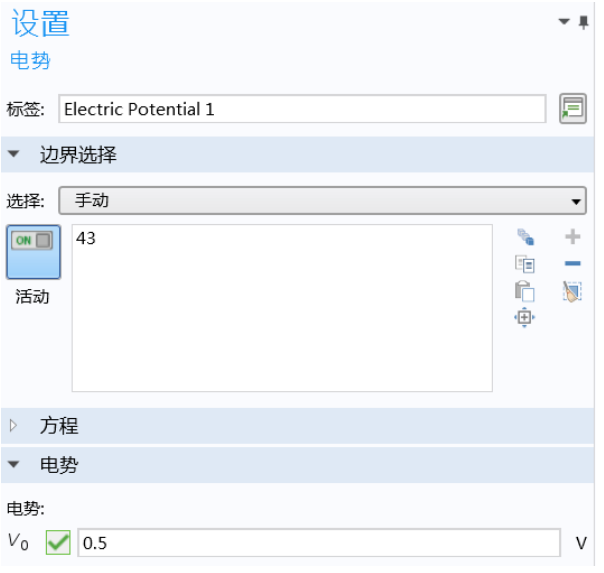
默认情况下，您可以将输入框链接到模型树中**参数**或**变量**节点下定义的参数和变量，或者链接到 App 树**声明**节点下声明的变量。如果需要访问更多的模型树节点属性，可以单击输入框**设置**窗口**源**栏的标题中的**切换到 “模型开发器” 并激活数据访问**按钮，如下图所示。



您也可以从模型开发器的**主屏幕**选项卡中的 **App** 组中访问它。



接下来，当您单击模型树节点时，在各个设置的旁边会出现复选框。下图中显示了选定的**电势**边界条件的复选框：



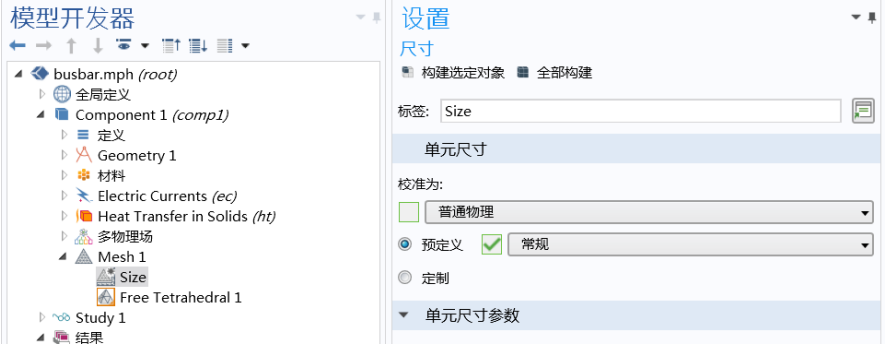
下图显示了输入框的**设置**窗口，现在可选的源中就包含了**电势**。



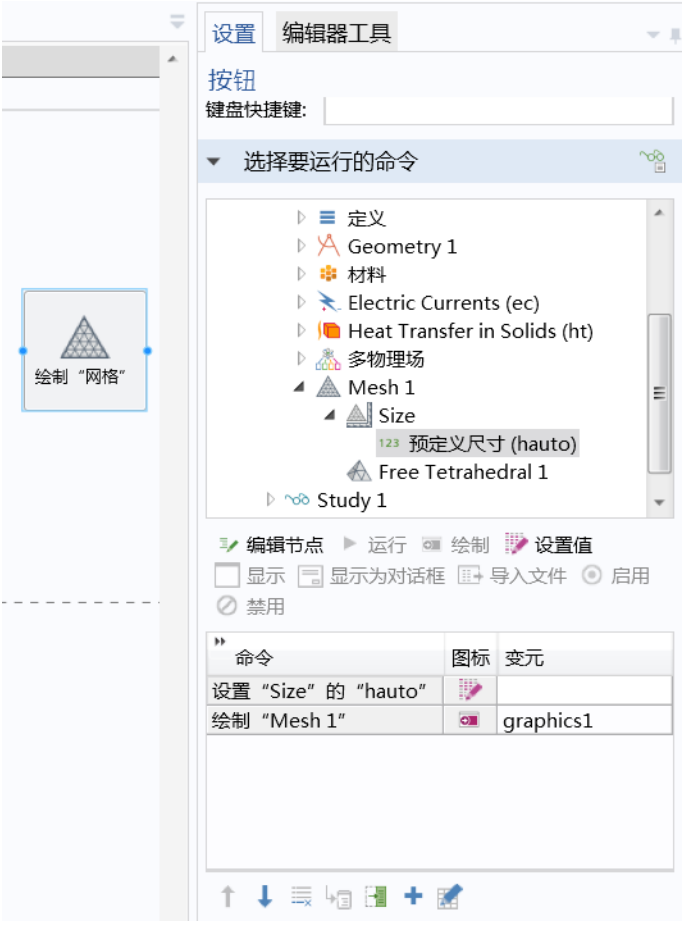
按钮的数据访问

数据访问可以用于按钮设置参数、变量的值，或者模型属性。例如，您可以为预定义的网格单元尺寸创建按钮。下图显示的设置在以下情况下可用：在**网格**

节点的**设置**窗口中，**序列类型**设为**用户控制网格**。在此示例中，**单元尺寸**的**预定义**属性可用，然后被选定。



下图显示用于根据**单元尺寸** > **指定**为**细化**来创建网格的按钮的设置窗口。



在上述例子中，使用**设置值**命令来设置**预定义网格尺寸 (hauto)** 属性的值。属性**预定义网格尺寸 (hauto)** 与之前显示的尺寸节点中的以下设置相对应：

指定网格尺寸	值
极端细化	1
特别细化 - 特别粗化	2-8
极端粗化	9

hauto 属性是一个双精度数，可以取任何正值。对于非整数值，线性插值用于定制网格参数。例如，您可以让滑块对象调整预定义的网格大小。有关滑块对象的更多信息，请参阅 254 页的 “滑块”。

通常情况下，对于单独的模型树属性，您可以通过在改变其值的时候记录代码，然后观察自动生成的代码来快速掌握它们的允许值。参见第 132 页 “记录代码”，了解更多信息。

您还可以使用组合框对象来直接访问从**极端细化**到**极端粗化**的所有可供选择的项。参见第 167 页 “组合框”，了解更多信息。

数据访问的小结

下表总结了各种表单对象和事件，以及菜单、工具栏和功能区项目中可以使用的 **数据访问**。

表单对象、事件，或者项	设置窗口中的栏
输入框	源
按钮	选择运行的命令
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）	选择运行的命令
表单集合	激活面板选择器 平铺或者选项卡
卡片堆	激活卡片选择器
信息卡片堆	激活信息卡片选择器
单选键	源
文本	源
列表框	源
滑块	源
工具栏 （工具栏项）	选择运行的命令
菜单项	选择运行的命令
功能区项	选择运行的命令
事件 （全局）	选择运行的命令 数据改变事件的源

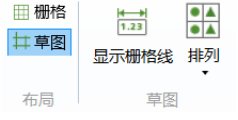
全局事件、菜单、功能区，或者工具栏项在设置窗口中提供了**选择运行的命令**栏，还提供了上面描述的这些功能。全局事件和很多表单对象在设置窗口中提供了**源**栏，以及输入框应用的上面描述的这些功能。参见第 63 页 “图形工具栏”、第 99 页 “主窗口”、第 106 页 “事件”、第 224 页 “表格”，以及第 231 页 “工具栏”，了解更多关于全局事件、菜单、功能区，以及工具栏项的信息。

草图和栅格布局

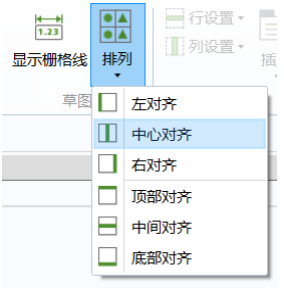
表单编辑器提供两种布局模式用于定位表单对象：草图模式和栅格模式。默认使用草图布局模式，供您使用像素来固定对象的位置和尺寸。使用栅格布局模式可以通过背景格点单元来决定对象的位置和尺寸。在栅格布局模式，表单分解为很多交叉的行和列，每个交叉位置上最多只有一个表单对象。这种模式更适合于设计可变尺寸的用户界面，例如设计一个可在多平台的网页浏览器上运行的 App。

草图布局

通过单击功能区的**布局**组中的**草图**或**栅格**按钮进行两种模式的切换。



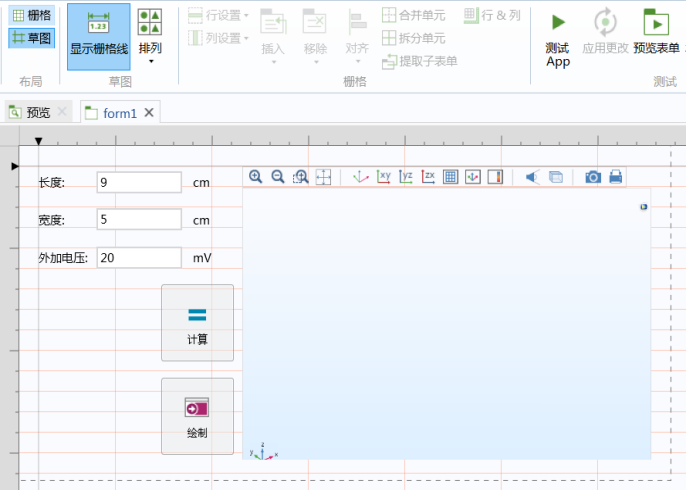
表单标签中的**草图**组有两个选项：**显示格点线**和**排列**。其中**排列**菜单允许您排列表单对象组。



草图格点

显示格点线选项显示草图格点，这样可供对象进行捕获。注意草图布局模式中使用的格点与栅格布局模式中的格点不同。默认的草图布局模式没有显示格点线。当没有显示格点线时，表单对象相对于其他表单对象的位置来进行拖动定位。

如果选中了**显示格点线**选项，表单对象的左上角在拖动时捕获到格点线的交叉点。



在**表单**的设置窗口，您可以修改草图格点的设置：

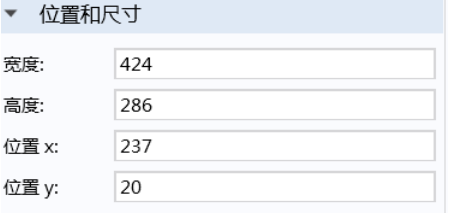
- **列宽度**
- **行高度**

- **对齐格点到边缘**
- **抓取区**
  - 可使用滑动条修改抓取区大小
- **只抓取到格点**
  - 清除这个复选框可以抓取到格点或其他表单对象的位置。



位置和尺寸

草图布局模式基于像素，因此表单对象的定位是由其左上角相对于屏幕的左上角的坐标来测定。随着对象向右移， x 坐标逐渐增大，从上向下移动时， y 坐标增大。您可以在表单对象的设置窗口中的**位置和尺寸**栏设置它的绝对位置。



表单对象分配所需的空间，或通过**宽度**和**高度**值来进行指定。表单对象可以重叠。

栅格布局

通过单击功能区的**布局**组中的**栅格**切换到栅格布局模式。

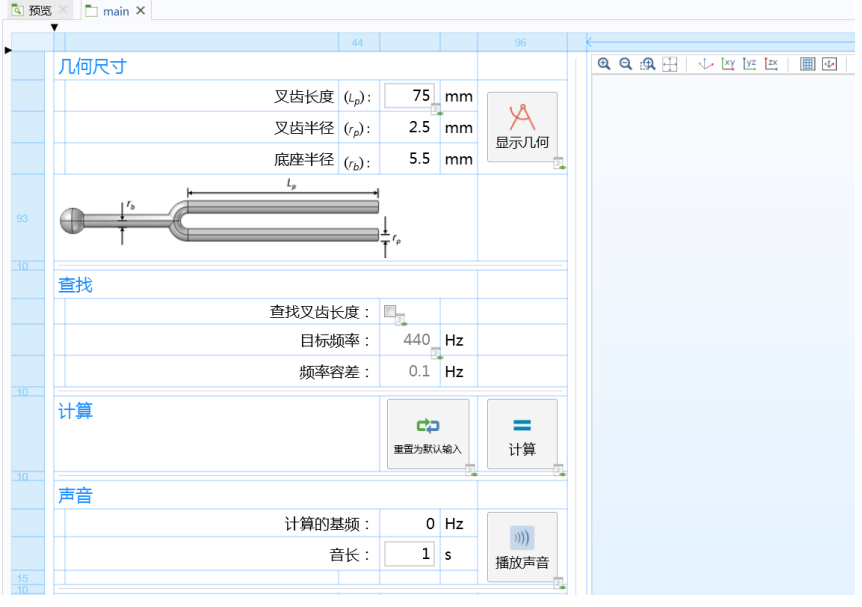


功能区**栅格**组中的按钮和菜单可以让您方便地访问以下命令：

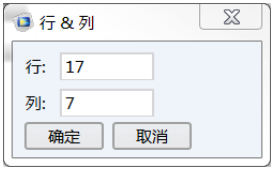
- 修改行和列增长规则 （**适应**、**增长**和**固定**），用来决定当用户界面尺寸发生变化时的布局 （**行设置**和**列设置**）
- 插入或删除行和列 （**插入**和**删除**）
- 在栅格单元中对齐表单对象 （**对齐**）
- 合并和拆分单元 （**合并单元**和**分裂单元**）
- 提取矩形阵列单元为子表单并插入到一个新表单中 （**提取子表单**）
- 定义行和列数 （**行 & 列**）

表单设置窗口和栅格

切换到栅格布局模式后，表单窗口会显示蓝色的格线。



如果想要定义行数和列数，可以单击功能区中的**行 & 列**按钮。



设置窗口的**包含表单对象的栅格布局**栏显示出列宽和行高。



如果需要如下图的表单编辑器所示的交互式选定表单，请单击表单的左上角。

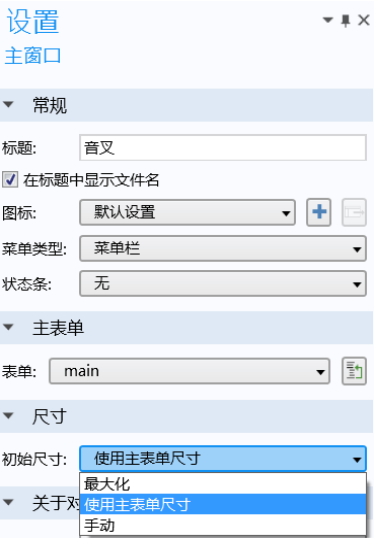


现在会显示一个蓝色的框架。如果需要交互式的修改表单的整体尺寸，您可以拖动它的右侧和底部边框。为此并不需要选定表单。

注意，如果您从草图模式切换到栅格布局模式，所有的行和列将拥有**适应**设置选项，并且不会显示框架的记号线。如果有行和列的**高度**或**宽度**设置为**增长**，则将分别在竖直或水平方向上显示出记号线供重新调整尺寸。

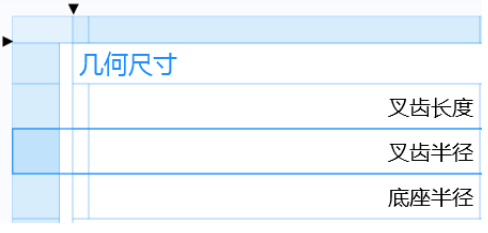
只有当表单的**初始尺寸**设置为**自动**时，交互式调整框架的尺寸才会影响到表单的初始尺寸。如果主窗口的**初始尺寸**设置为**使用主表单**尺寸，框架的尺寸还会影响主窗口的初始尺寸。



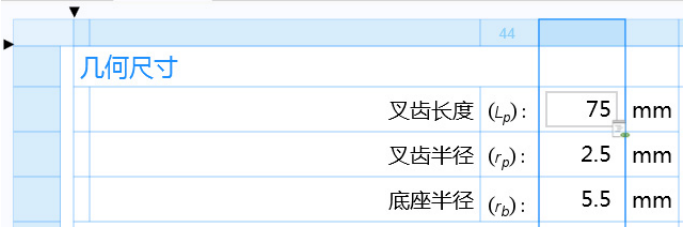


行和列

单击行的最左侧单元可以选中行，它只用来选择行，不能在这里插入表单对象。当选中行时，在功能区标签中会启用**行设置**菜单，以及**插入**和**删除**命令。下图显示第四行为高亮。



类似的，单击列的最上方单元选中列，该单元不包含任何表单对象。下图显示第三列高亮。在这种情况下，功能区标签中启用**列设置**菜单。

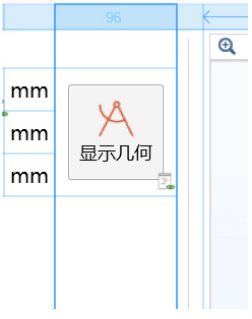


行设置和**列设置**有相同的三个选项：

- **适应**设置行高度和列宽度为表单对象可能给定的最小值
- **增长**设置行高度和列宽度正比于表单的总体尺寸
- **固定**设置行高度和列宽度为指定像素数字的固定尺寸

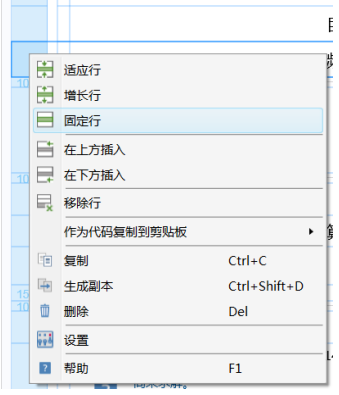


您可以通过拖动格线来交互式地改变行高度和列宽度。



在这种情况下，将显示像素值，并且**行设置**或**列设置**策略将自动地修改为**固定**。

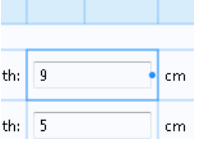
另一种从功能区修改**行设置**或**列设置**的方法是右键单击行或列，然后从菜单中进行选择。



当右键单击时出现的菜单，行或列给出选项包括插入、删除、复制、粘贴，以及复制行或列。

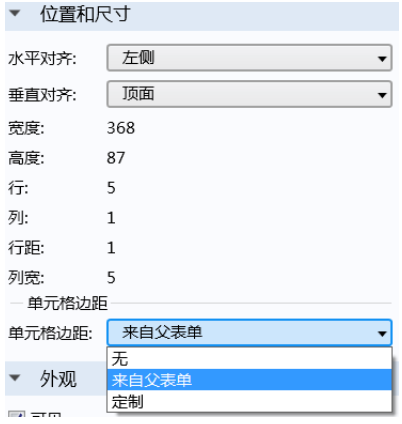
单元

单击一个独立的单元来选中它，单元被选中后显示深蓝色格线。



您可以选择**合并单元**和**分裂单元**来调整单元尺寸和布局到满足您的表单对象。

在栅格布局模式，您可以指定表单对象之间，以及其包含的单元之间的边距。



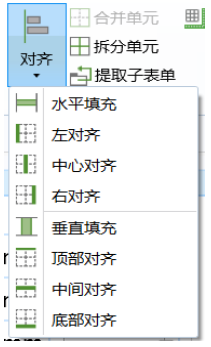
在这个表单对象的设置窗口，**位置和尺寸**栏有以下这些**单元格边距**的选项：

- 无
  - 没有单元格边距。
- 来自父级表单 （默认）
  - 表单的设置窗口中指定的边距，参见第 96 页 “继承列和单元格边距”。
- 定制
  - 定制只用于本表单对象的边距。

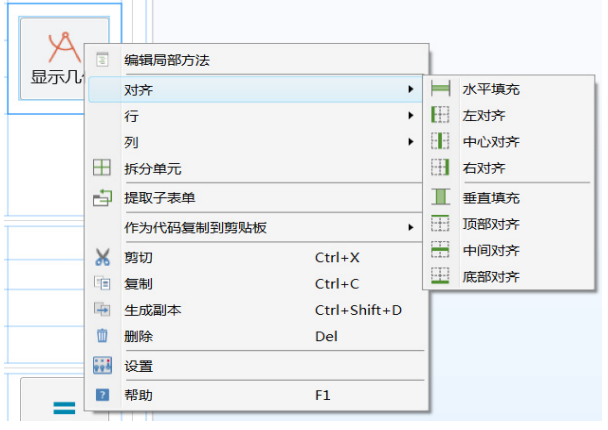
如果**水平对齐**和**垂直对齐**设置为**填充**，列或行的增长策略允许表单对象尺寸可变，您可以分别指定最小宽度或高度。最小尺寸可以设置为**手动**或**自动**。其中，**手动**选项可以指定最小尺寸的像素值；**自动**则允许最小尺寸为 0，除非表单对象的内容需要更大的值。最小尺寸设置用来在运行时确保滚动条可以在表单对象缩小到小于最小尺寸之前显示出来。

对齐表单对象

对齐菜单给您提供选项用于对齐单元中的表单对象，您还可以让表单对象动态地水平或竖直填充单元。

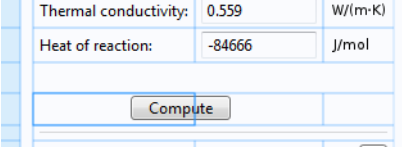


您还可以右键单击表单对象，然后从上下文菜单中选择。



拖放表单对象

您可以拖放表单对象来移动它们。单击并选中一个表单对象，然后将它拖到另一个没有被其他表单对象占用的单元。

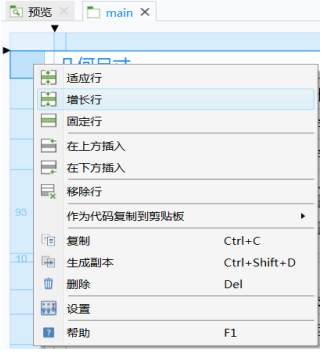


如果您将对象拖到另一个已经被占用的单元，则两个对象交换位置。

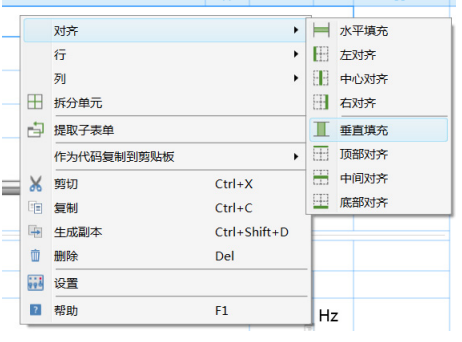
自动调整图形对象的尺寸

为了使 App 的图形对象的尺寸可以调整，请按照下列步骤操作：

- 修改包含图形对象的表单的布局模式为栅格布局模式
- 修改涵盖表单对象的行的**高度**设置为**增长**。为此请单击该行最左侧的列选中该行，然后修改表单的**设置**窗口中的**高度**设置。或者，右键单击，然后选择**增长行**



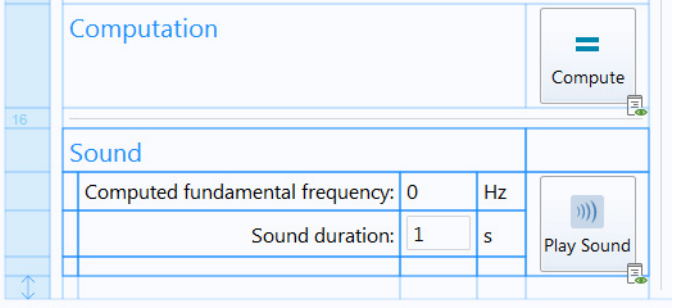
- 修改涵盖图形对象的列的**宽度**设置为**增长**，为此，请右键单击该列的最上面的行，然后选择**增长列**。
- 选择图形对象，然后修改**水平排列**和**垂直排列**为**填充**。您可以从设置窗口或右键单击图形对象，然后选择**对齐** > **水平填充**和**对齐** > **垂直填充**。



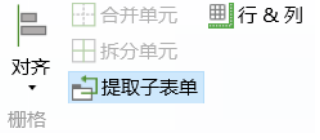
按照上面的操作步骤，您可以很方便地通过执行栅格布局模式的操作使得图形对象的尺寸可调整，例如添加、空行和列，以及合并单元，等。如果您已经处于栅格布局模式，则图形对象默认在两个方向均为**填充**。

提取子表单

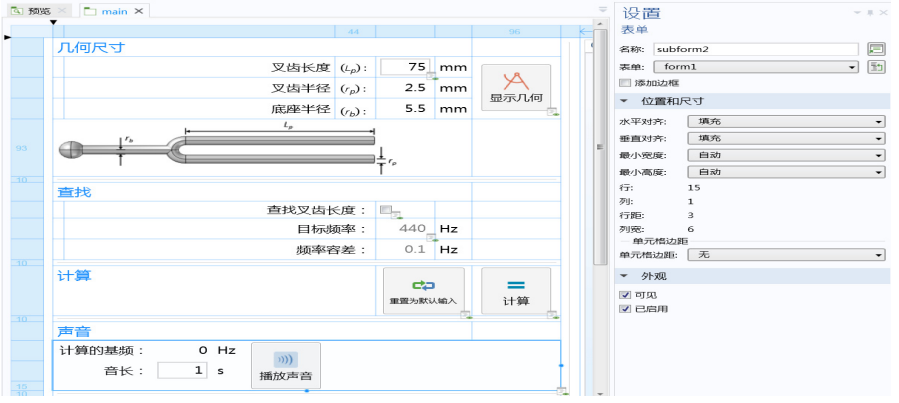
您可以选择表单中的矩形单元阵列，将它移动到一个新表单。首先使用 Ctrl+单击或 Shift+ 单击选择单元。

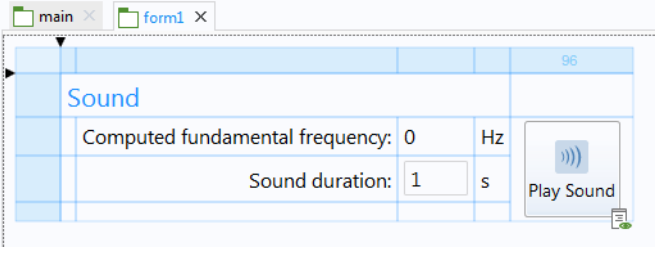


然后单击功能区中的提取子表单按钮。



这个操作使用选定的单元创建一个新表单，并且使用表单对象取代**表单**中的原始单元。在子表单的**设置**窗口，**表单**引用指向包含原始单元的新表单。





继承列和单元格边距

通过使用子表单，您可以有效地组织用户界面，例如，编组一套输入表单。下图显示了包含两个子表单的 App，分别是**梁尺寸**和**加强筋**。



请查阅前面的章节和第 168 页的 “表单”来了解更多关于添加子表单的信息。

当垂直对齐子表单时，您可能想要确保所有列宽度相等，如上例所示。为此您可以在子表单的**设置**窗口中使用**继承列**选项。下图显示了**名称**为 geometry_beam 的**梁尺寸**子表单 （左）和**名称**为 geometry_rebars 的**加强筋**子表

单的部分设置窗口。其中 geometry_rebars 子表单的继承列选项设置为 geometry_beam。





在子栏**单元格边距**，您可以指定添加在表单对象之间，以及它们包含的单元格边界之间的水平和垂直边距。这些设置将影响包含在表单中的所有表单对象，如果它们各自的**单元格边距**设置为**来自父级表单**，参见第 90 页 “单元”。

App 之间进行复制

您可以在多个同时运行的 COMSOL Multiphysics 会话之间复制和粘贴表单和表单对象。您还可以在一个会话中从当前 App 复制和粘贴到新加载的 App。

在栅格布局模式，可以在会话之间复制一个单元、多个单元、整行，以及整列。

当您在不同 App 之间复制和粘贴表单和表单对象时，复制的对象可能包含到其他表单和表单对象的引用，这样的引用可能或可能没有意义。当粘贴对象时应用规则的更多信息，请查阅 第 208 页的 “附录 B — App 之间进行复制”。

当在不同 App 之间复制和粘贴对象时，如果检测到可能存在的兼容性问题，会显示出一个信息对话框。在这种情况下，您可以选择取消粘贴操作。

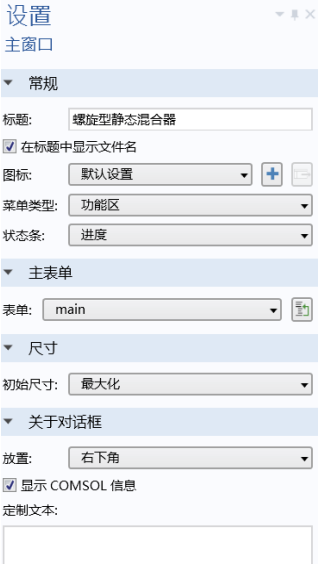
主窗口

在 App 树中，**主窗口**节点代表 App 的主窗口，同时它也是用户界面的顶层节点。其中包含窗口布局，主菜单明细，以及可选的功能区明细。

通用设置

设置窗口中包含**通用**栏，用于：

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



标题是 App 的主窗口顶部的文本，**图标**则显示在这些文本的最左侧。默认情况下，**标题**与用来创建 App 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，请保持勾选**在标题中显示文件名**复选框。

在**图标**列表中，从库中选择一个图片或者从本地文件系统中添加一个图片（*.png）到库，并用作图标。如果您添加一个新图片，它将会被加到图片库，并内嵌到 App。您还可以通过单击**添加图像到库并在这里使用**按钮右边的**导出**按钮来导出图标。

App 树的**主窗口**节点有一个子节点，命名为**菜单条**，使用**菜单**类型的设置，您可以将这个子节点从**菜单条**改为**功能区**。

状态栏列表控制在状态栏上的显示内容：选择**进度**可以显示进度条 （默认），或者**无**。注意，您还可以通过使用方法创建定制的进度条。

主表单

主表单栏包含主窗口显示的表单的引用信息。当使用表单集时，这个设置相当重要，因为它决定了当打开 App 时，首先将哪个表单显示为主窗口。

尺寸

在**尺寸**栏中，**初始尺寸**设置决定了当 App 启动时，首先显示的主窗口的尺寸。其中包含三种选项：

- **最大化**使得当运行 App 时窗口最大化。
- **使用主表单尺寸**将使用主表单的尺寸，参见第 37 页 “独立表单设置窗口”。主表单在**主表单**栏中定义，这个选项还添加主窗口自己所需的尺寸，包括：窗口框架及标题栏、主菜单、主工具栏，以及功能区。根据菜单类型为**菜单栏**或**功能区**的不同，这个尺寸自动计算出来。
- **手动**让您输入宽度和高度的像素尺寸。在这种情况下，不会向宽度和高度添加任何值。当使用这个选项时，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。

要了解更多**使用主表单尺寸**选项的信息，请查阅第 86 页 " 表单设置窗口和单元格”。

关于对话框

关于对话框栏包含**关于此 App** 对话框 （包含法律信息）定制部分的设置。通过**放置**选项，您可以选择**自动**、**文件菜单**、**功能区**或**右下角**。**右下角**选项会向 App 用户界面右下角的**关于此 App** 对话框放置一个超链接。在选中的情况下，**显示 COMSOL 信息**复选框将显示 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将显示在对话框中法律文本的上方。在**定制文本**框中，包含 `http` 或 `www` 的词将被理解为超链接 （如果可能）。例如，`http://cn.comsol.com` 或 `cn.comsol.com` 将被替换为超链接。

菜单条和工具栏

菜单条节点可以包含代表主窗口顶部菜单的**菜单**子节点。



在**菜单条**选项中，您还可以选择添加**工具栏**。**工具栏**节点和**菜单**节点拥有相同类型的子节点。

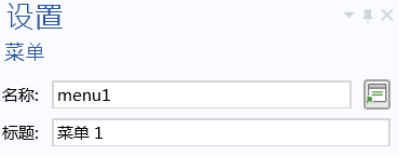


菜单，项和分隔线

菜单和**工具栏**的子节点可以是**菜单**、**项**、**切换项**或者**分隔线**，如下图所示：



菜单节点的设置项包括**名称**和**标题**。



菜单节点可以有子**菜单**节点，代表子菜单。

分隔线显示为菜单和条目组之间的水平线，没有设置项。

项的**设置**窗口与按钮类似，也包含一个命令序列。与按钮类似，条目可以关联到文本、图标，以及键盘快捷键。请查阅第 47 页 “按钮” 了解更多信息。类似地，**切换项**节点的**设置**窗口与切换按钮的类似。

下图显示了与创建报告方法相关联的**项**的**设置**窗口。



下图显示一个有**文件**菜单的 App 的示例。



当在 COMSOL Desktop 环境中运行 App 时，将总是会自动地显示一个**关闭 App** 的菜单项。

下图显示有**另存为**项的**设置**窗口。



您可以通过方法启用和禁用功能区、菜单，以及主工具栏。更多信息请查阅第 259 页 “附录 E — 内置方法库”。

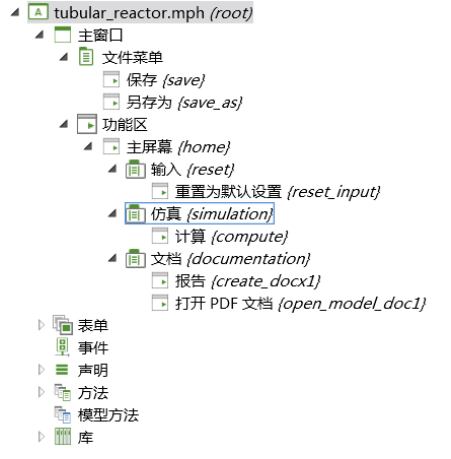
功能区

您可以选择在主窗口添加**功能区**，而不是**菜单条**。**功能区**节点包含带有一个或多个标签中放置工具栏的功能区明细，对于**功能区**选项，可以直接在主窗口节点下使用**文件**菜单。

功能区标签和功能区栏

功能区节点的子节点是**功能区标签**，后者的子节点是**功能区栏**。**功能区栏**的子节点可以是**项**、**切换项**、**菜单**，或者**分隔线**。

项和**菜单**提供了之前描述的**菜单条**和**工具栏**相同的功能。任意功能区栏的子节点添加的**分隔线**是 App 中**条目**和**菜单**组之间的垂直线。然而在 App 树中，**分隔线**显示为水平线，如下图所示。





事件

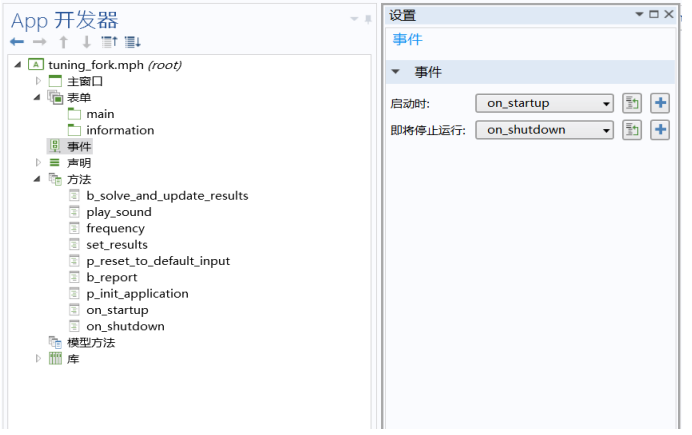
事件是 App 采取一个或多个操作所需的信号的任意活动 （例如，单击按钮、键入了键盘快捷键、加载表单，或者修改变量值等）。每个操作可以是之前描述的一个命令序列，或者也可以包含方法的执行。这些方法可以是关联到特殊表单对象的局部方法，或者可以在 App 的任意位置初始化的全局方法。全局方法在 App 树的**方法**节点列出，局部方法在表单或表单对象相关联的**设置**窗口中定义。当表单对象有关联的方法时，可以通过在对象上执行 Ctrl+Alt+ 单击来打开进行编辑。如果在没有方法的对象上执行 Ctrl+Alt+ 单击，则创建一个与该对象关联的新局部方法，并打开此方法以进行编辑。

初始化这些操作的事件可以是全局或局部类型，全局事件在 App 树的**事件**节点下列出，并且包含可以被各种数据实体的变化所触发的事件，这些数据实体包括全局参数或字符串变量等。全局事件还可以关联到 App 的启动和关闭。局部事件，与局部对象类似，在相关联的表单或表单对象的**设置**窗口中定义。

只要源数据发生变化，就会触发**事件**节点，无论它是由方法、表单对象，还是由其他任意途径引起变化。与表单对象关联的事件只能当用户修改表单对象的值时触发。

启动和关闭时的事件

全局或局部可以关联到 App 的启动 （**启动时执行**）和关闭事件 （**关闭时执行**）。要访问这些事件，请单击 App 树中的**事件**节点。



以下情况会触发关闭事件：

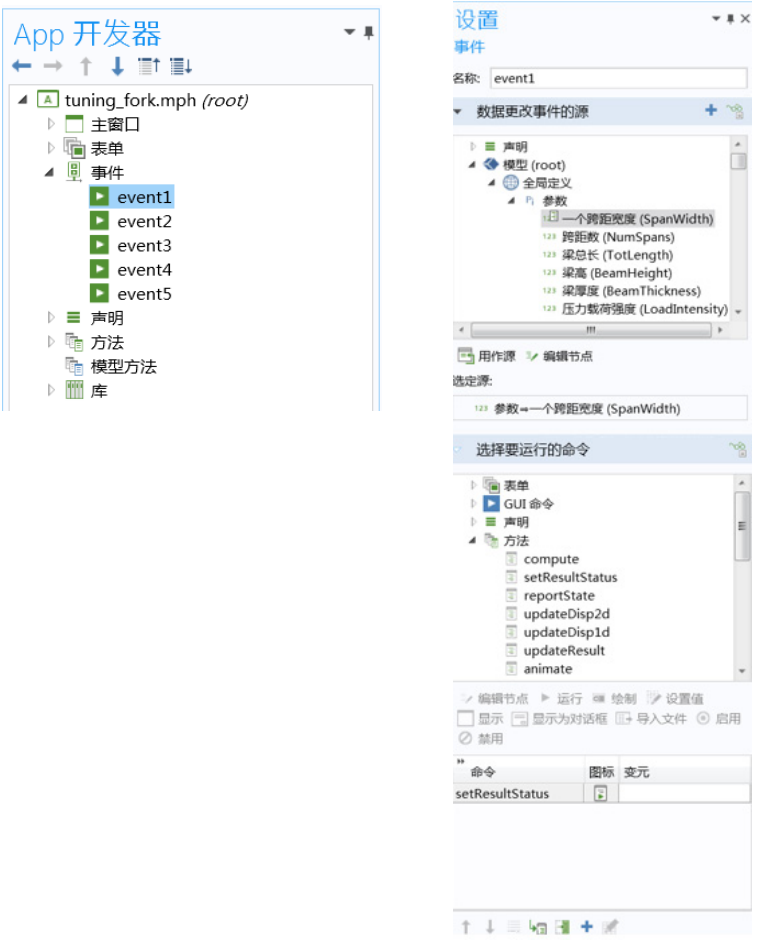
- App 的用户通过单击 App 窗口右上角的**关闭 App** 图标关闭 App 窗口
- 表单对象执行**退出 App** 命令
- 运行了使用命令 exit() 的方法

用法示例：在关闭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，在关闭时运行的方法可能会返回布尔值 true，从而取消关闭。

请注意，用于初始化图形的方法，例如**缩放到窗口大小**，需要作为表单的**加载时执行**事件来运行，而不是作为全局**启动时执行**事件。

全局事件

右键单击**事件**节点，选择**事件**来添加事件到 App。事件会监听运行的 App 中的变化，如果发生变化，就将运行一个命令序列。下图中，当字符串变量 SpanWidth 的值发生变化时，将运行方法 setResultsStatus。



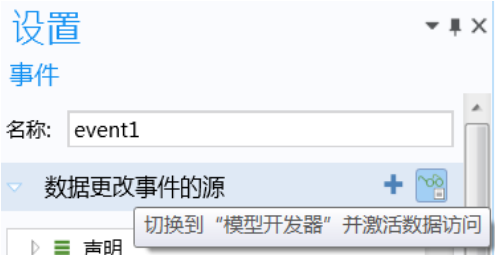
注意，因为这种类型的事件是全局性质，没有关联到特定的表单，当引用图形对象时，必须给出完整的路径：/form1/graphics1。

下面两节描述事件的**设置**窗口中用到的选项。

数据更改事件的源

本节描述 App 开发器窗口的树中部分视图，节点代表数据或有类似子节点。

您可以通过单击**数据更改事件的源**栏的题头上的**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。



请查阅第 110 页 “方法编辑器中的模型数据访问”了解更多信息。

请注意，**显式**选定对象也允许作为**数据更改事件的源**。这允许命令序列或方法在用户单击几何对象、域、面、边或点时运行。下图显示一个全局事件的对话

框，该全局事件在用户更改名为**出口**的**显式**选定对象的内容时将打开一个表单面板作为对话框。



选择运行的命令

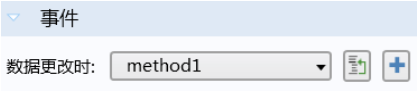
在**事件**的**设置**窗口，**选择运行的命令**栏与按钮的设置类似，同样用于定义一个命令序列。请查阅第 47 页 “按钮” 了解更多信息。

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，但是针对表单或单个表单对象而定义。这些事件没有关联到命令列表，而是直接指向一个全局或局部方法。

由数据更改触发的事件

对于确切类型的表单对象，您可以指定当数据发生变化时运行的方法。这个设置在表单对象的**事件**栏，如下图所示。



数据更改时执行下拉列表包含**无** （默认）， App 树中**方法**节点下任意可以使用的方法，以及局部方法 （可选）。

这种事件类型支持的表单对象有：

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

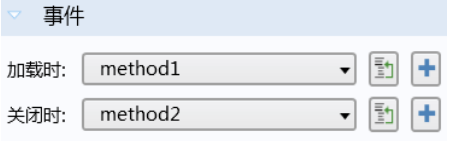
按钮与鼠标单击触发事件关联。菜单、功能区和工具栏项与选择触发的事件关联。相应的操作是在对象或项目的**设置**窗口中定义的命令序列。请查阅第 47 页 “按钮” 了解更多信息。

选择多个表单对象

您可以通过使用 Ctrl+ 鼠标单击，然后设置运行的方法，同时指定多个表单对象的**当数据变化时执行**事件。例如，通过这种方法，您可以快速指定由任意选定的表单对象初始化的数据变化事件运行一个方法，通知用户绘图和输出为无效。此功能并非适用于所有表单对象组合。

由加载或关闭表单触发的事件

当加载 （**加载时执行**）或关闭表单 （**关闭时执行**）时可以运行方法。



这种类型的事件可用于表单的**设置**窗口，通常用于表单显示为对话框或激活在表单集中用作面板的表单时。请注意，用于初始化图形的方法，例如缩放到窗口大小，需要作为表单的**加载时执行**事件来运行，而不是作为全局**启动时执行**事件。

使用局部方法

事件可以调用没有显示在 App 树中的局部方法。请查阅第 121 页 “局部方法”，了解更多信息。

声明

App 树中的**声明**节点用来声明全局变量和对象，添加除了模型中已定义项之外的全局参数和变量。**声明**节点下定义的变量可以用于表单对象和方法。在表单对象中，这些变量用于保存在其他表单对象或方法中用到的值。对于不需要在表单对象和方法之间传递，而只是用于方法内部的变量，不需要在**声明**节点中进行声明。在方法中，**声明**节点下定义的变量可以在全局范围内使用，并且可以直接通过它们的名称来使用。参见第 151 页 “访问全局参数”，了解如何访问在模型树中定义的全局参数的信息。

以下是不同类型的声明：

- **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**

右键单击**声明**节点或者通过功能区来访问这些声明类型。

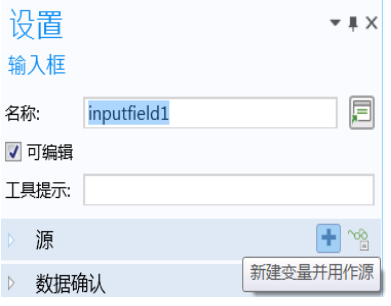


请注意，只有单击表单对象**设置**窗口中**名称**旁边的**创建快捷方式**按钮或对选定的表单对象使用 Ctrl+K 才能创建**快捷方式**。

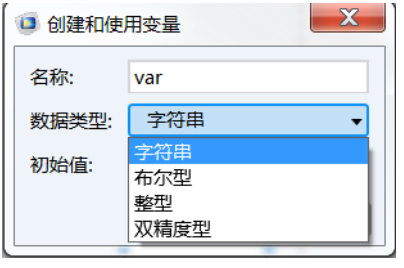
此外，头三个声明类型可以是以下数据类型：

- **字符串**
- **布尔变量**
- **整型**
- **双精度**

除了右键单击**声明**节点，您还可以单击很多类型的表单对象设置区中的**源项**栏中的**创建新变量并用作源**按钮。



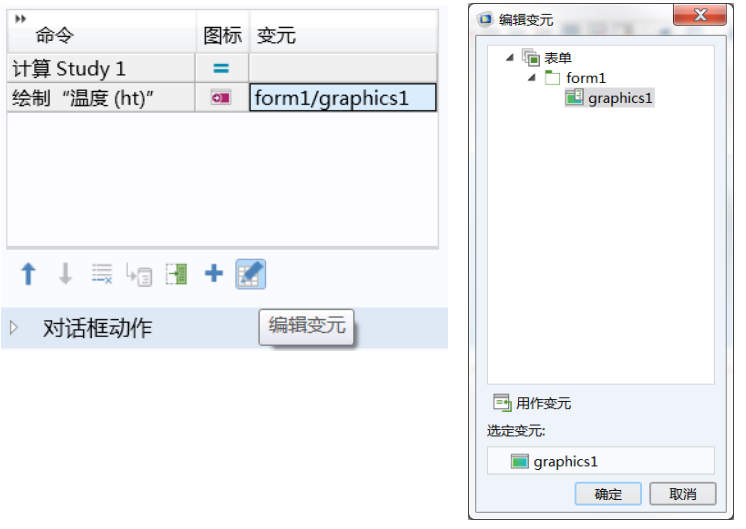
这将打开一个对话框，让您快速声明一个标量变量。



使用声明作为命令的输入变元

命令序列中使用的命令可以有输入变元，例如一个按钮可以带入一个输入变元。参见第 47 页 “按钮”，了解更多信息。

下图显示了一个包含输入变元 form1/graphics 的绘制温度命令的命令序列。



您可以使用声明作为命令的输入变元。

如果需要使用标量变量、一维数组或二维数据作为输入变元，您可以使用相应的变量名称。通过索引可以访问数据的单个元素，或者二维数据的某行或列。例如，您可以使用 `my_variable(1)` 访问一维数组 my_variable 的第一个分量。二维数据的元素可以通过使用两个索引提取为标量，例如， `my_matrix(2,3)`。索引可以是其他声明的变量，例如， `my_variable(n)`。

对于需要图形对象作为输入变元的命令，只允许使用字符串类型的声明，如果有必要，还需要加上正确的索引。如果有一个图形对象命名为 graphics1，且还有一个字符串命令为 `graphics1`，将使用字符串声明的内容。例外的情况是使用单引号，例如’ `graphics1`’，此时将使用图形对象 `graphics1`。这个规则同样适用于其他的命令和输入变元组合。

变量名称

变量名称是不包含空格的文本字符串，其中可以包含字母、数字和下划线。不允许使用保留名称 root 和 parent，以及 Java® 编程语言的关键字。

标量

标量声明用于定义用作字符串、布尔变量、整型或双精度的变量。

字符串

标量字符串变量类似于全局参数或者模型中的变量，但是存在一定差异。模型中的参数或变量有很大限制，它的值必须是一个有效的模型表达式，而标量字符串变量则没有这样的限制。您可以通过方法中的转换函数，使用字符串变量来表征双精度数、整数，或者布尔变量。参见第 260 页 “转换方法”，了解更多信息。您可以在很多表单对象中使用字符串变量作为源，例如输入框和组合框、卡片堆，以及列表框。

下图显示了字符串变量 graphics_pane, email_to 和 solution_state 的**设置**窗口。



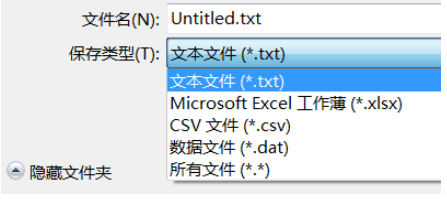
字符串声明，以及其他声明，可以通过**变量列表**表格下方的**从文件加载**和**保存到文件**按钮加载到表单或者保存到文件。

从文件加载和**保存到文件**按钮用来加载 / 保存以下文件格式：

- 文本文件 （.txt）
- Microsoft®

- Excel® 工作薄 （.xlsx）
  - 需要LiveLink™*for*Excel®
- CSV 文件 （.csv）
- Data 文件 （.dat）

下图显示了可供选择的这些文件格式的下拉列表。



下图描述如何使用声明的字符串，图中显示为字符串变量 viewCard 用作卡片堆对象的源 （**激活卡片选择器**）时的**设置**窗口。



请查阅 第 197 页 “卡片堆”，了解更多信息。

布尔变量

您可以使用布尔变量作为选择框、其他表单对象和方法的源。一个布尔变量可以有两个状态：true 或 false，默认为 false。

下图显示了两个布尔变量的声明。



代码示例

在下面的代码示例中，布尔变量 bvar 的值由选择框控制。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

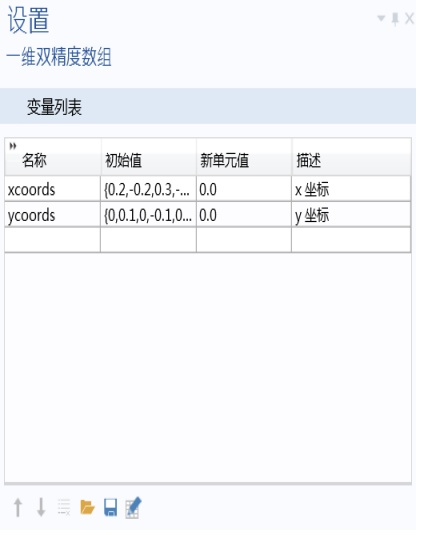
除了需要分别满足值为整型或双精度外，整型和双精度变量与字符串变量类似。



一维数组

一维数组节点声明一个或多个指定的字符串、布尔变量、整型或双精度数组，您可以从表单对象和方法中进行访问。一维数组中的单元数量没有严格的限制，例如您可以使用一个一维数组储存表格中的一列，其中行数可变。设置窗

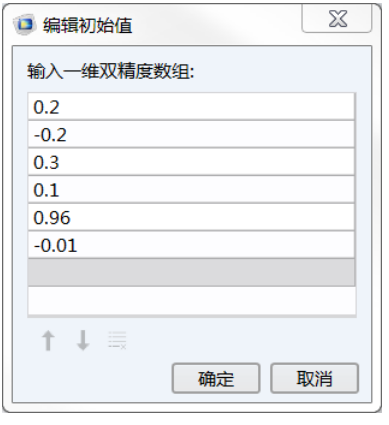
口中包含一个表格，每行可以指定一个变量数组。在下图中，声明了两个双精度数组，分别是 xcoords 和 ycoords。



列**新单元值**用于从表单对象添加一行时赋给新单元的值。字符串、布尔变量，以及整数数组的功能与双精度数数组类似。

初始值

初始值可以是任意长度的一维数组，如果需要编辑初始值，请单击**变量列表**下方的**编辑初始值**按钮。这将打开一个对话框，可以键入每个分量值。下图显示了一个一维双精度数数组的初始化示例。



数组语法

数组的定义必须以花括号作为起始和结束 （{ 和 }），每个单元之间必须以逗号隔开。如果您需要在数组单元中使用特殊字符 （例如，空格和冒号），可以用单引号隔开 （'）。下表显示了一些一维数组的示例：

数组语法	结果数组
{1, 2, 3}	一个含有单元 1、 2 和 3 的三单元数组
{}	一个空数组
{'one, two', 'three by four'}	一个包含特殊字符的两单元数组
{{1, 2, 3},{'one, two', 'three by four'}}	一个包含一个三单元数组和一个二单元数组的二单元数组

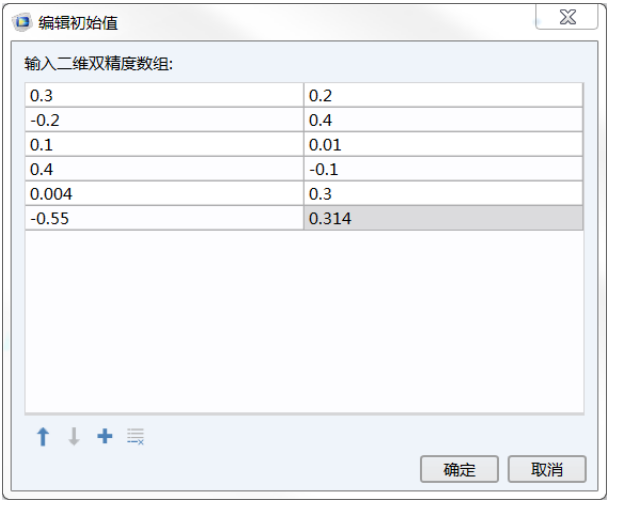
二维数组

二维数组节点声明一个或多个二维数组，您可以从表单对象和方法中进行访问。下图显示了一个声明为 xycoords 的二维双精度数数组。



初始值

默认 （或初始）值可以是一个任意尺寸的二维数组。如果需要编辑初始值，请单击**变量列表**下方的**编辑初始值**按钮。这将打开一个对话框，其中可以输入每个分量的值。下图显示了一个二维双精度数数组的示例。



数组语法

下表显示了一些二维数组的示例：

数组语法	结果数组
{{}}	一个空二维数组
{{'5','6'},{'7','8'}}	一个 2X2 的字符串矩阵
{{1, 2, 3}, {4, 5, 6}}	一个 2X3 的双精度数矩阵

对于二维数组，行数对应于第一个索引，因此 {{1,2,3},{4,5,6}} 相当于矩阵：

1 2 3

4 5 6

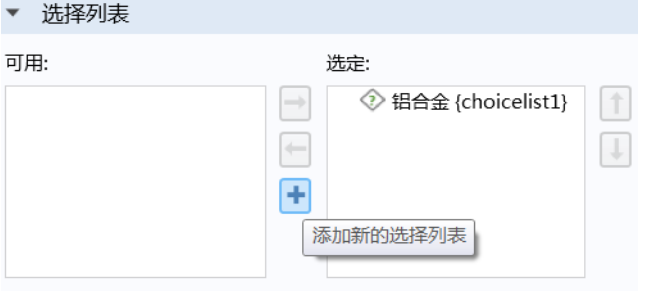
假设上面这个 2X3 的矩阵储存在一个二维数组变量 arr，则单元 arr[1][0] 等于 4。

选择列表

选择列表节点包含可用于组合框、单选键，或者列表框的列表。选择列表的**设置**窗口中包含**标签**、**名称**，以及一个包含**值**和**显示名称**列的表格。在第一列输入属性值 （**值**），并在第二列 （**显示名称**）输入对应的显示给用户的文本（例如，在组合框列表中）。**值**总是解释为字符串。下面的示例中，当从组合框回车时， mat1 将转变为 “mat1”。



除了通过右键单击**声明**节点来创建选择列表，您还可以在使用这种列表的表单对象的设置窗口中单击**添加新选择列表**按钮，如下图所示。

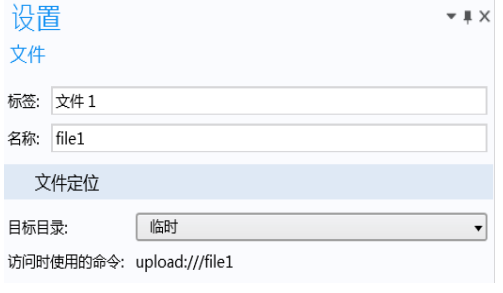


激活条件

您可以右键单击**选择列表**节点，添加一个**激活条件**子节点。使用激活条件可以视变量值而定，在两个或多个选择列表之间进行切换。第 148 页 “使用组合框改变材料”就是一个典型的用到激活条件的选择列表示例。

文件

文件声明主要用于方法代码中使用内建方法 importFile 时的文件导入。参见第 274 页 “文件方法”，了解更多方法 importFile 和其他文件处理方法的详细信息。此外，**文件**声明节点下的条目还可以用于**文件导入**对象。下图显示了一个文件声明的**设置**窗口。



用户选定的文件可以在表单对象或方法中使用语法 upload:///file1，upload:///file2 等进行引用，其中文件名句柄 （file1， file2，等）可以用来引用由用户在运行时选定的实际文件名。

参见第 236 页 “附录 C — 文件处理和文件方案语法”，了解更多关于文件声明和处理的信息。

单位集

单位集节点包含可由 组合框、单选键或列表框用于更改单位的列表。单位集的**设置**窗口包含两个栏，即**单位组**和**单位列表**。



单位组表中的每行都是一个单位组，代表在 App 用户界面的上下文中具有特殊意义的单位的集合。每列代表标有**值**和**显示名称**的一组单位。

单位列表表中的每行都是一个单位列表，列包含具有相同尺寸的单位。例如，mm、 cm、 dm、 m、 km。**单位列表**表的题头时在**单位组**栏中定义的**名称**和**显示名称**。单位列表可指定引用**单位集**的表单对象在运行 App 时可切换显示的单位。

上图展示了**单位集**在允许切换显示公制单位和英制单位的 App 中的使用。在此示例中，定义了两个单位组：国际单位制和英制单位制。**单位集**的**标签**已更改为单位制。

值列包含字符串值，代表当前选择的单位组。这些字符串值可以使用相应方法进行操作。**显示名称**列是显示在用户界面中的字符串。**初始值**列表包含默认单位组 （上述示例中为国际单位制）。

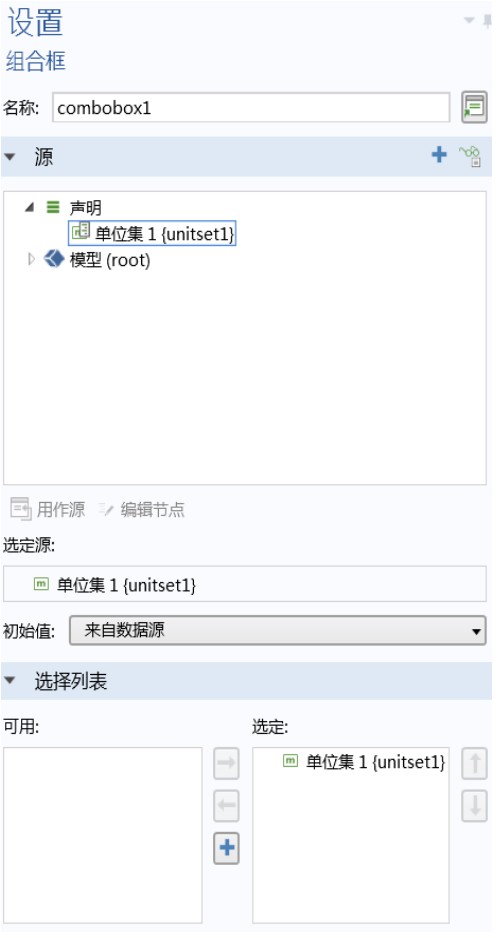
在上述示例中，**单位列表**表有三列：**名称**、**国际单位制**和**英制单位制**。**国际单位制**和**英制单位制**列基于**单位组**栏中的组动态地创建。表中的每行分别与允许的长度单位和电势单位相对应。

下图显示一个示例 App，其中组合框表单对象用于在**国际单位制**和**英制单位制**单位组中进行选择。





下图显示将上述示例的**单位集**用作**源**的组合框的**设置**窗口。



这样，**单位集**可代替**选择列表**用于创建单位选择的组合框。不使用组合框，以类似方式使用列表框或单选键对象。

下面两个图显示**长度**和**应用电压**的两个输入框的相应**设置**窗口。



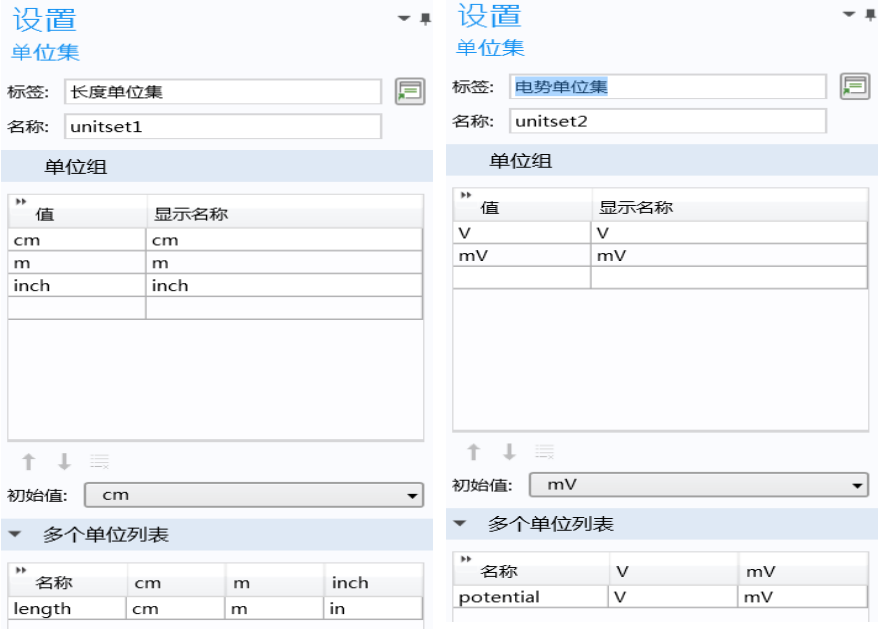


单位尺寸检查设为**附加来自单位集的单位**。**单位集**设为**单位制 {unitset1}**（此示例中使用的是**单位集**声明的用户定义标签）。单位集分别设为**长度**和**电势**。使用**附加来自单位集的单位**时，**数值验证**栏（位于**数据验证**下方）指向**单位集**的**初始值**；在此示例中，分别为 **cm** 和 **mV**。**最小**和**最大**值在 App 运行时自动缩放，单位由 App 用户更改。有关输入框对象设置的更多信息，请参见 76 页的 “输入框”。

下图分别展示使用两个**单位集**声明分别设置长度和电势单位。



下图显示单位集声明的相应**设置**窗口。



请注意，在此示例中，通过使用三个**单位集**声明，您可以对**长度**和**宽度**输入框分别进行长度单位设置。下图显示的示例中，三个组合框已用于替换单位标签，每个组合框使用单独的**单位集**声明作为其源。

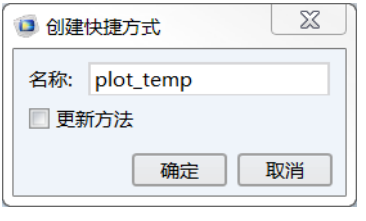


当需要更大灵活性时，您可以结合使用**选择列表**和**单位集**。例如，对于组合框，您可以使用**单位集**作为**选定源** （字符串）并选择不是**单位集**的一个**选择列表**。

快捷方式

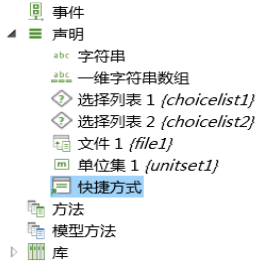
表单对象及其他用户界面注释通过使用特定的语法在方法中引用。例如，使用默认命名方案 `form3/button5` 指向 `form3` 中名为 `button5` 的按钮，`form2/graphics3` 指向 `form2` 中名为 `graphics3` 的图形对象。您还可以更改表单和表单对象的默认名称。例如，如果 `form1` 是您的主表单，那么您可以将其名称更改为 `main`。要按名称简化引用表单对象以及菜单、功能区和工具栏项，您可以使用定制 名称创建 快捷方式。在对象或项的**设置**窗口中，单击**名称**框右侧的按钮并键入您选择的名称。



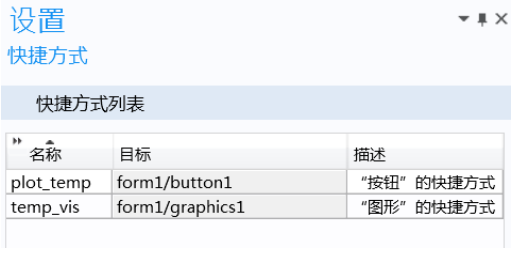


要创建或编辑快捷方式，还可以使用键盘快捷键 Ctrl+K。

您创建的所有快捷方式在 App 树中**声明**下的**快捷方式**节点中都可用。



在下面的**快捷方式设置**窗口中，已分别为按钮和图形对象创建了两个快捷方式，即 `plot_temp` 和 `temp_vis`。



快捷方式可以在其他表单对象或方法编辑器中的代码中引用。下面的示例显示用作温度绘图的输入变元的快捷方式 `temp_vis`。



当对象被重命名、移动、复制及生成副本时，快捷方式会自动更新。它们在 App 方法中可作为只读 Java® 变量使用，就像字符串、整数、双精度数和布尔变量声明一样。

建议使用快捷方式，这样，在 App 用户界面结构更改时，无需调整方法编辑器代码。

“模型开发器”中也提供快捷方式，可与 “App 开发器”配合使用。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选择的名称。



快捷方式的定制名称可用作方法中的全局变量，并将 （以此为例）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鼠标点击从图形对象中选择特定坐标处的数据。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。

可以从 “编辑器工具”窗口中使用图形数据声明的不同属性，如下图所示。



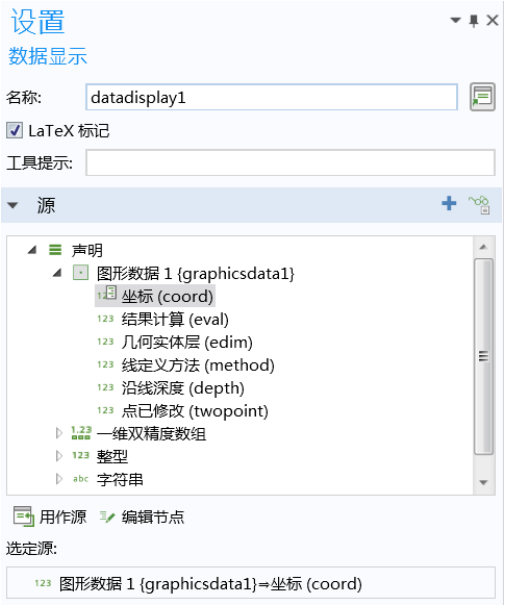
要使用**图形数据**声明节点进行 数据采集，请在图形对象的**设置**窗口中选中**数据采集**复选框，并选择**图形数据** 节点作为**数据采集的目标**，如下图所示。



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**值对应于鼠标指针确定的位置处的场值。**坐标**值对应于该位置处的坐标。请注意，在 “模型开发器” 中，其对应于**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。



还可以使用**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**以及**点已修改**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，这些设置提供与**域点探针**相同的点选择方法；当**几何实体层**设为**边界**时，则提供与**边界点探针**相同的点选择方法。**线定义方法**、**沿线深度**以及**点已修改**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，实现模型开发器的模型树节点中没有包含的标准运行命令的操作。例如，方法可以运行循环、执行输入和输出，以及发送信息和警告用户。

COMSOL 方法使用 Java® 编程语言来编写，这意味着您可以使用所有的 Java® 语法和库。除了 Java® 库， App 开发器有自己的内建库，用于开发 App 和修改模型对象。模型对象是保存内嵌在 App 中的底层 COMSOL Multiphysics 模型的状态的数据结构。请查阅第 259 页 “附录 E — 内建方法库”和 *Application Programming Guide* 了解更多这些内置方法的信息。

“App 开发器”中 App 树的内容通过 App 对象来访问，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写代码，方法编辑器可直接访问和更改运行的 App 的用户界面外观，例如，按钮文本、图标、颜色和字体。

方法分为两种类型：App 方法和模型方法。 App 方法修改正在运行的 App 的模型对象，而不是当前会话中由 “模型开发器”表示的模型对象。

模型方法与 App 方法类似，重要区别在于模型方法直接修改当前会话中由 “模型开发器”表示的模型对象。

App 方法既可以是全局，也可以是局部性质。全局方法显示在 App 树中，可以通过所有 App 方法和表单对象进行访问。可以从 App 方法中调用模型方法，但不能反过来调用。局部方法与某个表单对象或事件关联，可以从对应的设置窗口打开。请查阅第 121 页 “局部方法”了解更多关于局部方法的信息。

除了少数例外情况，本节的内容适用于 App 方法和模型方法。有关模型方法的更多信息，请参阅第 172 页的 “模型方法”。

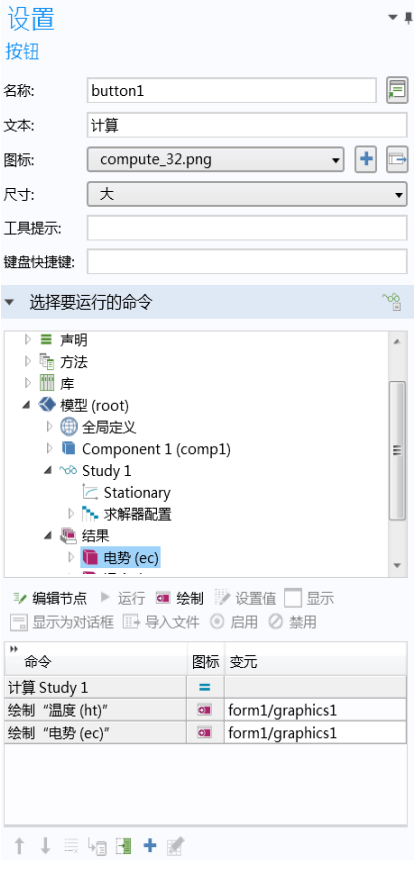
有很多工具和资源帮助您创建方法的代码，在下面的章节中将进一步为您说明，这些工具允许您通过复制 - 粘贴代码或自动生成代码块来提高代码编写效率。

转换命令序列为方法

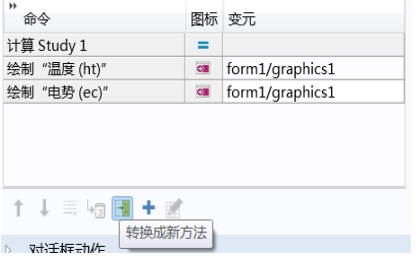
在表单编辑器中，单击**设置**窗口中已有的命令序列下的**转换成新方法**按钮，这个命令序列将被一个等效的 App 方法所取代。

假如您已经创建了一个计算按钮，希望当计算完成时发出警告音。现在，我们来看看如何使用方法编辑器来实现。

在本节的后半段，您还将了解如何不使用方法编辑器来实现这个需求。下图显示了计算按钮的**设置**窗口。

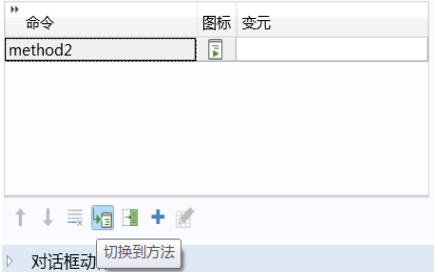


单击命令序列下的**转换成新方法**按钮。

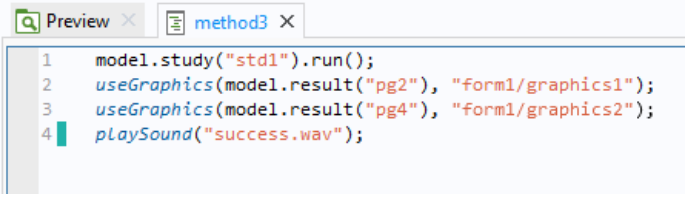


本例中的命令序列被方法所取代，即 method3。

单击**切换到方法**按钮，打开方法编辑器，并激活到 method3 选项卡。



在方法编辑器中，添加一个内建方法 playSound 来播放声音库中的声音文件 success.wav，其语法如下图所示。



新增的行以左侧绿色条标识出来。

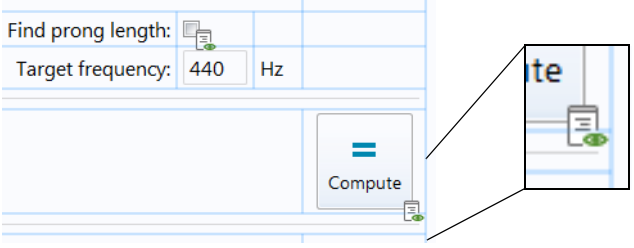
注意，在上例中，您并不一定需要使用方法编辑器。在命令序列中，选择**库** > **声音**中的文件 success.wav，然后单击下面的**运行**命令按钮，如下图所示。



然而，有很多内建的方法并没有对应的命令序列节点，参见第 259 页 “附件 E – 内建方法库”，了解更多信息。

带关联方法的表单对象

带有关联方法的表单对象通常以一个特殊图标来标识，如下图所示。在这个案例中，名称为 **Find prong length** 的复选框和 **Compute** 按钮都有关联的方法。



在表单对象上执行 Ctrl+Alt+ 单击将在方法编辑器中打开对应的方法。如果表单对象还没有关联方法，则创建一个与表单对象关联的局部方法，并在方法编辑器中打开。

如果关联的方法存在编译错误，则将显示其他图标，如下图中所示。



语言单元窗口

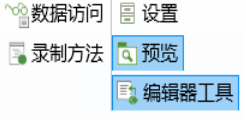
方法编辑器中的**语言单元**窗口显示很多语言结构的列表。双击或右键单击列表中的条目，可以将模板代码插入到当前选定的方法：



参见 第 128 页 “语言单元示例”。

方法编辑器中的编辑器工具

单击**方法**选项卡中的**主**组中对应的按钮，可以打开**编辑器工具**窗口。



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，最多有 8 种不同选项：

- **获取**
- **设置**
- **设置所有**
- **创建**
- **运行**
- **启用**
- **禁用**
- **编辑节点**

选择前 7 个选项将在当前选定的方法中添加对应的代码。**编辑节点**选项将您带到模型树节点的**设置**窗口。

下图显示了一个带有 6 个选项的节点示例。



当选定一个节点时，编辑器树下的工具栏显示可以生成代码的的选项。

当使用表单编辑器时，编辑器工具窗口也是一个重要的工具。参见第 45 页“表单编辑器中的编辑器工具”，了解更多信息。

键盘快捷键

考虑一个有一行按照下面这种方式指向模型对象的代码的方法：

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

如果您将鼠标放到 “surf1” 然后按下键盘上的 F11，右键单击并选择**切换到节点**，或者单击功能区的**切换到节点**，将会高亮显示**编辑器工具**窗口中对应的表面绘图节点。

单击**编辑节点**将打开它的**设置**窗口。参见第 257 页 “附录 D — 键盘快捷键”，了解更多键盘快捷键的信息。

方法编辑器中的数据访问

如果要访问模型树节点的单个属性，可以单击模型开发器功能区中**开发工具**选项卡 **App** 栏中的**数据访问**按钮：

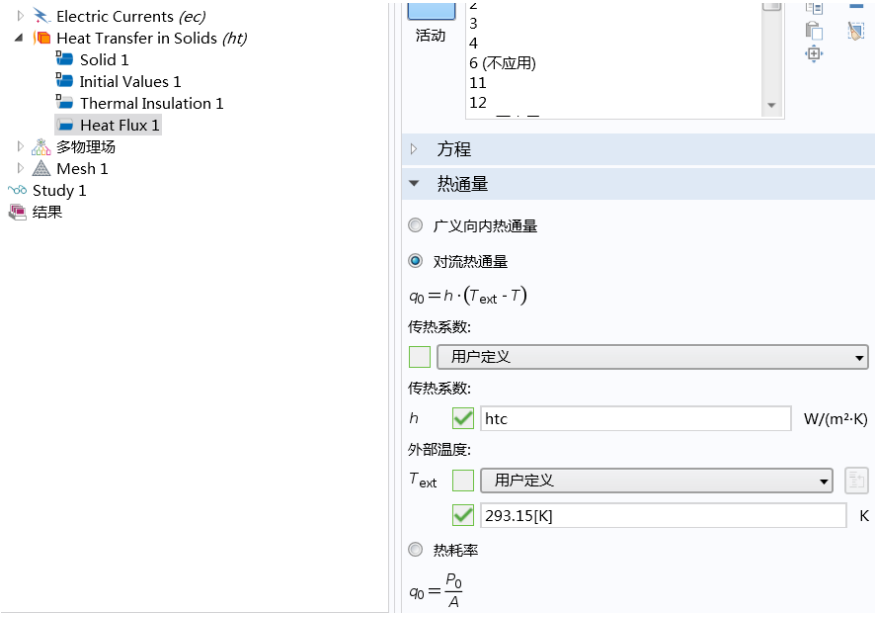


或者，对于确切的表单对象，您可以单击**设置**窗口的**源**栏题头中的**数据访问**按钮。参见第 56 页 “模型数据访问”。

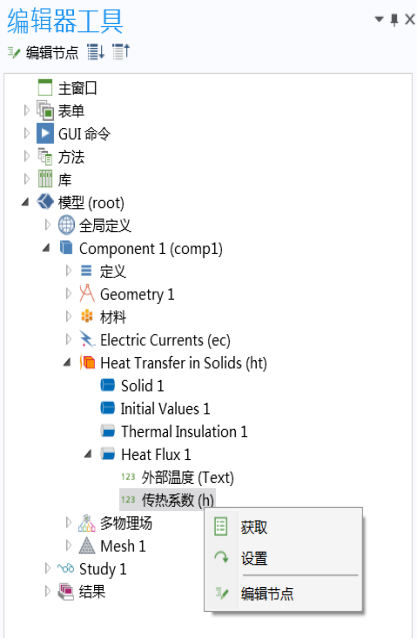
以这种方式启用**数据访问**的原因是模型通常包含数百乃至上千个可供访问的属性，这个列表太长以至于不实用。

当您单击模型树节点时，例如下图所示的**热通量**节点，复选框显示在单个属性后面。这个示例基于 COMSOL Multiphysics 简介中的 busbar 教学模型。

下图中，选定了**传热系数**和**外部温度**复选框：



如果您现在切换到**编辑器工具**窗口，将看到在**热通量**节点下显示出更多的节点。右键单击并使用**获取**或**设置**，可以在激活的方法窗口中生成相应的代码，如下图所示：



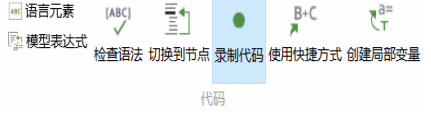
上面的示例中，**传热系数**和**外部温度**属性的**获取**和**设置**将生成以下代码：

```
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]");
```

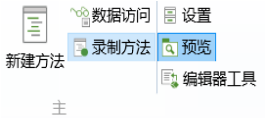
录制代码

单击方法编辑器的**代码**栏中的**录制代码**按钮，可以记录您在模型树中执行的操作对应的命令序列，如下图所示。



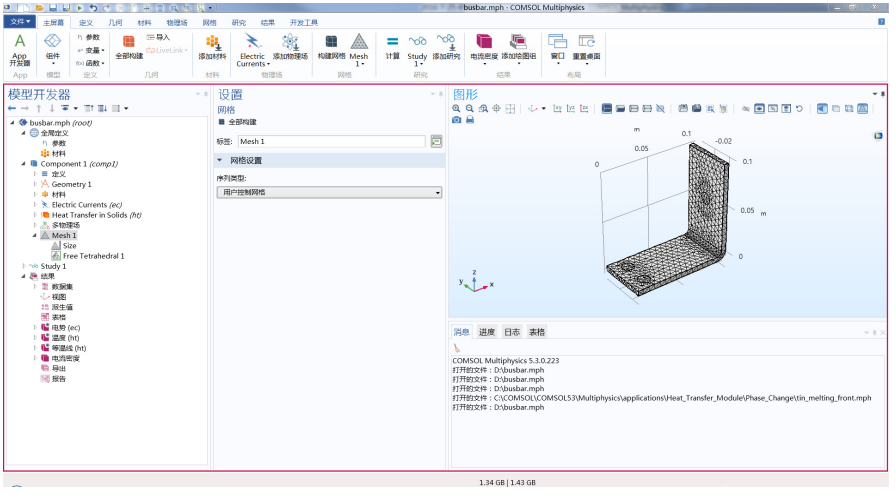
也可以录制 App 树中的某些操作，包括用于在 App 运行时修改用户界面的方法，例如，更改文本标签的颜色。

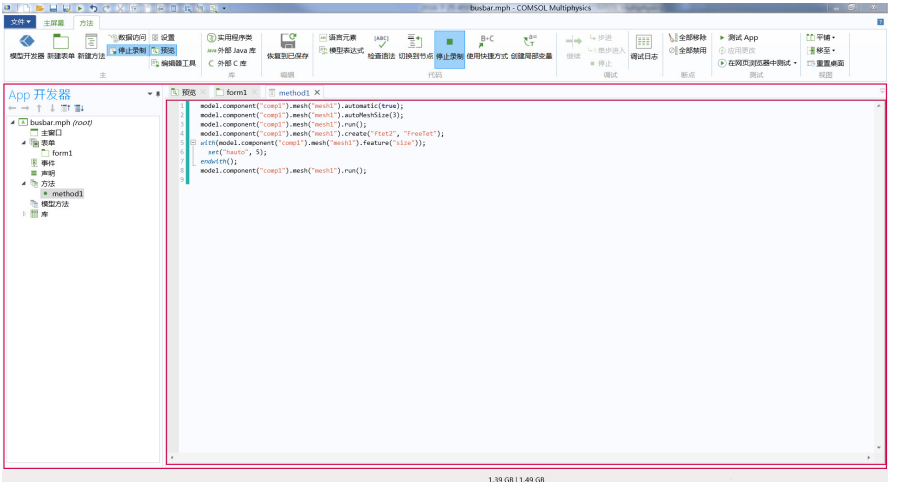
要录制新方法，可单击表单编辑器或方法编辑器功能区**主**栏中的**录制方法**按钮。



也可以单击 “模型开发器”功能区**开发工具**选项卡中的**录制代码**按钮。

在录制代码的过程中， COMSOL Desktop 窗口将有红色边框提示：





要停止录制代码，请单击模型开发器或 App 开发器功能区中的**停止记录**按钮 。





之前的**数据访问**章节解释了如何设置 busbar 教学模型的**传热系数**和**外部温度**属性的值，按照以下步骤可以使用**录制代码**生成相似的代码：

- 基于 busbar 模型 （MPH 文件）创建一个简单的 App。
- 在模型开发器窗口，单击**录制方法**，或打开方法编辑器，单击**录制代码**。
- 修改**传热系数**的值，例如， 5。
- 修改**外部温度**的值，例如，300[K]。
- 单击**停止记录**。
- 如果还没有打开，请打开记录下的代码的方法。

结果代码如下所示：

```
with(model.physics("ht").feature("hf1"));
  set("h", "5");
```

```
  set("Text", "300[K]");
endwith();
```

在这种情况下，自动记录的代码包含 with() 语句，使得代码更简洁。参见 第 130 页” With 语句”了解更多信息。

要生成对应于 App 对象更改的代码，可使用**录制代码**或**录制方法**，然后转至表单编辑器，例如，更改表单对象的外观。以下代码对应于文本标签颜色从默认的**继承**到**蓝色**的更改：

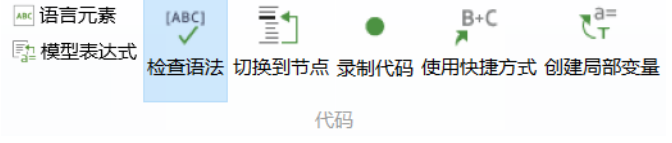
```
with(app.form("form1").formObject("textlabel1"));
  set("foreground", "blue");
endwith();
```

参见 *Application Programming Guide* 了解更多修改模型对象和 App 对象的信息。

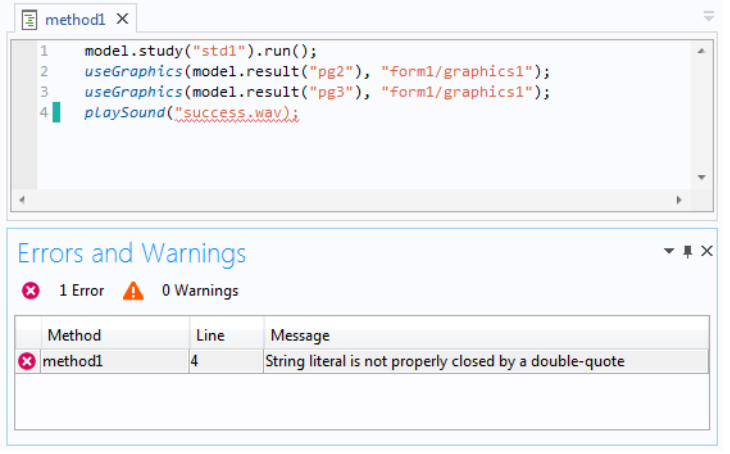
使用录制代码的工具快速学习如何与模型对象或 App 对象互动。自动生成的代码将向您显示出属性、参数，以及变量的名称，可以使用字符串和字符串 - 数字转换在模型属性中赋新参数值。通过记录代码时使用**数据访问**，您可以很方便地实现代码，例如，使用获取来提取参数值，在方法中进行处理，然后使用设置将它设置回模型对象。参见第 130 页 “方法编辑器中的模型数据访问”，了解更多**数据访问**的信息。

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**窗口中的信息，了解语法错误或没有用到的变量。



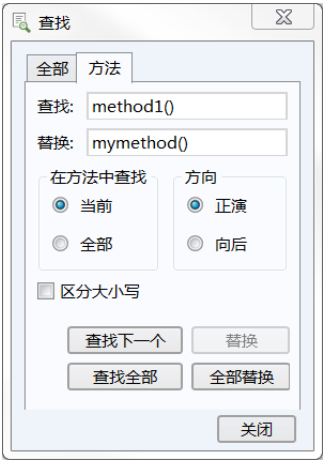
除了**错误和警告**窗口中的信息，还会在代码中以红色波浪下划线指示语法错误，如下图所示。



查找和替换

单击快速访问工具栏中的**查找**按钮，打开用于在方法中查找和替换字符串的对话框，如下图所示。





快速访问工具栏位于 COMSOL Desktop 用户界面的功能区的左上角。

全部标签用于在模型开发器和 App 开发器中同时查找字符串和变量。

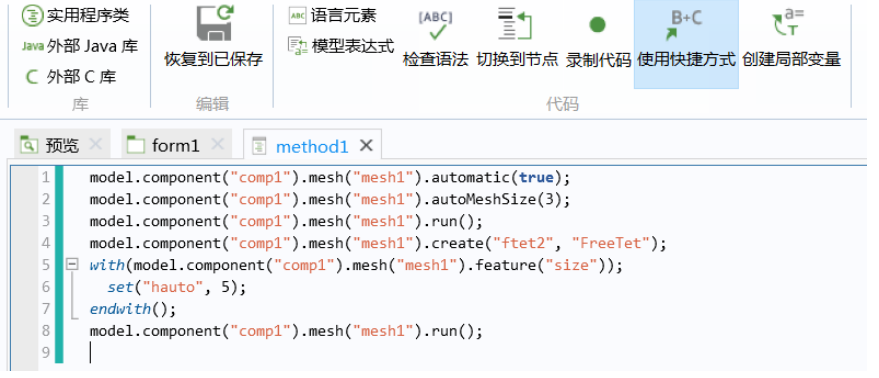
模型表达式窗口

方法编辑器中的**模型表达式**窗口显示一个预定义表达式的列表，用作输入和输出的变元。双击或右键单击列表中的条目，可以插入一个表达式：



使用快捷方式

如果您查看下面这个示例，将会注意到，最后两行代码以 `model.result("pg1")` 开头。

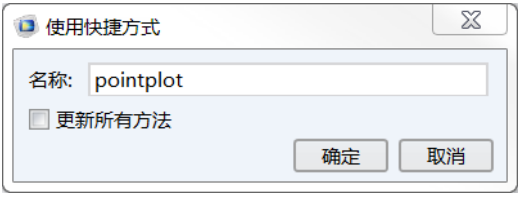


使用快捷方式按钮通过将这些实例替换为变量名称来简化代码。

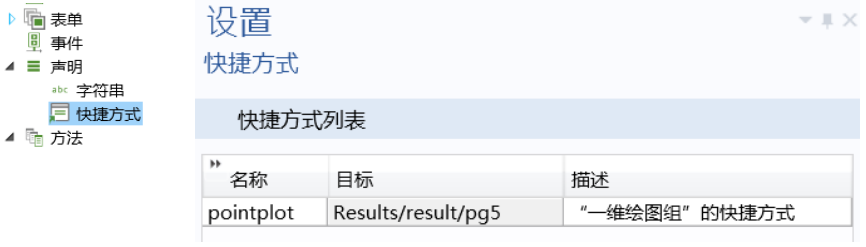
上面的示例中，光标已位于第一次出现 `model.result("pg1")` 的位置，单击**使用快捷方式**按钮将源代码转换成下图所示的样子。



以前缀 `model.result("pg1")` 开头的代码已替换为变量 `pointplot`，单击**使用快捷方式**按钮，将打开一个**使用快捷方式**对话框，您可以在**名称**框中输入一个合适的变量名称，此例中为 `pointplot`。



该变量作为快捷方式存储在**声明**节点中，如下图所示，一同显示的还有相应的**设置**窗口。



语法高亮、代码折叠和缩进

代码中的不同语言单元按照不同风格显示，参见下图：

```
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  }
```

这个示例中包含 5 种风格：

- 关键词，例如 if、 else、 for、 while、 double，以及 int 以粗体蓝色字体显示
- 内建方法显示为斜体蓝色字体
- 字符串显示为红色字体
- 注释显示为绿色字体
- 代码提示符显示为黑色字体

您可以在**首选项**对话框中定制语法高亮方案，参见下一节方法编辑器首选项。

您可以展开和折叠部分对应于代码段的代码，例如，for、while、if 和 else 语句的部分。这个特征可以被禁用，参见下一节 “方法编辑器首选项”的内容。

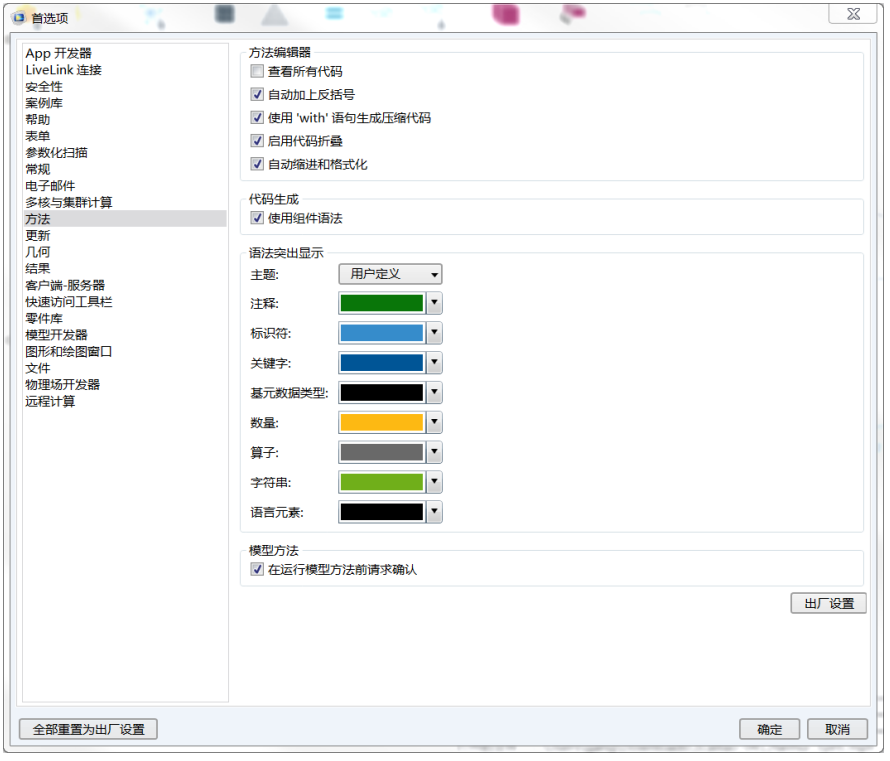
当编写代码时，按下键盘上的 Tab 键可以自动缩进代码行，在前面插入所需的空白。当键盘焦点离开方法编辑器时，也会自动进行缩进和空格的格式重排。您可以在**首选项**的**方法**栏，通过清除**自动缩进和格式化**复选框，禁用这种功能。

方法的名称

方法的名称是一个不包含空格的文本字符串，其中可以包含字母、数字和下划线。不允许使用保留名称 root 和 parent，以及 Java® 编程语言的关键词。

方法编辑器首选项

如果要访问方法的**首选项**，选择**文件** > **首选项**，然后选择**方法**。



默认情况下，方法编辑器只显示最相关的代码，如果需要查看方法中的所有代码，请选择**查看所有代码**复选框。

复选框**自动加上反括号**控制方法编辑器是否自动加上反括号，例如，花括号 {，方括号 [，以及括号 （。

复选框**使用’ with’语句生成压缩代码**控制自动生成代码时使用 with 语句。请查阅第 130 页 “With 语句” 了解更多信息。

如果选中复选框**启用代码折叠**，您可以展开和折叠与 for、 while、 if 和 else 语句相关联的对应代码段。

选中复选框**自动缩进和格式化**将确保代码一贯缩进和格式化。

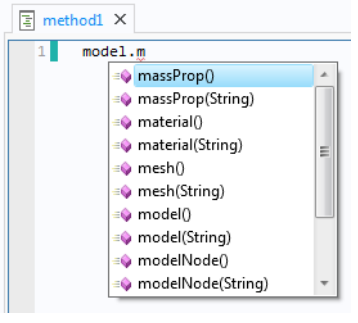
使用组件语法选项将生成方法语法，其中还包括模型组件范围。
在**语法突出显示**下，**主题**列表中包含两个预定义主题，**现代**（默认）和**经典**。可以选择**用户定义**来定义语法高亮模式，其中的颜色可以赋给不同的语言单元。

运行模型方法时，清除您不想确认的**在运行模型方法前请求确认**选项。

Ctrl+Space 和 Tab 实现代码完备

当在方法编辑器中键入代码时， App 开发器可以提供建议实现代码完备。在键入过程中，可能打开一个实现完备的列表。在一些情况下，当从列表中选定条目时，在单独的窗口中显示出详细信息。

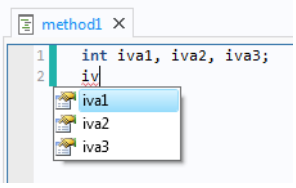
代码完备总是可以通过键盘快捷键 Ctrl+ 空格进行请求。当访问模型对象的一部分时，您将获得可能实现完备的列表，如下图所示：



通过使用箭头键从列表中选择条目，并按 Tab 或回车键确定选择。

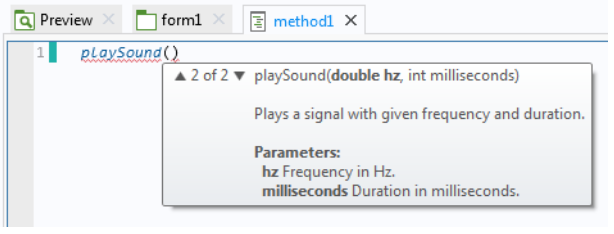
如果列表太长，您可以通过键入开头的几个字符进行过滤。

例如，如果您键入变量或方法名称的头几个字符，然后按下 Ctrl+ 空格，将显示出可能实现完备的条目：



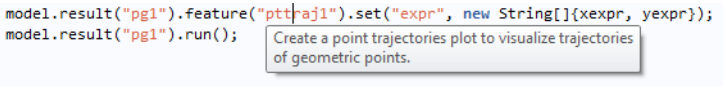
上例中，只有匹配 iv 的变量会显示出来。这个示例显示局部方法的变量也会显示在这个完备建议的列表中。

您还可以使用 Ctrl+Space 了解与模型对象不直接相关的内置方法的语法相关信息。键入命令名称并使用 Ctrl+Space 打开包含各种可供使用的用法说明的相关信息的窗口。

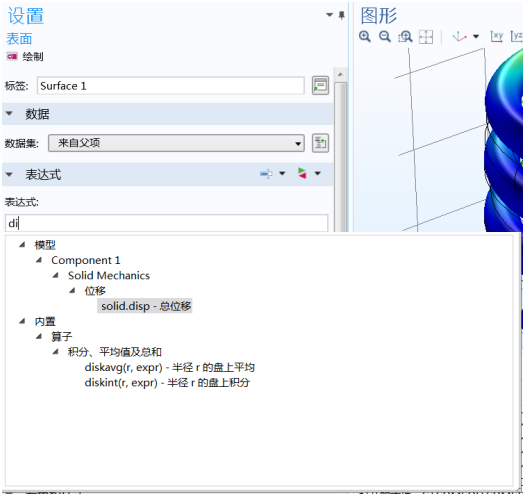


有关可用内置方法的列表，您可以使用 147 页所述的语言单元窗口或参阅 284 页的 “附录 E - 内置方法库”。

将鼠标悬停在方法调用、属性名称、声明或快捷方式的不同部分时，工具提示中显示相似的信息。



键盘快捷键 Ctrl+ 空格还可以用于模型开发器。例如，当在**结果**中的**表达式**键入时，使用 Ctrl+ 空格来查找匹配的变量，如下图所示。

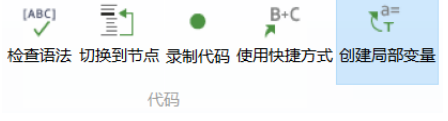


创建局部变量

您可以自动设置局部变量的类型。例如，您可以键入

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

然后单击功能区中 “方法”选项卡的 “代码”组中的**创建局部变量**。



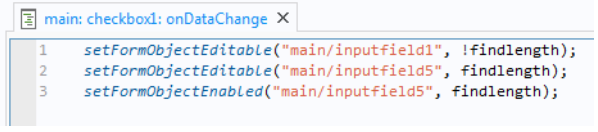
然后，代码变为

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

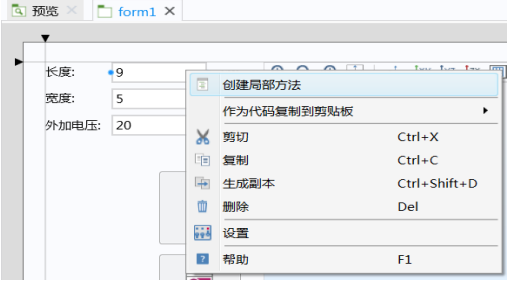
其中， `GeomList` 是 `model.geom()` 的数据类型。

局部方法

您可以向按钮、菜单项和事件添加局部方法，在 App 树的**方法**节点下不会列出局部方法。在局部方法的方法窗口，以标签形式显示它关联的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，然后从菜单中选择**创建局部方法**，如下图所示。

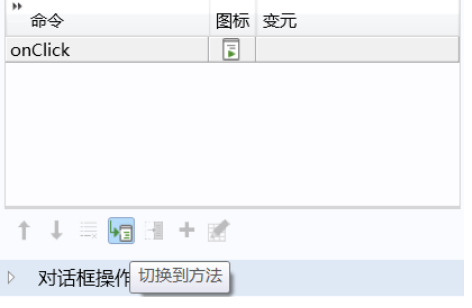


按钮、菜单项和全局事件的局部方法

对于按钮、功能区、菜单、工具栏项，以及全局事件，您可以通过单击命令序列下的**创建局部方法**按钮来添加局部方法，如下图所示。



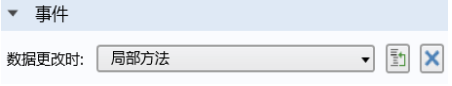
这个按钮的功能与**转换成新方法**按钮类似，参见第 13 页 “创建新方法”。唯一的区别在于它会创建一个在 App 树的全局方法中不可见的局部方法。它还会在创建后，将新方法打开在方法编辑器中。可以使用快捷键 Ctrl+Alt+ 单击来创建局部方法。单击按钮**切换到方法**将打开局部方法。下图显示调用关联到一个按钮的局部方法。



为了避免在局部方法中出现破坏代码的风险，当在命令序列中已显示有一个局部方法时，您无法使用**转换成新方法**。

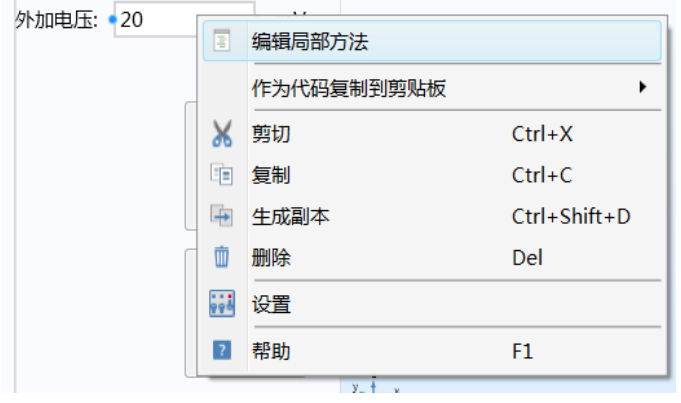
表单和表单对象事件的局部方法

如果要为表单或表单对象事件创建局部方法，请单击**设置**窗口的**事件**栏中的**创建局部方法**按钮。选定的**数据改变时执行**的方法从**无**改为**局部方法**，如下图所示，并打开方法编辑器。



如果要在方法编辑器中打开已有的局部方法，请单击**切换到源按钮**。单击**删除局部方法**按钮可以删除局部方法。

除了使用 Ctrl+Alt+ 单击，您还可以右键单击表单对象，然后从上下文菜单中选择**编辑局部方法**或**编辑方法**。



参见第 87 页 “事件”，了解更多信息。

模型方法

模型 方法与 App 方法类似，最重要的区别在于模型方法直接修改当前会话中由“模型开发器”表示的模型对象。像 App 方法一样，也使用 “方法”编辑器进行编辑。模型方法可用于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，您可以录制先计算 “研究 1”的过程的代码；然后基于“研究 1”的解计算 “研究 2”；以此类推。

与 App 方法相比，模型方法具有额外限制，如下所示：

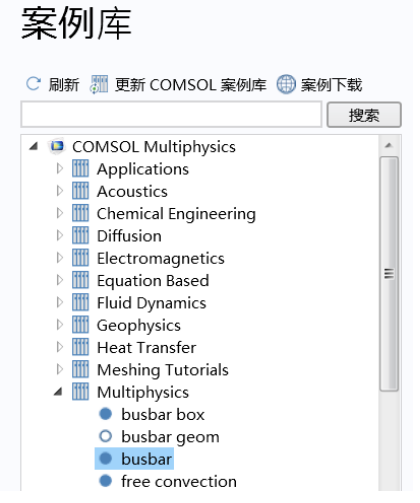
- 模型方法不能用于修改 App 对象。
- 模型方法不能调用 App 方法或 Java® 实用程序类。
- 与**语言元素**窗口中的分支用户界面对应的许多内置 App 方法不能从模型方法中调用。

从其他模型方法调用的模型方法可能具有输入和输出变元。直接从 “模型开发器”调用的模型方法不能具有输入或输出变元，“模型开发器”功能区中仅显示这些模型方法。

计算后自动生成报告

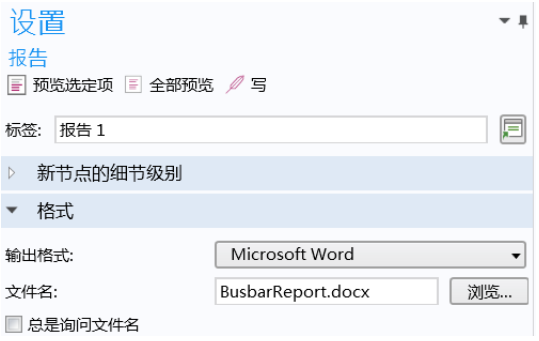
作为使用模型方法的示例，考虑先计算解，然后生成报告的过程。这可以实现自动操作，方法是先录制 “模型开发器”中的相应操作，然后运行模型方法。

我们先看 《*COMSOL Multiphysics 简介*》中描述的母线板示例。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**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，在系统上具有写权限的位置选择一个文件名，例如：`C:\COMSOL\BusbarReport.docx`

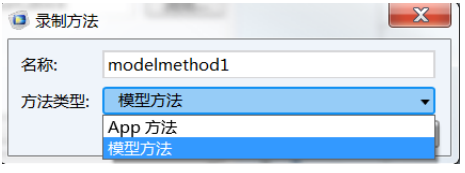
- 单击**写**生成报告并将报告保存到文件。



- 关闭 Microsoft® Word 文档。
- 单击 （**模型开发器**）功能区中的 **开发工具**选项卡，然后单击**录制方法**按钮。

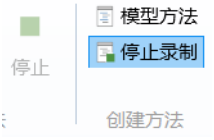


- 在**录制方法**对话框中，选择**模型方法**。

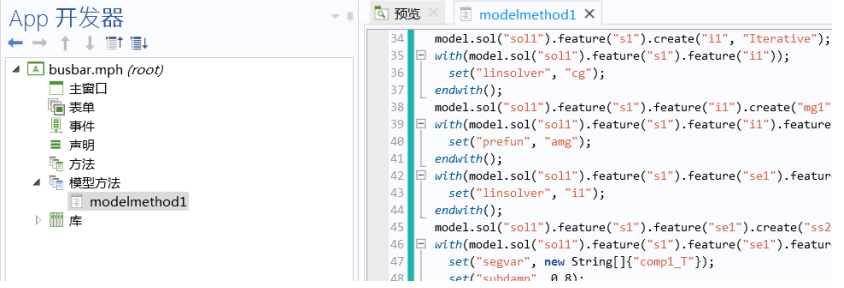


- 在模型树中，右键单击**研究 1** 节点，并选择**计算** （或使用**计算**的功能区选项）。
- 在模型树中，右键单击**报告 1** 节点，然后选择**写**。

- 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*。

- 您可以在此处检查和边界生成的代码，并将方法的名称更改为**计算和报告**（以此为例）。

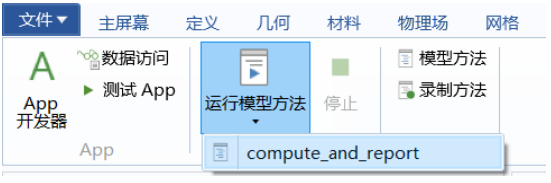


- 通过单击功能区中的**模型开发器**按钮返回到模型开发器。

- 在**全局定义** > **参数**中，将**长度**更改为 15[cm]。



- 在功能区中，单击**开发工具**选项卡，然后从**运行模型方法**菜单中选择**计算和报告**。

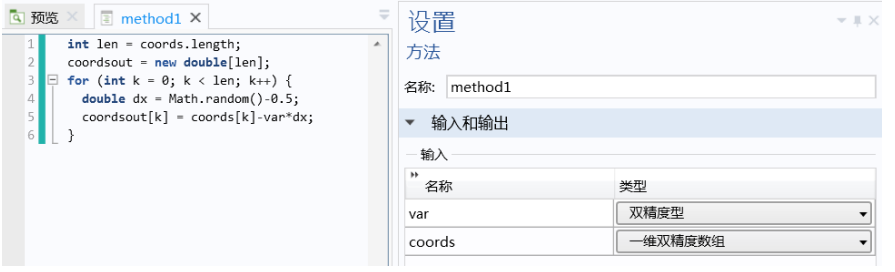


请注意，模型中可以有多种模型方法。

带输入输出变元的方法

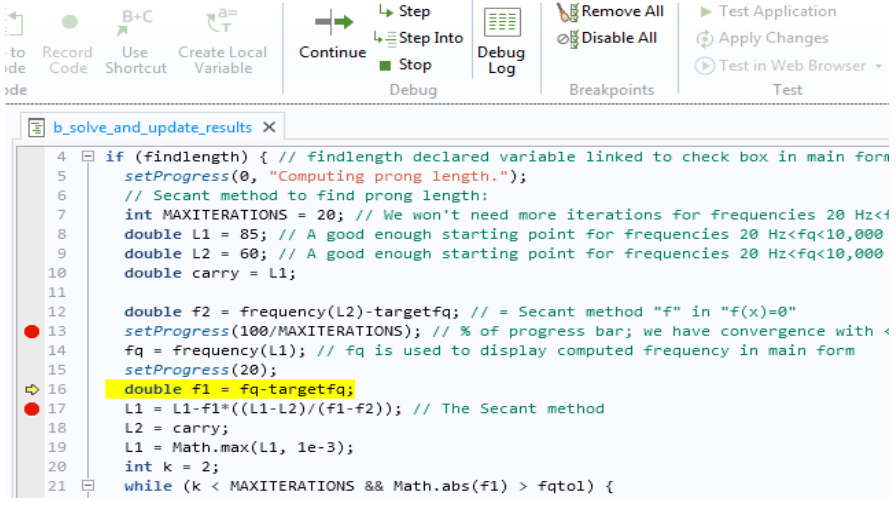
方法可以带一些输入变元和一个输出变元。您可以在激活的方法窗口的**设置**窗口中定义输入和输出变元。如果**设置**窗口不可见，可以单击功能区的方法选项卡的**设置**。下图显示的方法有两个输入变元， var 和 coords，以及一个输出，

coordsout。它用来向数组 coords 添加随机值。随机值的程序由输入变量 var 控制。新值储存在数组 coordsout 中。



当您从方法调用另一个方法时， Ctrl+Alt+ 双击可以打开该方法的窗口。一个方法可以调用本身形成递归。

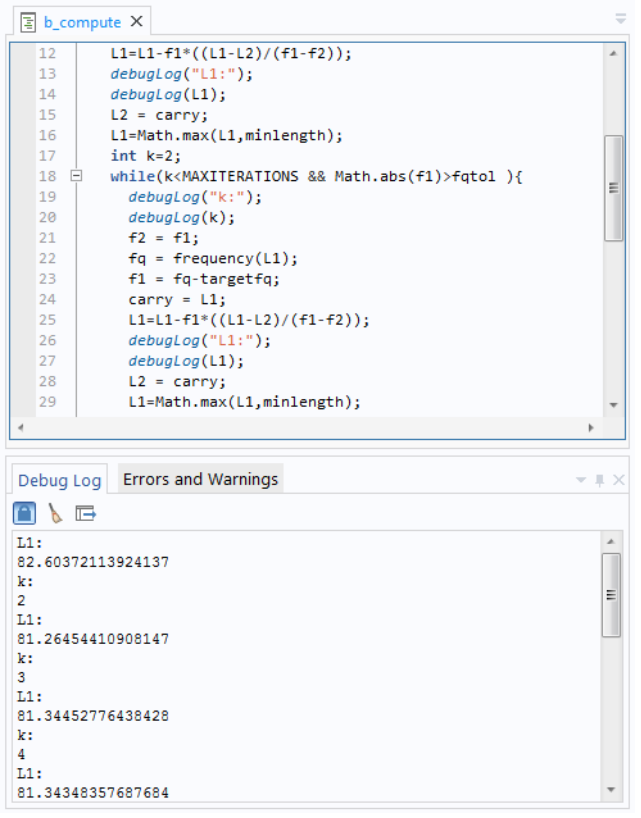
为了调试的目的，在代码行号的左侧灰色列中单击，可以设置断点，如下图所示。



在功能区中，**调试**组包含可供调试方法的工具。当您运行 App 时，方法将在断点处中断。单击**步骤**按钮进入方法的下一行。上图显示方法现在停在黄色高亮的行。

单击**继续**按钮可以运行方法，直到遇到下一个断点。单击**停止**按钮停止正在运行的方法。如果可能，单击**单步进入**按钮可以步进到下一个方法中。使用**全部删除**按钮可以删除所有断点。除了删除，您还可以通过单击全部禁用来禁用所

有的断点。单击**调试日志**可以在一个单独的**调试日志**窗口显示调试信息，如下图所示。



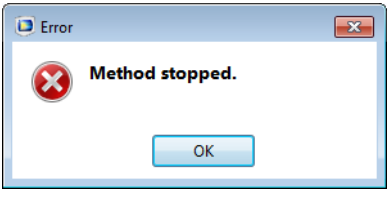
使用 debugLog 命令可以在**调试日志**窗口显示变量的值。下面的代码描述使用命令 debugLog 显示字符串和一维双精度数组分量的值。

```
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");
  }
}
```

参见第 265 页 “调试方法”，了解更多用于调试的内建方法的信息。

停止方法

您可以在测试 App 的过程中通过使用键盘快捷键 Ctrl+Pause 停止运行方法。这时会显示一个对话框，如下所示：



模型对象

模型对象提供大量的方法，包括设置和运行操作序列的方法。方法编辑器的**转换成方法**、**记录代码**、**模型代码**，以及**语言单元**等产生这样的模型对象方法的语句。请查阅 COMSOL Multiphysics 简介的 “附录 C — 语言单元和保留名称”、*Application Programming Guide* 以及 Programming Reference Manual，了解更多与模型对象及其方法相关的信息和示例代码。

语言单元示例

Java® 编程语言用来编写 COMSOL 方法，这意味着通常可以使用 Java® 指令和语法。请查阅 *Application Programming Guide* 和 *Programming Reference Manual* 了解更多信息。

模型对象中的一元和二元算子

下表描述了当访问一个模型对象时可以使用的一元和二元算子，例如，当定义材料属性和边界条件，以及用于后处理和图形化的结果表达式等。

优先级	标记	描述
1	() {} .	编组、列举、范围
2	^	幂
3	! - +	一元：逻辑非、负、正
4	[ ]	单位
5	* /	二元：乘、除
6	+ -	二元：加、减
7	< <= > >=	比较：小于，小于或等于，大于、大于或等于
8	== !=	比较：等于、不等于
9	&&	逻辑与
10	||	逻辑或
11	,	列举中的单元分隔符

方法中的一元和二元算子 （JAVA® 语法）

下表描述在方法的 Java® 代码中最常用到的一元和二元算子。

优先级	标记	描述
1	++ --	一元：追加和追减
2	++ -- + - !	一元：加、减、正号、负号、逻辑非
3	* / %	二元：乘、除，取模
4	+ -	二元：加、减
5	!	逻辑非
6	< <= > >=	比较：小于、小于或等于、大于、大于或等于
7	== !=	比较：等于、不等于
8	&&	二元：逻辑与
9	||	二元：逻辑或
10	?:	条件三元

优先级	标记	描述
11	= += -= *= /= %= >>= <<= &= ^= |=	分配
12	,	列表中的单元分隔线

访问声明节点中的变量

声明节点中的定义的变量可以作为方法中的全局变量，不需要再进行声明。

内建基本数学函数

方法中使用的基本数学函数依赖于 Java® 数学库，下面是一些示例：

```
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();
```

上面的代码相当于：

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

访问全局参数

您通常会使用**编辑器工具**窗口来生成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 的值设置为 10 cm：

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

获取全局参数 L，然后将它存储在双精度数变量 Length：

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

此示例中关于基本**单位制**的计算是在模型树根节点中定义的。

使用下式返回参数 L （如果存在）的单位：

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

如果要将双精度数的值写入全局参数，您需要将它转换为字符串。原因是全局参数是模型表达式，可能包含单位。将变量 Length 的值乘上 2，然后将它写入参数 L，并且包含单位 cm：

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

使用下式返回与单位与基本单位制不同的参数的值：

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

如果参数的值比较复杂，实部和虚部可以作为长度 2 的双精度矢量返回：

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

比较字符串

在 Java® 中比较字符串可以使用 .equals() ，而不是使用 == 算子。这是因为 == 算子比较字符串是不是相同对象，而不是考虑它们的值。下面的代码演示字符串比较：

```
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
```

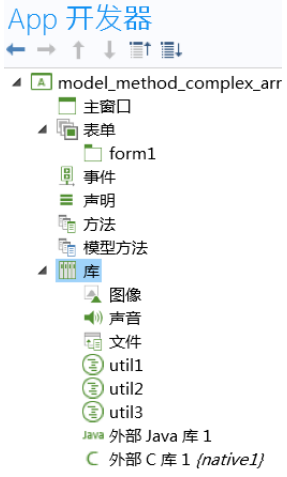
警告和信息

方法 alert、 confirm，以及 request 显示对话框，其中可以包含文本字符串和可选的用户输入。下面的示例使用 confirm 询问用户在 App 中使用直接或者迭代式求解器。然后基于回答，在信息对话框中使用 alert 函数显示根据选定求解器类型估算的内存需求：

```
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.");
```

库

在 App 树中，**库**节点包含图片、声音，以及内嵌到 MPH 文件中的文件，这样您不需要另外随 App 一起分发它们。此外，**库**节点可能包含 Java® 实用类节点，以及外部 Java® 和 C 的库节点。有关使用使用实用程序类和外部库的更多信息，请参阅 *Application Builder Reference Manual*。



内嵌的文件可以被表单对象或方法通过使用语法 embedded:///file1、embedded:///file2 等来引用。例如，使用语法 embedded:///compute.png 来引用图片文件 compute.png。

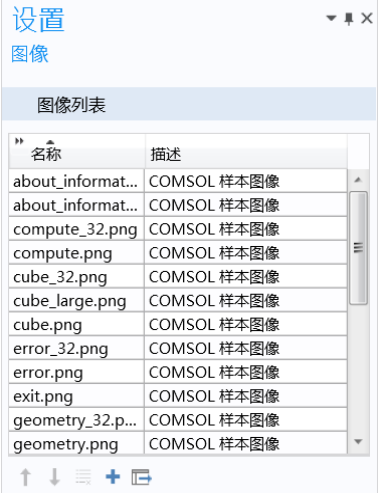
注意，您不需要在文件名中包含扩展名，可以使用任意的名称。如果需要减少 MPH 文件的大小，可以删除没有用到的图片、声音，或其他文件。

如果需要管理 App 的用户实时加载的文件，您有多种选择，包括使用**文件**声明和**文件导入**表单对象。而不是使用**库**节点。参见第 122 页 “文件” 和 第 201 页 “文件导入”，以及第 236 页 “附录 C – 文件处理和文件方案语法”，了解更多加载文件的信息。

图像

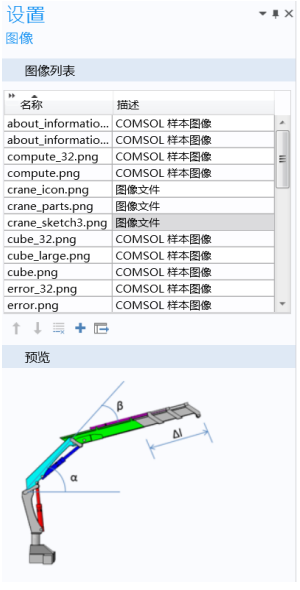
图像库包含 PNG 格式的很多预加载样本文件，如果您希望内嵌其他图像文件，单击**图像列表**下的**添加文件到库**按钮。在 COMSOL 安装文件夹中有很多可供

选择的图像，它们位于 data/images。图像用作图标，可以在图像表单对象或方法中引用。用作图标的图像有两种尺寸：16*16 像素 （小）和 32*32 像素（大）。



支持的文件格式包括：JPG、 GIF、 BMP，以及 PNG。

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



要导出选定的图像，可单击**预览**按钮右边的**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。如果您希望内嵌其他声音文件，可以单击**声音列表**下的**添加文件到库**按钮。在 COMSOL 安装文件夹中有很多可供选择的声音，它们位于 data/sounds 文件夹内。



如果要播放声音，可以单击声音文件名，然后单击**声音列表**下的**预览**按钮。

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

如果要在 App 中播放声音，可以在按钮、功能区、菜单，或者工具栏项目的**设置**窗口添加命令。在**选择运行的命令**栏，选择声音文件，然后单击树下的运行按钮。这个操作将在命令序列中添加一个**播放**命令，如下图所示。



在方法中，您可以使用内置的 playSound 方法来播放声音，例如：

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

文件

默认情况下，**文件**库是空的，可以单击**添加文件到库**来内嵌您希望添加到 App 的任意类型的文件。



单击**添加文件到库**按钮右边的**导出选定的文件**按钮，可以导出选定的文件。



内嵌的文件可以在方法中使用语法 embedded:///data1.txt、 embedded:///data2.txt 等进行引用。参见第 122 页”文件声明”、第 236 页 “附录 C — 文件方案语法” 和第 274 页 “文件方法”，了解更多信息。

附录 A— 表单对象

这个附录提供关于表单和表单对象的信息，扩展了第 33 页 “表单编辑器”。下面的列表中带有 * 标记的项目已经在该节详细描述，这里主要讨论其他剩余项目。

所有表单对象列表

- 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
  - 进度条
  - 日志
  - 信息日志
  - 结果表

- 子表单
  - 表单
  - 表单集合
  - 卡片堆
- 组合
  - 文件导入
  - 信息卡片堆
  - 数组输入
  - 单选键
  - 选择输入
- 杂项
  - 文本
  - 列表框
  - 表格
  - 滑块
  - 工具栏
  - 定距片

切换按钮

切换按钮对象是具有两种状态的按钮：**选定**和**清除**，如下图所示。



此栏中的信息也适用于**菜单切换项**和**功能区切换项**。

使用切换按钮来启用和禁用热源

切换按钮的两个状态保存在一个链接的布尔变量。下图显示了一个根据按钮的状态启用和禁用热源的设置窗口。在**源**栏中选定了布尔变量 heat_source。

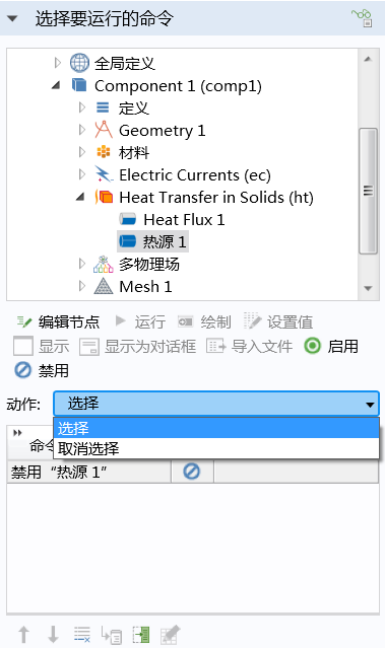


启用对应于布尔变量 heat_source 等于 true，它反过来对应于选定了切换按钮。禁用对应于布尔变量 heat_source 等于 false，也就是对应于清除了切换按钮。

在**源**栏下是**选择运行的命令**栏，包含**操作**的选择，分别代表对应于**选定**和**清除**的两种不同的命令。下图显示了代表**禁用热源**命令的**清除**的设置窗口。



接下来的图显示了代表启用热源命令的选定的设置窗口。



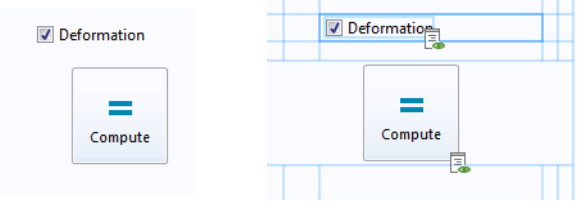
切换按钮类似于链接到布尔变量的复选框，对于切换按钮，通过命令序列来定义操作。而对于复选框，则通过事件来定义操作。参见下节中的描述。

复选框

复选框有两个值：on 表示选定， off 表示清除。复选框的状态通常储存在一个**声明**节点中的布尔变量。

使用复选框控制图形化

下图来自于一个通过是否选择复选框来控制禁用或启用形变绘图的 App。



左侧的截屏显示运行中的 App，右侧则显示栅格布局模式下对应的表单对象。

下面的示例表示复选框的状态储存在布尔变量 def ormation，其中显示设置窗口。



下图显示复选框的**设置**窗口。



您可以通过从**源项**栏的树中选择已声明的布尔变量并单击**作为源使用**将复选框与之关联。

默认情况下，复选框的文本标签将根据与之关联的布尔变量的**描述**中获取其名称。

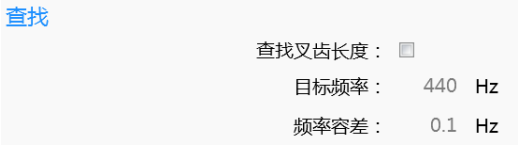
变量 deformation 的**初始值**被**选定的值** （on）或**清除的值** （off）所覆盖，不需要进行编辑。当在方法中使用时，值 on 和 off 分别是 true 和 false 的别名。这些值可以作为布尔变量在 if 语句中使用。

下面的代码语句来自于当布尔变量 deformation 的值发生变化时对应于**数据改变时运行**事件的局部方法。

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

使用复选框来启用或禁用表单对象

下图显示了一个依赖于是否选定复选框启用或禁用输入框的 App 的部分截图。



下图显示复选框与用来储存复选框状态的布尔变量 findlength 进行关联的**设置**窗口。使用复选框来启用或禁用表单对象。



下面的代码语句来自于当布尔变量 findlength 的值发生变化时产生**数据改变时运行**事件的局部方法。

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

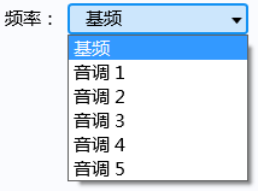
组合框

组合框相当于下拉列表框和可编辑的文本框或者不带可编辑内容的下拉列表框。

使用组合框改变结果中的参数

为了描述组合框的用法，让我们来考虑一个 App，其中用户从一个结构振动分析得到的 6 个不同模态中选择一个用于图形化显示。本例使用固体力学物理场接口和特征频率研究，可以用于所有此类分析。

6 个模态对应于 6 个不同的特征频率，用户可以从组合框中选择：

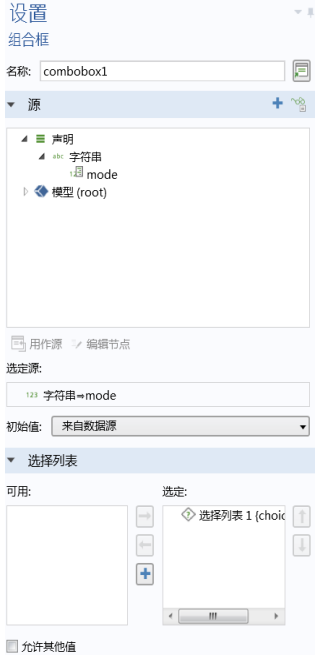


本例中，组合框用来控制 字符串变量 mode 的值，下图显示了它的**设置**窗口。



选择源

下图显示了这个组合框的**设置**窗口。



在**源项**栏，您选择一个由组合框控制值的标量变量，然后单击**作为源使用**。在组合框的**设置**窗口中的**初始值**列表，选择定义组合框的默认值的方法，这些选项有**第一个允许值**（默认）和**定制值**。对于**定制值**选项，需要在关联的编辑框中输入一个默认值。您输入的默认值必须存在于允许使用的值列表中。

选择列表

可变的模态 1-6 对应于基本的刚体模态，并不是这个 App 关心的事情，因此第一个受到关注的模态是 7。一个选择列表允许您在模型中隐藏实际的模态值，只在显示名称列中显示出字符串；前面的几个非刚体模态命名为基频、泛音 1、泛音 2，等等。

在**选择列表**栏，您可以添加**选择列表**，为组合框贡献允许值。下图显示了与本例关联的**选择列表**的声明。



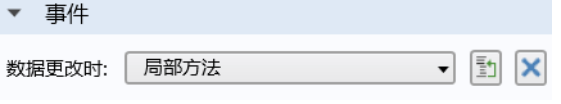
字符串变量 mode 可以取以下 6 个值：7、 8、 9、 10、 11 或 12。在**显示名称**列中的字符串将显示在组合框中。

在组合框的**设置**窗口，您可以选择**允许其他值**复选框，使得组合框中可以输入任意值。这样的组合框可以接受任意值，并没有被约束在由选择列表定义的值范围内。本例中，只允许使用 6 个预定义的值。

参见第 120 页 “选择列表声明”，了解更多关于选择列表的信息。

事件

在**事件**栏，指定当组合框的值发生变化时运行什么方法，据此由用户修改用作源的字符串。在这种情况下，当变量 mode 的值发生变化时，将运行一个局部方法，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 的值与绘图组 pg1 中的特征频率设置链接起来。在这种情况下，字符串取值 "7"、 "8"、 "9"、 "10"、 "11" 或 "12"。

上面的代码可以通过使用方法编辑器的记录代码功能自动生成：

- 切换到模型开发器，并且单击**记录新方法**。
- 默认情况下，当使用特征频率研究进行结构力学分析时，会创建一个**模态形状**绘图组。在这个绘图组中，可以修改**特征频率**从模态 7 改为模态 8。下图中，这对应于在**模态形状**绘图组的**设置**窗口中，从 440Hz 改为 632.89Hz。



- 单击**停止记录**。

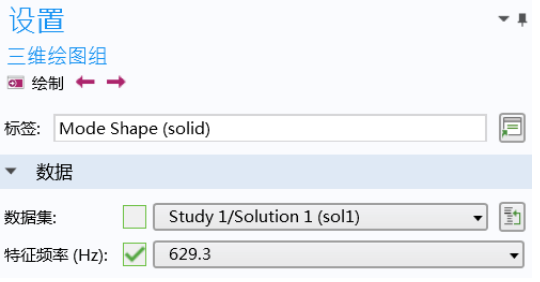
得到的代码如下所示：

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

现在将字符串” 8”替换成 mode，完成上面列出的代码。这段代码将储存在一个方法中。将这段代码从 method1 复制到创建组合框相关联的局部方法的代码中，然后删除 method1。

使用数据访问

更快，但是不那么通用的使用组合框的方法，是使用**数据访问**与**编辑器工具**的组合。例如在本节中，您可以启用**数据访问**，然后在模态形状绘图组的设置窗口，选择特征频率，如下图所示。



在**编辑器工具**窗口，特征频率参数显示为循环水平。为了创建一个组合框，右键单击循环水平，然后选择**输入**。



通用的名称**循环水平**用于解参数，如果解有两个或更多参数，则会有两个或更多循环水平可供选择。

下图显示了对应的组合框的设置窗口。

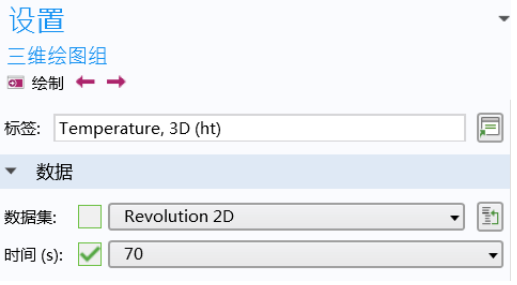


当使用编辑器工具插入一个组合框时，将自动生成名选择列表**循环水平**。注意，以这种方式生成的选择列表不会显示在**声明**节点下，并且用户不能修改它。如果需要更大的自由度，例如给出每个参数或特征频率值的名称，您需要如前面的章节中所述那样手动声明选择列表。

使用组合框改变时间

在**瞬态**研究步骤中指定的时间参数列表可以用于**结果**节点下的许多位置。在 App 中，可以通过与参数的上一部分中描述的方式类似的方式访问各个时间参数，即使用**数据访问**和**编辑器工具**。

在下图的**设置**窗口中，**数据访问**已用于访问温度绘图中的**时间**内参数列表。



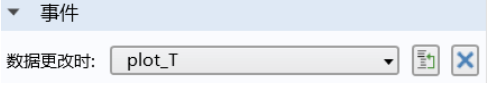
在**编辑器工具**中，**时间**列表现在具有一个句柄，如下图所示。



选择**输入**后，可以将其用作**源**来创建组合框，如下图所示。



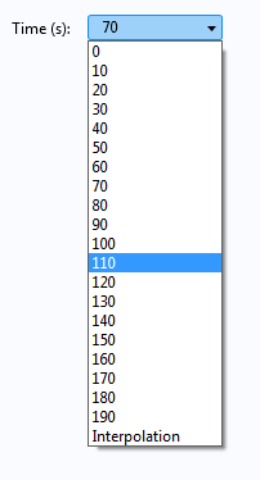
组合框有多种用途，例如，可更新与不同时间参数相对应的绘图。要使绘图在用户使用组合框选择新时间参数时自动更新，可在其设置窗口底部向组合框添加事件。在下图中，调用了 `plot_T` 方法来更新温度绘图。



以下代码行显示 了方法 `plot_T` 的内容：

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

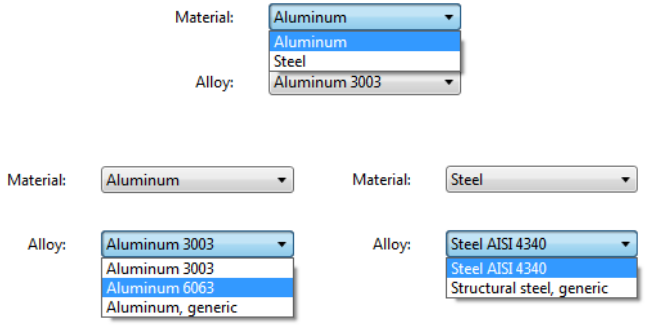
最终结果是 App 用户界面中的一个组合框，如下图所示，在用户选择新的**时间**列表值时，将自动更新温度绘图。



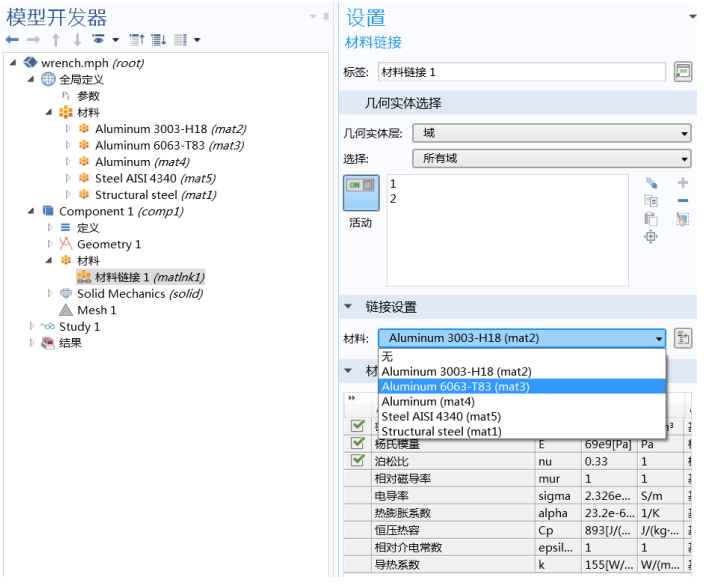
使用组合框改变材料

让我们来考虑一个 App，其中使用组合框用来选择材料。在这种情况下，激活条件 （参见第 121 页 “激活条件”）的使用将使得用户界面的设计更为灵活。

下图显示了一个 App 的截屏，其中用户可以使用名为 **Material** 的组合框从两种材料中进行选择，分别是 **Aluminum** 或 **Steel**。第二个称为 **Alloy** 的组合框显示 **Aluminum** 合金或 **Steel** 合金，根据 **Material** 列表中的选择而变。



材料选择可以通过在内嵌模型中使用全局材料和材料链接来实现，如下图所示。



每种材料通过字符串进行索引：mat1, mat2, ..., mat5。一个事件监听全局变量值 alloy 的变化，该值由组合框进行控制。当值发生变化时，将运行下面列出的这个方法。

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

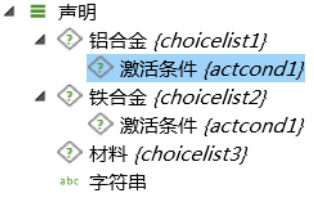
下图显示了两个字符串变量的声明，material 和 alloy，它们分别受到组合框 **Material** 和 **Alloy** 的控制。



App 利用了三个选择列表：**Aluminum Alloys**、 **Steel Alloys** 和 **Material**。

激活条件

一个激活条件用于选择列表 **Aluminum Alloys** 和 **Steel Alloys**，如下图所示。



组合框 **Material** 的**设置**窗口如下图所示。



注意，组合框 **Material** 使用字符串变量 material 作为源，选择列表 **Material** 用于定义字符串变量 material 的一组离散的允许值。材料选择列表的设置窗口如下图所示。



组合框 **Alloy** 的**设置**窗口如下图所示。



注意，组合框 **Alloy** 同时使用 **Aluminum Alloys** 和 **Alloys** 选择列表，其中 **Aluminum Alloys** 选择列表如下图所示。

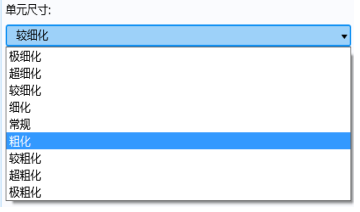


下图显示选择列表 **Aluminum Alloys** 的激活条件。



使用组合框修改单元尺寸

当创建一个组合框时，您可以使用**数据访问**功能重现模型开发器中已有的组合框。例如，考虑在一个 App 中使用组合框来修改网格的单元尺寸，如下图所示。



切换到**模型开发器**，然后选择**网格**节点 （我们假设这个模型只有一个网格）。在网格节点的设置窗口，选择**用户控制网格** （如果没有选定）。在网格节点的尺寸节点，选择预定义。单击功能区中的**数据访问**，这样就可以访问预定义网格尺寸的组合框，如下图所示。



选择列表左边的绿色复选框，让它可以作为 App 开发器中的组合框的源。然后当您回到 App 开发器时，将会发现网格尺寸的选择列表现在暴露出来作为新复合框的**设置**中的潜在的**源**。

要插入复合框对象，有两种方法：

- 从功能区的**插入对象**菜单中选择**复合框**。在复合框的**设置**窗口，源栏中选择节点**预定义尺寸 （hauto）**，然后单击**作为源使用**按钮。
- 在**编辑器工具**窗口的**网格 > 尺寸**节点下选择**预定义尺寸 （hauto）**，右键单击，然后选择**输入**，如下图所示。



下图显示了对应的组合框的**设置**窗口。



修改**初始值**为**来自源数据**，确保模型的单元尺寸设置在这种情况下使用**标准**为默认的单元尺寸。现在从模型开发器选定的选择列表**预定义尺寸 （hauto）**就会被选定为 App 开发器的复选框的选择列表。这个选择列表并不同于 App 树的声明节点下的选择列表，因为它引用自模型开发器。因此，如果您希望一个受到更多限制的选择列表，就不能编辑它。反过来，您必须从复选框的源列表

中删除它，并通过在声明节点下进行声明来创建一个您自己的新选择列表。例如，您可以如下图所示创建三个选择列表。



要了解哪个值用于方法中的**单元尺寸**列表，可以使用**记录新方法**，然后将值从**标准**修改为**细化**，继而改为**粗化**，然后回到**标准**。单击**停止记录**，阅读自动生成的代码。**单元尺寸**属性的名称为 hauto，与**细化**、**标准**和**粗化**对应的值分别是 4、 5 和 6。如下面显示的自动生成的代码所示。

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

参见第 79 页 “按钮的模型数据访问”，了解更多**单元**尺寸的信息。

使用单位集代替选择列表

如果组合框将用于更改单位，则可以使用**单位集**代替**选择列表** （仍在组合框**设置**窗口的**选择列表**栏中将其选中）。

方程

方程对象可以在键入 LaTeX 方程编辑框中键入表达式显示 LaTeX 方程。



当离开该文本框时，会在下面的方程预览中显示渲染后的 LaTeX 语法。

线

使用**线**表单对象在表单中添加水平或垂直线，可以用于分隔表单对象组。对于水平线选项，您还可以添加显示线中的文本。



网页

网页对象可以将网页内容显示为用户界面的一部分。

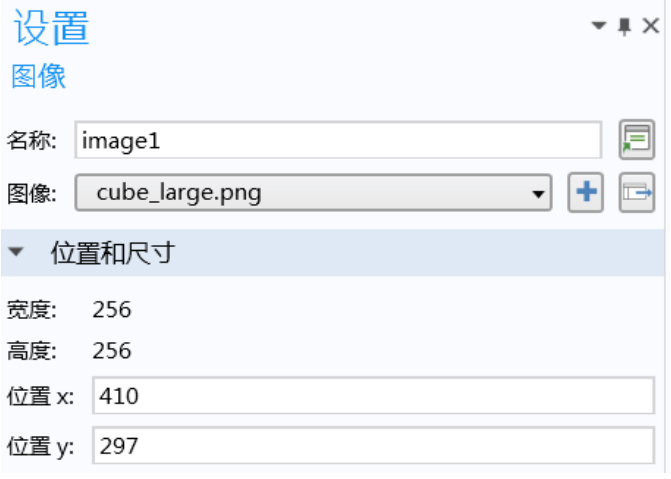


您可以从**源项**列表中通过四种不同方式指定页面的源：

- 使用默认的**页面**选项，在列表下的文本区中键入 HTML 代码，其中
- 分别使用 <html> 和 </html> 作为起始和结束标签。
- 使用 **URL** 选项链接到互联网上的网页。
- 使用**文件**选项指向一个本地文件源，其中包含 HTML 代码。在**文件**编辑框中键入文件名，或者单击**浏览**从本地文件系统中找到文件。
- 使用**报告**选项内嵌一个 HTML 的报告，这个选项不会激活浏览器预览。

图像

使用**图像**表单对象在表单中添加图像。图像对象与图形不同，前者包含的图片对象不能交互。通过下拉列表从库图像中选择一个图像文件，或者单击**添加图像到库并在这里使用**按钮从本地文件系统中选择文件。下图显示了一个引用**库**节点中定义的图像 cube_large.png 的图像对象的**设置**窗口。



如果您从文件系统中选择图像文件，该文件将内嵌到 App，并且添加到**库**节点下的**图**像列表中。

当您修改图像的 x- 和 y- 位置时，宽度和高度由图像文件决定。

> 您可以从剪贴板通过 Ctrl+V 来粘贴图像到表单窗口。例如，您可以从 PowerPoint® 幻灯片演示软件复制 - 粘贴图像。这些图像将自动添加到图像库，内嵌到 App。粘贴的图像的名称自动设置为：pasted_image_1.png、 pasted_image_2.png，等等。

视频

视频对象嵌入表单中的视频文件。受支持的视频文件格式为 MP4 (`.mp4`)、OGV (`.ogv`) 和 WebM (`.webm`)。

添加到表单后，**视频**对象通过图像以表单编辑器的形式表示，如下图所示。



下图显示了**视频**对象的**设置**窗口。



可用设置有：

- **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”。

初始静音选项专用于想播放初始状态下声音已关闭的视频。例如，如果视频设为自动开始，此选项使用户可以选择是否打开声音。用户可以从视频控件中启用声音 （如果已选中**显示视频控件**复选框），也可以在视频播放器中右键单击。

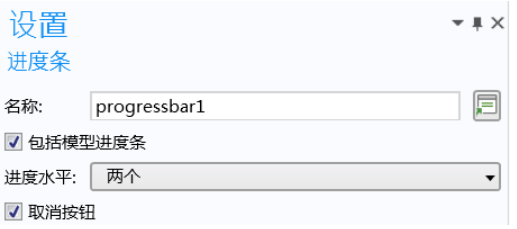
进度条

进度条对象显示定制的进度条，或者一组进度条，它们基于通过方法更新的值。使用进度条可以提供 App 运行的剩余时间反馈信息。下图显示了包含一个进度条的进度条对象的**设置**窗口。



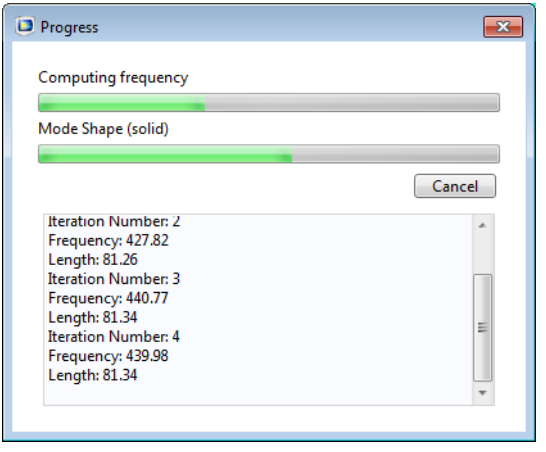
注意，在 App 的状态栏上可见的内建进度条由**主窗口**节点的**设置**窗口控制 。默认情况下，内建的进度条显示 COMSOL Multiphysics 内建核心算法的进度，例如，几何操作、网格剖分，以及求解。通过使用 setProgress 方法，您可以定制显示在内建进度条中的信息。请查阅第 267 页 “进度方法”，了解更多信息。

下图显示有两级进度条的进度条对象的**设置**窗口。

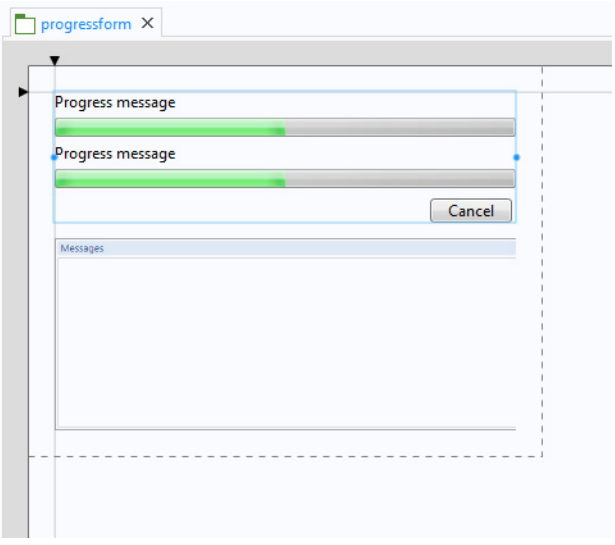


本例中，进度条对象是表格 progressform 的一部分，用于表述两级进度条和一个信息日志。

下图显示了正在运行的 App 中相应的进度对话框。



下图显示了表单 progressform。



下面的代码段显示用于更新进度条和信息日志的典型内建方法。

```
// 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");
```

在上面的示例中，更新两级进度条的核心功能位于如下调用语句：

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

请查阅第 226 页 “附录 E — 内建方法库”和第 267 页 “进度方法”，了解内建方法及其语法的更详细信息。

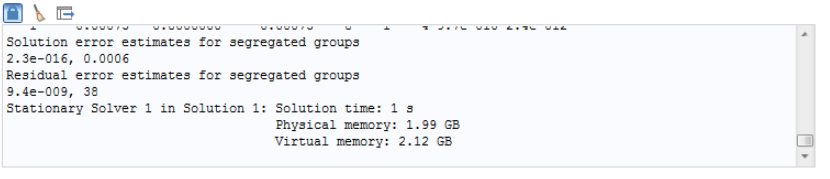
日志

日志表单对象添加一个日志窗口，显示来自内建的 COMSOL Multiphysics 核心算法的信息，例如几何操作、网格剖分，以及求解。

包含标准日志工具栏复选框默认选中，当它被选中时，会在 App 中包含您可以在 COMSOL Desktop 中看到的日志窗口的工具栏。



下图显示一个包含日志窗口的 App 用户界面的一部分。



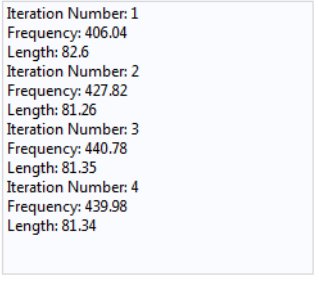
信息日志

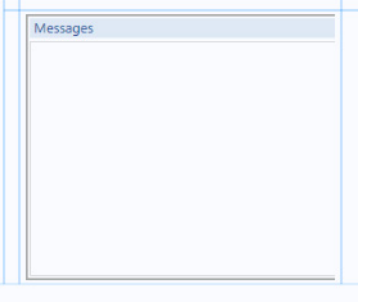
信息日志添加一个窗口，其中您可以显示信息提示用户关于 App 运行的操作。可以通过语法 message(String message) 实现内建的信息方法来实现这个特征。第 251 页 “GUI 相关的方法”。



默认会选中**包含标准信息日志工具栏**，当它被选中时，会在 App 中包含您可以在 COMSOL Desktop 中看到的**日志**窗口的工具栏。**显示 COMSOL 信息**复选框在默认情况下也是被选中的，可以显示来自内建的 COMSOL Multiphysics 核心算法的信息，例如几何操作、网格剖分，以及求解。清除这个复选框将只能显示来自 App 自身的信息。

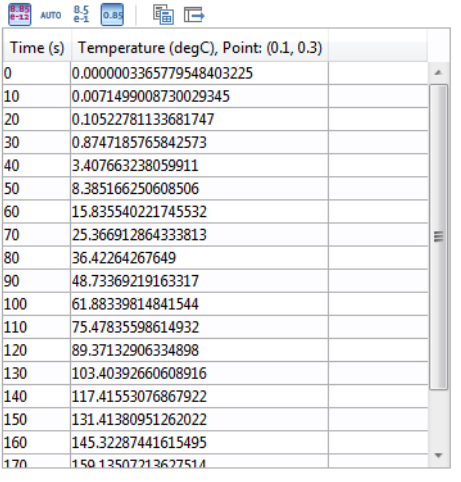
下图显示定制的信息窗口，包括来自方法的收敛信息 （左）和对应的**信息日志**表单对象 （右）。





结果表

结果表对象用于在表格中显示数值结果。



Time (s)	Temperature (degC), Point: (0.1, 0.3)
0	0.0000003365779548403225
10	0.0071499008730029345
20	0.10522781133681747
30	0.8747185765842573
40	3.407663238059911
50	8.385166250608506
60	15.835540221745532
70	25.366912864333813
80	36.42264267649
90	48.73369219163317
100	61.88339814841544
110	75.47835598614932
120	89.37132906334898
130	103.40392660608916
140	117.41553076867922
150	131.41380951262022
160	145.32287441615495
170	159.13507213627514

结果表数据的源是**结果**下的**派生值**或**表格**节点的子节点。下图中使用一个**表格**节点用作源 （通过在树中选择这个选项，然后单击**作为源使用**按钮。）



结果表工具栏

包含标准结果表工具栏复选框默认为选中，这种情况下，会包含一个提供以下按钮的工具栏：

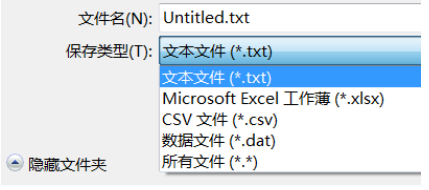
- **全精度**
- **自动标记**
- **科学标记法**
- **十进制计数法**
- **复制表和题头到剪贴板**
- **导出**

导出按钮用于导出为下列文件格式：

- 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：



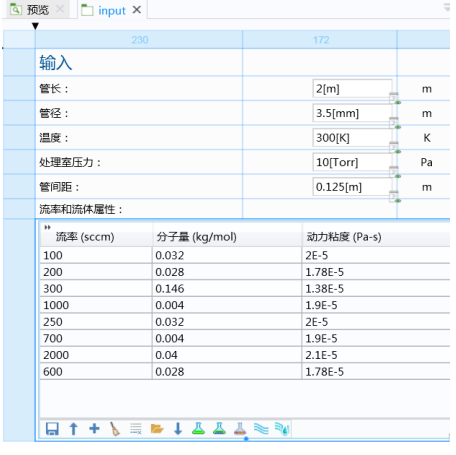
从方法控制结果表

有一个内建的方法 useResultsTable() 来修改在特定的结果表对象中显示哪个表格。参见第 288 页 “GUI 相关方法”，了解更多内建方法相关信息。

一个**表单**类型的表单对象用来在一个或多个子表单中组织 “主”表单。如果需要内嵌子表单，您可以从**设置**窗口的**表单**引用中通过选择想要链接的表单来创建链接。下图显示的示例中，表单 main 的其中一个单元链接到表单 input。



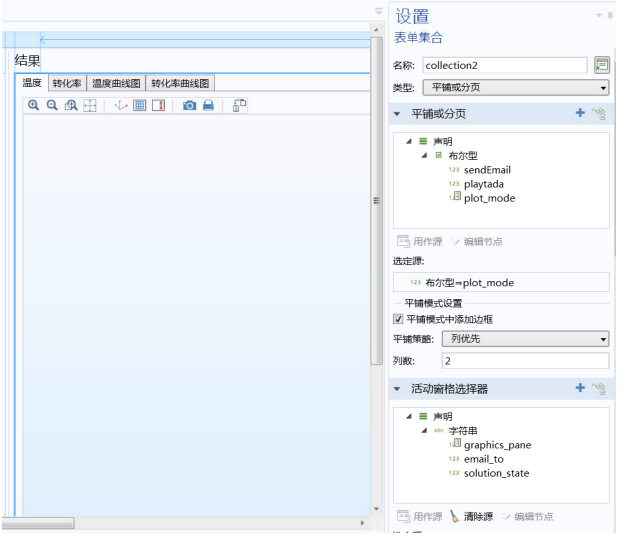
下图显示引用的表单 input。



如果您使用的是栅格布局模式，可以使用功能区中的**提取子表单**按钮快速创建子表单。请参见 105 页的 “提取子表单”。

表单集合

表单集合对象在一个主表单中包含一些表单，或者面板。在本例中，一个主窗口中以选项卡形式显示了四个表单。



有四种不同布局选项，从**类型**列表，可以选择：

- **选项卡**，默认设置，以选项卡面板形式显示表单。
- **列表**，在表单面板左侧显示列表，您可以从中选择要显示的表单。
- **栏**，以单独的栏显示每个表单。
- **平铺或者选项卡**，根据一个布尔变量的值显示两种方式之一，参见本节后面的描述了解更多信息。

在**面板**栏的**使用选定表单作为面板**的列表中，每个表单代表一个面板。这些将会在 App 中按照它们在列表中的顺序显示，您可以通过单击右侧的上移和下移修改顺序。

您可以在**激活面板选择器**栏通过链接到字符串变量来激活控制选项卡 （或列表条目）。

在上面的示例中，如果您在一些地方将会运行的方法 （例如，按钮方法）修改了面板选择器 pane 的值，将会激活对应于新值的面板，如下面的示例所示。

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

对于将**类型**设置为**栏**的表单集合，**激活面板选择器**没有效果。使用**激活面板选择器**是一个可选项，只用于如果您想要通过单击选项卡之外的一些方法来控制激活选项卡的情况。如果要删除用作**激活面板选择器**的字符串变量，请单击树下的**清除源**工具栏按钮。

平铺或者选项卡选项根据设置窗口顶部的**平铺或者选项卡**栏中作为源使用的一个布尔变量的值按照两种方式之一来显示。



选项卡模式与类型设置为选项卡的表单集合相同。在平铺模式，所有表单同时显示在单元格中，平铺模式的布局可以通过在子栏平铺模式设置中的设置进行控制。

卡片堆

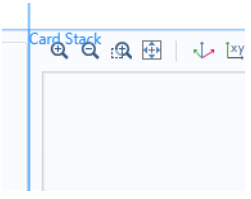
卡片堆是包含卡片的表单对象。

卡片是另一种类型的表单对象，一种只用于卡片堆的上下文的对象。翻阅卡片堆中的卡片，每次只显示一张卡片。您需要将卡片堆与控制显示哪张卡片的数据源进行关联。每张卡片指定一个值，与卡片堆的数据源进行比较。卡片堆显示出匹配值的每一张卡片。如果没有能够匹配的卡片，则什么都不显示。

使用卡片堆翻阅图形对象

考虑这样一个 App，其中显示给用户的图形依赖于标量变量的值。这个变量可能随着用户单击而变，例如，单选键。变量也可以随着计算得到的值进行变化，例如，模型树中**全局计算**节点的值。

下图显示了表单编辑器中的卡片堆。



在这个示例中，卡片堆包含带有图形对象的卡片。

下图显示了一个拥有 5 张卡片的**卡片堆**设置窗口，其中有一个字符串变量 display 作为**激活卡片选择器**。

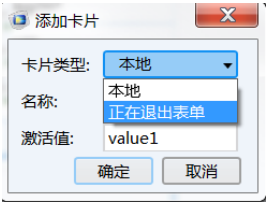


通过在**卡片**栏中单击卡片表中的卡片，随着单击表格下的不同工具栏按钮，您可以实现下列卡片操作：

- **删除**
- **编辑**
- **未来卡片**
- **生成副本**

在卡片堆**设置**窗口，卡片栏的表格包含**卡片**列中的卡片，以及关联的**激活值**列中的激活值。卡片堆基于它们的激活值来决定显示哪张卡片。本例中，激活值为字符串 geometry、 velocity、 particle1，等等。

单击**添加卡片**按钮会显示如下对话框。

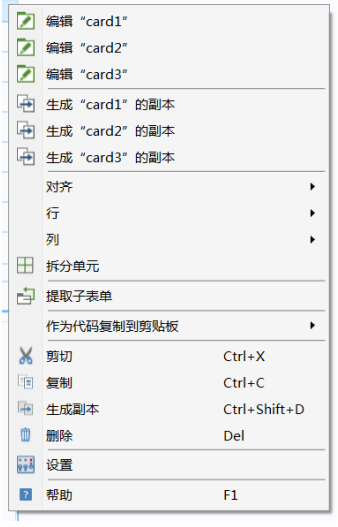


默认情况下，**卡片类型**设置为**局部**，意味着在包含它的卡片堆对象中定义为局部卡片。如果**卡片类型**设置为**已有表单**，您就可以选择一个已经作好的表单。**已有表单**的设置是直接从表单编辑器单击对应的卡片堆**设置**窗口中的节点或者单击卡片栏中的**编辑**按钮来实现。

下图显示了单击**卡片**栏表格中的**编辑**后**卡片**的**设置**窗口。

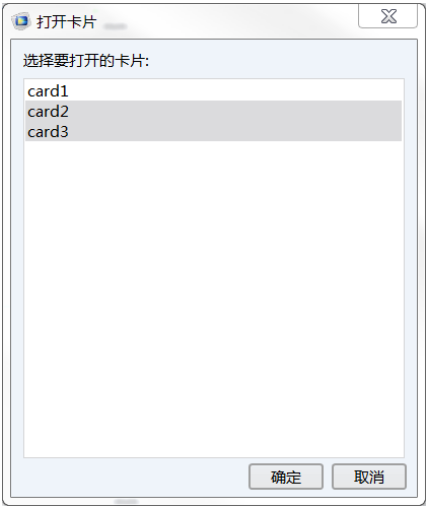


如果需要访问局部定义的卡片，可以右键单击表单窗口中的卡片堆，从卡片堆中选择不同的卡片，如下图所示。

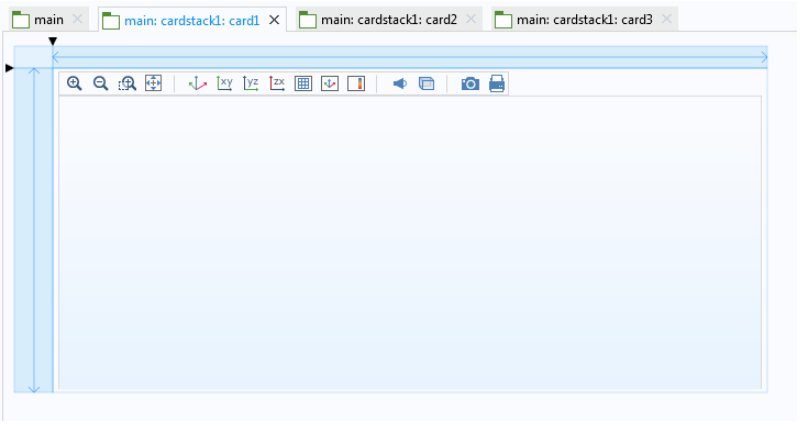


从这个菜单，您还可以复制卡片。

要编辑卡片，您也可以使用 Alt+ 单击，打开对话框，可以同时选择多个卡片。



下图显示 card1 及其图形表单对象。



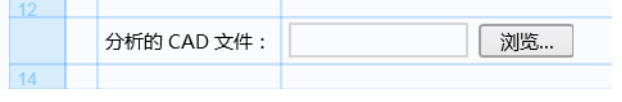
文件导入

文件导入对象用于显示与输入框关联的文件浏览器，通过浏览找到文件或输入路径和名称。当事先准备的 App 中没有文件时，可以用它来供 App 用户在运行时启用文件导入。

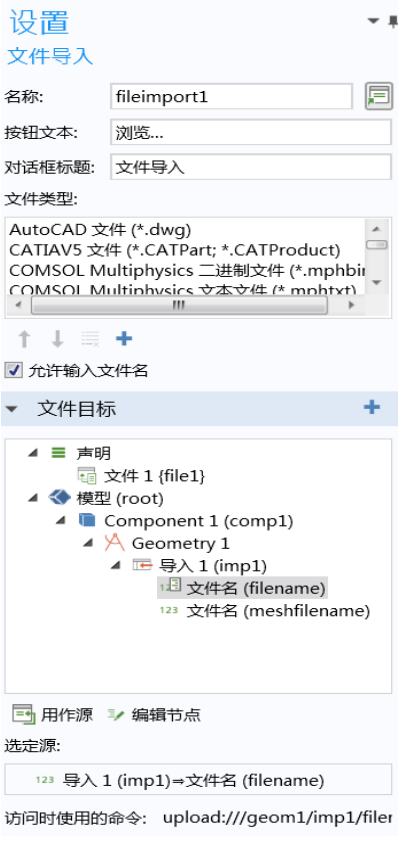
考虑这样一个 App，其中可以在运行时选择一个 CAD 文件并导入，如下图所示。



下图显示对应的**文件导入**对象。



文件导入对象的**设置**窗口有**文件目标**栏，该栏中您可以选择任意允许输入文件名的树节点。如下图所示，其中选择了几何**导入**节点的**文件名**。



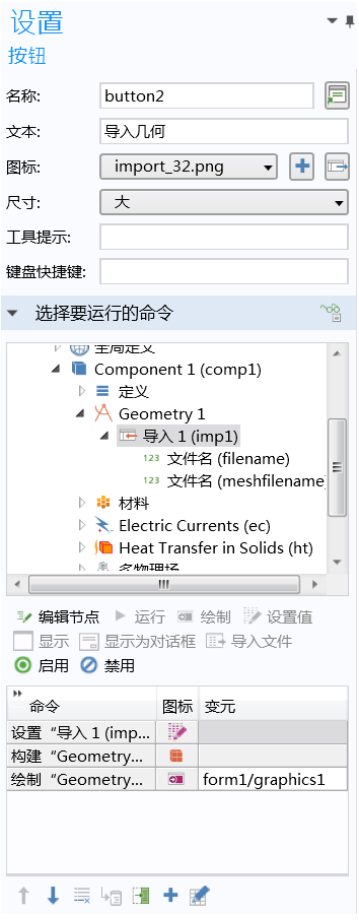
在这个 App 中，**文件类型**表格指定只允许使用 CAD 文件。您还可以进一步通过单击文件类型列表下的**添加**和**删除**按钮来控制允许哪些**文件类型**。单击**添加**按钮显示的对话框如下图所示：



文件导入对象的其他用法

如果不需要文件路径和名称的输入框，则可以使用其他的文件导入方法来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，或者按钮。在这种情况下，您可以在按钮或这些项目的命令序列中使用**打开文件**命令。

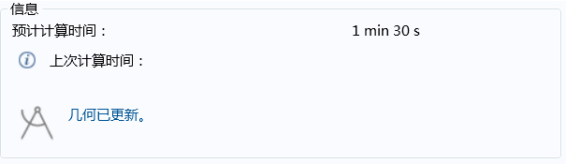
下图显示用于导入一个 CAD 文件的按钮，以及它的**设置**窗口。



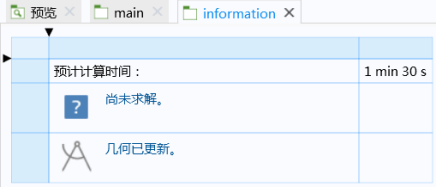
文件导入对象还可以引用**文件**声明。参见第 122 页 “文件声明”了解更多信息；以及第 236 页 “附录 C – 文件处理和文件方案语法”，了解常用的文件处理方面的更多信息。

信息卡片堆

信息卡片堆是一种特殊类型的**卡片堆**对象，用于显示由用户给定的 App 和解之间的相互关系的信息。下图显示了一个运行中的 App 的一部分，其中用到了信息卡片堆显示预期的计算时间信息。



对应的表单对象如下图所示：



下图显示设置窗口，其中字符串变量 solution_state 用作源。



与**卡片堆**对象类似，但是对于**信息卡片**，每个卡片都有一个图标和文本。上图中，字符串变量 nosolution、 inputchanged 以及 solutionexists 的值控制显示哪个信息卡片。

本例中，信息卡片堆附有一个数据显示对象，其中模型树中的**期望计算时间**的信息节点用作源。下图显示了它的**设置**窗口。

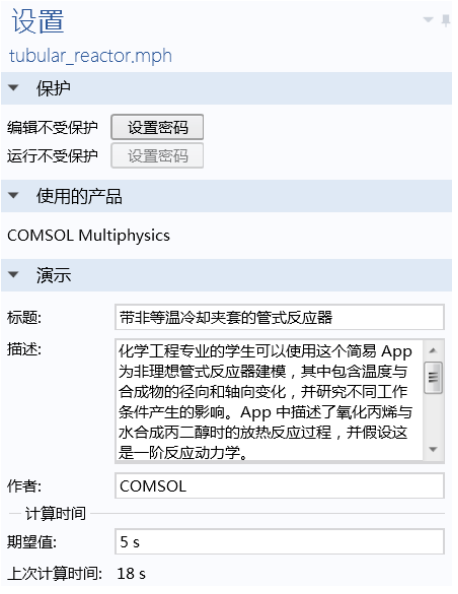


注意，模型树中的信息节点只会在使用 App 开发器才显示出来。当可以使用时，它们会显示在表单对象的设置窗口中的**源**栏。

您还可以在每个研究下找到**上次计算时间**的信息节点。在**模型**节点下直接找到的**上次计算**信息节点对应于上次计算的研究使用的计算时间。

信息节点可以用作输入框、文本以及数据显示等对象的源。对于输入框和文本对象，为了可以访问信息节点，需要清除**可编辑**复选框。

期望计算时间从 App 树的根节点获得数据，如下图所示。



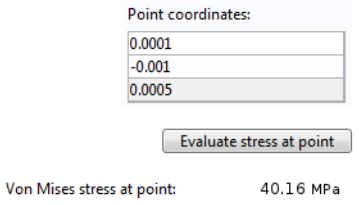
如果计算时间受方法支配，例如，如果重复调用同一个研究，您可以通过使用内建方法 timeStamp 和 setLastComputationTime 来手动测量计算时间。请查阅 中第 273 页 “日期和时间方法”，了解更多信息。

数组输入

数组输入对象有一个用于输入数组或矢量值的输入表格，支持字符串数组作为数据源。您可以添加可选的标签、记号和单位。

使用数组输入对象实现三维点坐标输入

考虑这样一个 App，其中用户需要输入点的三维坐标用于计算应力。下图显示一个带有一个数组输入、一个按钮、一个文本标签，以及一个数据显示对象的 App 的截屏。



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称为 samplecoords 的源，它是一个双精度数类型的一维数组。在创建**数组输入**对象之前，通过声明**一维双精度数组**创建出这个数组，其**设置**如下。



在数组输入对象的**设置**窗口：

- **长度**，键入数据的长度，通常是一个正整数。默认为 3。
- **显示矢量**列表，选择**表格** （默认）将数组分量显示为表格，或者选择**分量**将每个数组分量显示为单独带标签的输入框。
- **值**表格，键入数组分量的初始值。
- **布局选项**，提供添加可选的标签和单位的设置。

本例中，当用户单击标记为**在点计算应力**的按钮时，运行下面这个方法。

```
with(model.result().dataset("cpt1"));
  set("pointx", samplecoords[0]);
  set("pointy", samplecoords[1]);
  set("pointz", samplecoords[2]);
endwith();
```

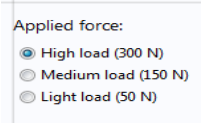
其中值 pointx、 pointy，以及 pointz 将被用作计算应力的坐标。

单选按钮

单选按钮对象有固定数量的选项，您可以从中选择一个。它常用于您只含有少量选项的情况。

使用单选键选择载荷

考虑这样一个 App，其中用户可以从三个预定义载荷中选择一个，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。



在**初始值**列表，选择单选键如何决定初始值，选项包括**来自数据源**、**第一个允许值**（默认），以及**定制值**。对于**定制值**选项，需要从选择列表给定的允许值列表中选取一个。

在选择列表栏，您可以添加选择列表，给单选键提供允许值，其中每个允许值代表一个单选键。

单选键的名称从与之关联的选择列表的**显示名称**列获取，下图显示本例的选择列表。



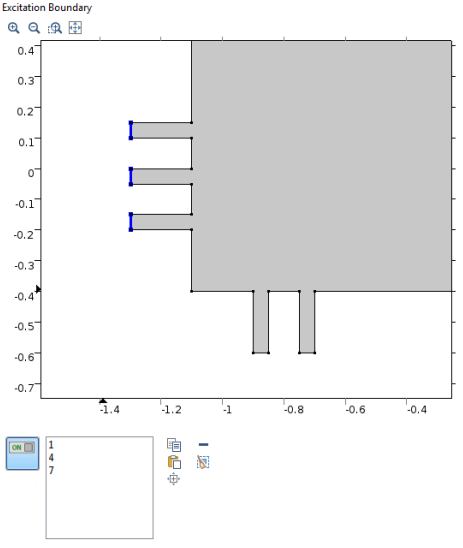
使用单位集代替选择列表

如果单选按钮将用于更改单位，则**单位集**可以代替**选择列表** （您让在单选按钮对象设置窗口的**选择列表**栏将其选中）。

选择输入

在模型开发器中，指定名称的选择可以让您分组域、边界、边，或者点，用于分配材料属性、边界条件，以及其他模型设置。您可以在**组件 > 定义**节点下通过添加子节点来创建各种类型的选择。这些可以在整个模型组件中重用。**显式**选择类型可以让您基于实体序号分组域、边界、边或点。

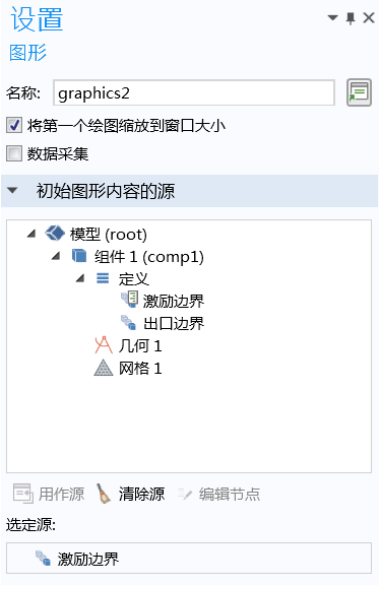
在 App 开发器中，您可以允许 App 的用户交互式地通过**选择输入**或**图形**对象来修改属于**显式**选择的实体。在下面的示例中，内嵌模型包含通过**显式**选择定义的边界条件。**选择输入**和**图形**对象用来让用户选择入射波激励的边界。



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

为了可以通过单击直接选择图形对象，您可以链接一个边界到用于分组边界的显式选择，如下图所示。选择这个显式选择，然后单击**作为源使用。**

在下图中，有两种显示选择：**激励边界**和**退出边界**，且图形对象 graphics2 链接到**激励边界**选择。



当一个图形对象按照这种方式直接链接到一个显示选择时，图形对象显示几何，并且用户可以通过在边界上单击形成交互。选中的边界将添加 （或移除）到对应的显式选择。

为了让它能够通过数字进行选择，您可以链接一个选择输入对象到显式选择，如下图所示。



在选择输入对象中，您可以复制、粘贴、删除、清除，并且缩放到选择。

您可以选择使用图形对象作为选择的源，且不指定任何选择输入对象。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

文本对象是从字符串变量获取默认文本的文本框，下图显示了一个文本对象的**设置**窗口。



从**源**栏的树中选择字符串变量，然后单击**作为源使用**。在**值**编辑框中键入初始文本。默认情况下，**初始值**文本从这个编辑框获得。如果要使用字符串变量作为**初始值**文本，请将初始值设置改为**来自数据源**。

默认已清除**可编辑**复选框，如果选中它，文本对象可以用于在运行的 App 中键入备注。如果用户修改文本，它将被储存在用作数据源的字符串变量，不管**初始值**如何设置。

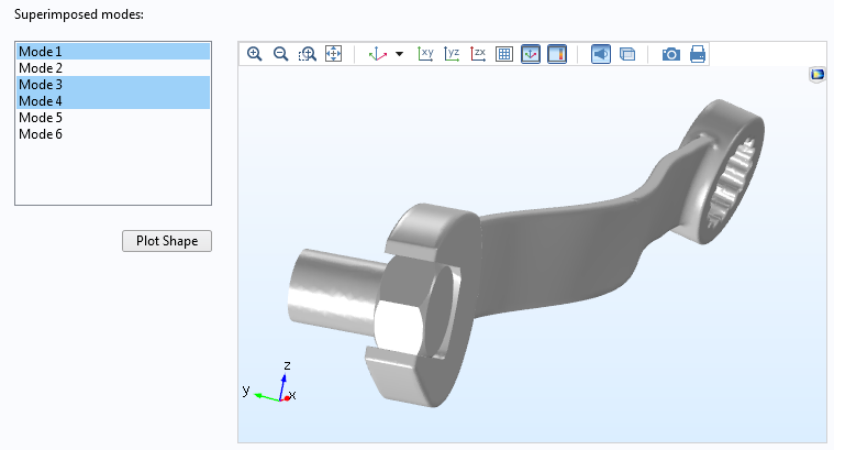
默认选中复选框**换行**，清除这个复选框将禁用文本换行。如果文本不适合将显示滚动条。

列表框

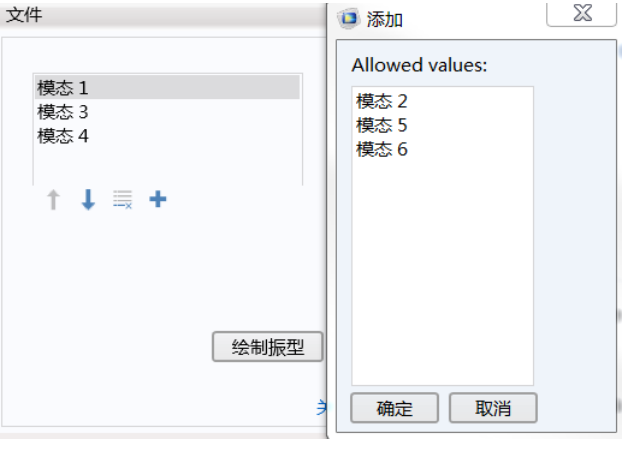
除允许同时选择多个选项外，**列表框**对象与单选键对象类似。

使用列表框叠加振动模态

考虑这样一个 App，其中某个机械零件的前六个模态可以进行叠加，并且通过从列表框中进行选择来图形化，如下图所示。



另外一种做法，下图显示列表可以显示为对话框。



本例的列表框的**设置**窗口如下图所示。



在列表中**选择值**允许您从两个不同的方法中选择，**列表框**或者**对话框**。

您可以使用任意标量或数组声明作为源，从树中选择，然后单击**作为源使用**。如果您使用字符串数组为源，可以在运行的 App 中使用 Shift+ 单击或 Ctrl+ 单

击从列表中选择不止一个项目。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，其**设置**窗口如下图所示。



在选择列表栏，您可以添加选择列表，给列表框提供允许值。下图显示本例使用的选择列表。



振动模态 1-6 对应于小型刚体的模态，在这个 App 中不是我们关注的，因此**值**列从 7 开始。选择列表可以让您隐藏模型中的模态值，只给出**显示名称**列中的字符串。头六个非刚体模态命名为 Mode 1， Mode 2，等等。

下面的方法使用 COMSOL Multiphysics 中的算子 with() 来图形化叠加模态。本例进行了一些简化，因为其中忽略了模态的幅值和相位的效应。

```
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");
```

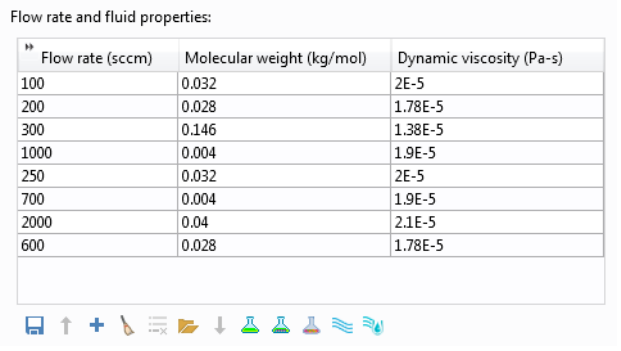
假设用户通过使用选择框选择了模态 1、 3 和 5，上面的方法创建一个表达式 with(1,u)+with(3,u)+with(5,u)。这个表达式用于在位移图中显示 x 位移 （因变量 u）。以同样的方法，方法自动创建与 y 和 z 位移关联的变量 v 和 w。注意上例的结果中使用的命令 with() 与第 259 页 “With，获取和设置”中描述的用于简写语法的内建命令 with() 不同。

使用单位集代替选择列表

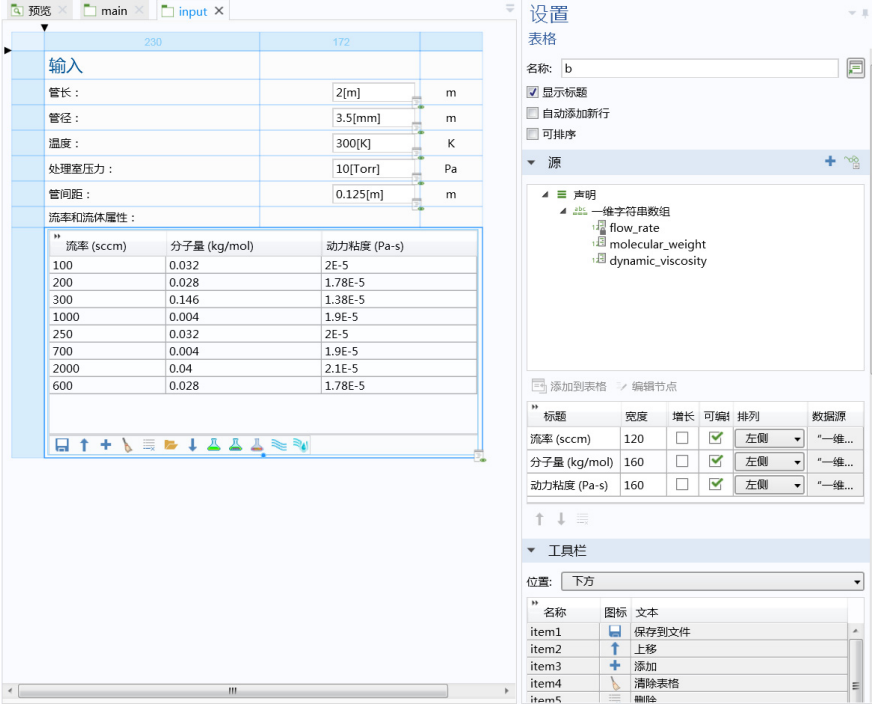
如果列表框将用于更改单位，则**单位集**可代替**选择列表**使用 （在列表框**设置**窗口的**选择列表**栏中仍将其选中）。

表格

表格对象代表一个行和列可以用于定义输入或输出的表格。下图显示了一个运行中的 App 示例，其中使用一个表格对象来接收三列输入。



下图显示对应的表格对象及其**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**作为源使用**。

三个复选框控制表格的整体外观：

- **显示题头**
- **自动添加新行**
- **可排序**

自动添加新行复选框确保当用户填写表格时总会自动在后面显示一个空行。如果所有用于表格的源的一维字符串数组在各自的声明**设置**窗口的**新单元值**都不为空值，则此功能失效。在这种情况下，如果在关联的表格工具栏上有**添加**按钮，则可以单击它来添加新行。

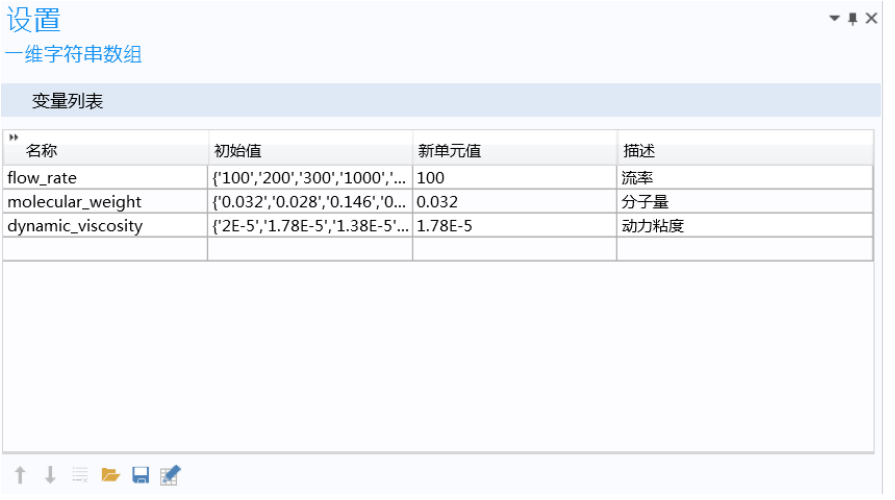
可排序复选框使得它可以通过单击列标题来进行排序。

源项栏包含有以下五列内容：

- **题头**
- **宽度**
- **增长**
- **可编辑**
- **排列**
- **数据源**

这个表格中的每行定义表格对象中的一列内容。选项**增长**允许各个列在调整表单大小时增长。该选项仅适用于栅格模式并且表格的**水平对齐**需设为**填充**。

对应于这三列的定义行的初始值的字符串数组如下图所示：



工具栏

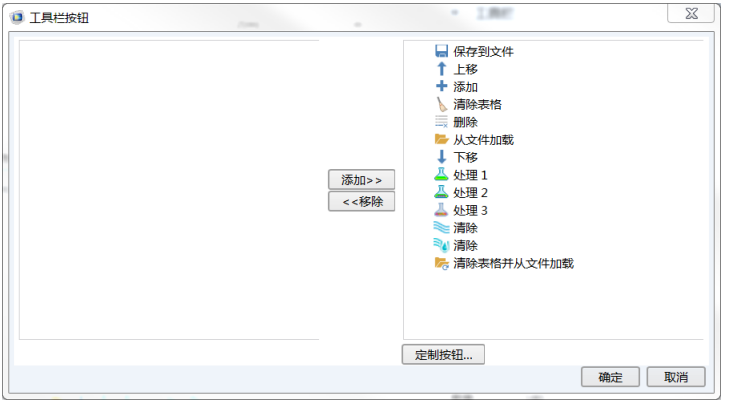
在这个栏中，您可以选择哪个工具栏按钮可以用来控制表格的内容。**位置**列表定义工具栏相对于表格的位置，提供下列选项：

- **下方**
- **上方**
- **左**
- **右**

如果要在工具栏中添加按钮，可以单击表格下方的**添加工具栏**按钮：



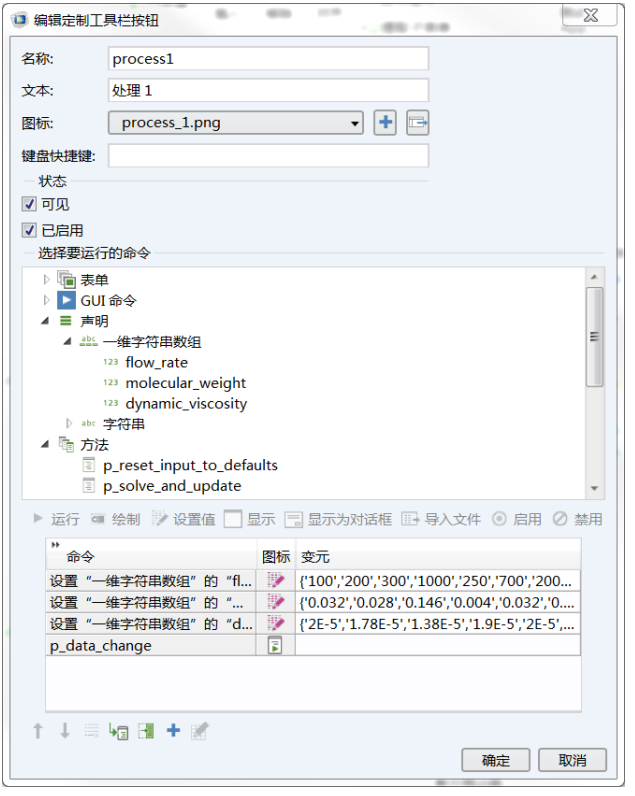
然后会显示下面这个对话框：



您可以添加下列按钮：

- **上移**
- **下移**
- **添加**
- **删除**
- **清除表格**
- **清除表格和从文件加载**
- **从文件加载**
- **保存到文件**

此外，您可以通过在工具栏按钮对话框中单击**定制按钮**来添加定制的按钮。下图显示了用来定义一个定制按钮的**编辑定制工具栏按钮**对话框。在这种情况下，按钮 **item1** 用于设置确切过程的默认值。

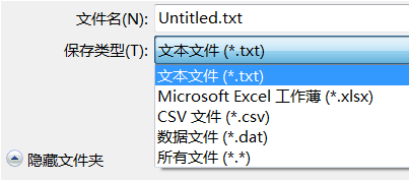


选择运行的命令栏与菜单、功能区和工具栏项目，以及按钮的设置类似。

从文件加载和**保存到文件**按钮用于从下列文件格式加载或保存到它们：

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

如下图所示：



CSV 文件允许使用的分隔符为逗号、分号和 Tab 键， DAT 和 TXT 文件允许使用的分隔符为空格和 Tab 键。

滑块

滑块是一种使用滑块控制来选择数值输入的表单控件。

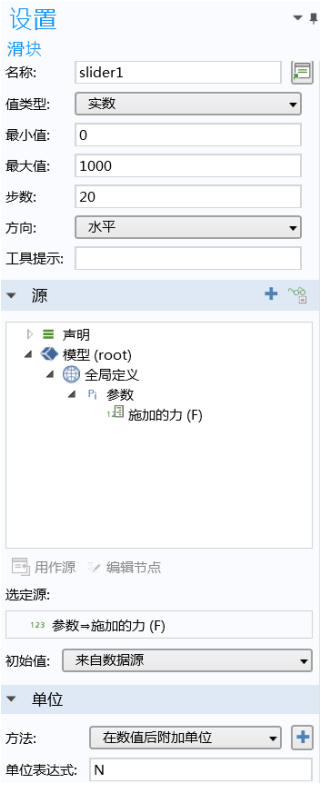
使用滑块来改变结构载荷的幅值

考虑这样一个 App，其中载荷的大小可以通过滑块控制来改变，如下图所示。



在本例中，滑块后带有一个输入框，用于显示选定的值。

下图显示了滑块的**设置**窗口。



本例中，滑块使用全局参数 F 作为源，您可以选择任意参数、变量，或者声明的标量变量作为源。从树中选择，然后单击**作为源使用**。

从值类型列表中，可以选择**整数**或**实数** （默认），依据滑块的数据源的数据类型确定。

您可以通过定义滑块的**最大值**和**最小值**，以及**步数**来决定滑块的数据源的取值范围。方向可以是水平，也可以是竖直。您还可以设置工具提示，当在滑块上悬停时将显示出来。**添加单位到数**选项可供您关联一个单位到滑块，当一个数传递给源变量之前，这个单位使用标准括号标记法来进行关联，例如 [N]。上面的示例中，输入框和滑块都激活了**添加单位到数**选项。

在**初始值**列表中，选择**来自源数据**或者**定制值**来确定滑块的初始值。

超链接

超链接对象在表单中嵌入一个超链接。下图显示一个超链接示例。



下图显示对应的**设置**窗口。



超链接对象支持您在 Web 浏览器中可以使用的 URL 类型，包括：

- 网页：当用户单击网页超链接时，其将在用户的默认浏览器中打开。 URL 字符串格式应为：`protocol://address`，其中，协议是传输协议；例如，HTTP 或 HTTPS。 Web 地址可以是一部分，也可以是完整地址，但建议使用完整的 Web 地址。
- 电子邮件：电子邮件地址格式指定为：`mailto:emailaddress`。这将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，收件人栏设为指定的地址。这种以交互方式从 COMSOL App 发送电子邮件的方法与使用内置方法不同。参见第 298 页的 “电子邮件方法” 了解电子邮件内置方法的更多信息。

工具栏

工具栏对象包含带有工具栏按钮的工具栏明细表。下图显示了一个带有**另存为**、**计算**，以及**绘图**按钮的工具栏。



下图显示了这个工具栏的**设置**窗口。



工具栏项表中的每一行包含一个**项目**，对应于工具栏按钮，或者**分隔符**。使用表格下方的按钮添加项目或者分隔符、修改行序，或者删除行。单击**编辑**按钮

可以显示与每行关联的**设置**窗口。下图显示了项目 **item1**，即**另存为**项目的设置窗口。



文本框中的文本将显示为当在工具栏按钮上悬停时的工具提示。**图标**、**键盘快捷键**，以及**选择运行的命令**代表按钮对象的同一个功能。请查阅第 42 页 “按钮”，了解更多信息。

定距片

定距片对象在用户界面中不可见，只能用于栅格布局模式。它定义固定大小的空间，用来确保相邻的表单对象有足够的空间显示各自的内容。通常情况下，您可以表格或者图形对象后使用定距片来确保它们可以正确地显示出来。如果

用户修改了窗口的尺寸，改到比定距片的尺寸还要小，则窗口的有效尺寸通过显示滚动条来显示。下图显示了一个定距片的设置窗口。



附录 B — App 之间进行复制

App 树中的很多节点可以在 App 之间进行复制和粘贴，包括：表单、表单对象、方法、 Java® 实用函数、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，以及功能区项。

当您在不同的 App 之间复制和粘贴表单和表单对象时，复制的对象包含对其他对象和项目的引用。在复制的 App 中，这样的引用不一定有意义。当从剪贴板粘贴对象时，会应用下面这些规则：

- 当复制对象时，将包含表单对象中的声明引用，但并不一定会被粘贴。如果没有兼容性声明，才会粘贴。如果存在兼容性声明，将会被后者取代。兼容性声明是指具有相同名称和类型的定义。例如，字符串声明与整数声明不兼容。已有的声明可能有无效默认值，但进行粘贴时不会进行这种检查。
- 引用的全局参数可能会有不同单位，但仍然会被认为是兼容的。
- 当复制对象时，不会自动包含直接引用自另一个表单对象的表单或表单对象。如果存在相同名称，直接引用将指向已有的对象。如果原始引用存在于复制的对象中，则在引用中使用该对象，无论是否已经存在有相同名称的其他对象。复制引用的名称将被修改，避免名称冲突。
- 模型树中的对象将不会被自动复制。例如，指向几何的图形对象，或者指向由数据访问暴露出来的底层设置的输入框。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。
- 对不存在对象的引用将会在粘贴时被删除掉。但是有一个例外，在按钮的命令序列中将保留所有命令，如果它们指向不存在的引用，则会加上无效标记。
- 复制 - 粘贴操作中保留局部方法。然而，不会对方法的代码进行更新。当复制全局方法时，采取同样的处理方法。
- 按钮或菜单项的命令序列中命令的变元将会保留。
- 如果可行，所有的图像引用将自动复制和添加到图像库。如果已经存在相同名称的图像，则会用它来替代复制的版本。
- 不会自动复制引用的文件、声音或者方法。但是，可以手动复制和粘贴方法。

- 所有粘贴对象，如果存在与已有的对象名称冲突，将会被重命名。对这些重命名的对象的引用将自动得到更新。

附录 C — 文件处理和文件方案语法

文件处理可能是 App 的一个重要特征。例如， App 可能需要含有实验数据的电子表格作为输入，导入 CAD 文件，或者生成报告并导出。 App 开发器提供了读取和写入整个或部分文件的工具，根据运行 App 的系统不同，实现方法会有较大差异。文件系统可能存在差异，例如运行 COMSOL Multiphysics 和开发 App 的计算机，以及安装 COMSOL Server 以及运行 App 的计算机。

COMSOL Server 文件处理

通常情况下，您通过网页浏览器或者基于 Windows® 的 COMSOL Client 运行 App 时，不能直接在本地路径读和写文件。 App 及其方法运行在服务器上，并不清楚客户端这边 （运行网页浏览器或 COMSOL Cleint 的计算机）的文件系统。

然而，当使用网页浏览器和 COMSOL Client 时，还是有办法实现客户端文件系统上的读和写文件。

一个**文件导入**对象用来询问用户所需操作的文件。用户因此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 的文件系统，并使它成为 App 及其方法可以使用的对象。例如，可以使用这种方法让用户在运行时提供 CAD 文件或实验数据文件。参见第 241 页 “文件导入” 章节。

在命令序列 （例如按钮）中，您可以通过运行**导出**或**报告**节点，导出由内嵌的模型生成的数据。参见第 249 页 “文件导出” 章节。

使用文件命令保存和打开文件

在命令序列中使用的编辑器树中，**文件命令**文件夹包含保存和加载 App 和文件，以及退出 App。



打开文件命令将从服务器上挑选由方法、模型生成，或者内嵌在 App 中的文件，并在客户端上使用关联的 App 打开它。例如，这可以用来打开客户端文件系统中的 PDF 文件，或者在客户端上显示从模型导出的文本文件或图片。下图中，**打开文件**命令用来打开 App 的 PDF 文档。相应的 PDF 文件通过存储在**库** > **文件**节点中来嵌入到 App。这些文件使用下一部分 （第 266 页的 “文件方案语法”）描述的 `embedded:///` 文件方案语法来引用。

如果需要从方法打开文件，可以使用内建方法 fileOpen，参见第 278 页 “系统方法”。

如果需要保存文件，可以使用与**打开文件**类似的**文件另存为**命令。它将获得来自服务器文件系统的文件，向用户显示一个**另存为**对话框，用户可以浏览客户端上的位置，保存文件。这个操作相当于在网页浏览器中从链接下载文件。在下图中，保存文件命令用于保存存储在**库 > 文件**节点中的 CAD 模型。



如果需要从方法保存文件，可以使用内建方法 fileSaveAs，参见第 297 页 “内建 GUI 方法 “。

保存 App 和 **App 另存为**命令可以在一些表单对象中使用，其中 **App 另存为**命令将显示一个另存为对话框，用户可以指定客户端上的文件路径，然后完整地保存整个 App。

类似的，**在服务器上保存 App** 和 **App 在服务器上另存为**命令可以用来将整个 App 保存在服务器的文件系统中。参见第 297 页 “内建 GUI 方法”，了解相应的内建方法的信息。

总之，支持从客户端文件系统进行文件的上传和下载，但是 App 不能在后台不经过用户浏览到文件的源或目标路径就悄悄地实现。

模型命令

在命令序列中使用的编辑器树中，**模型命令**文件夹中包含两个命令：**清除所有解**和**清除所有网格**。这些命令分别用来在保存 MPH 文件之前清除解和网格数据，从而减少文件的大小。



文件方案语法

为了 App 的可移植性， App 开发器允许您通过文件方案来使用虚拟文件位置。文件方案可以看作文件在文件系统中的指针，但是 App 不需要知道文件实际储存在哪里。不同的目标需要使用不同的文件方案：

- `user` 文件方案用于同一个用户运行 App 的不同次数中保持文件一致。
- `common` 文件方案功能类似，但是它用于所有用户之间共享的文件。
- `temp` 文件方案用于当 App 关闭后会立即删除的文件。
- `embedded` 文件方案用于在 App 本身中储存文件，当您希望使得 App 自包含并发送给其他人时会非常有用。
- `upload` 文件方案用于用户在运行 App 时上传的文件，例如用户浏览到的 CAD 文件。

下表列出所有可用文件方案的总结。

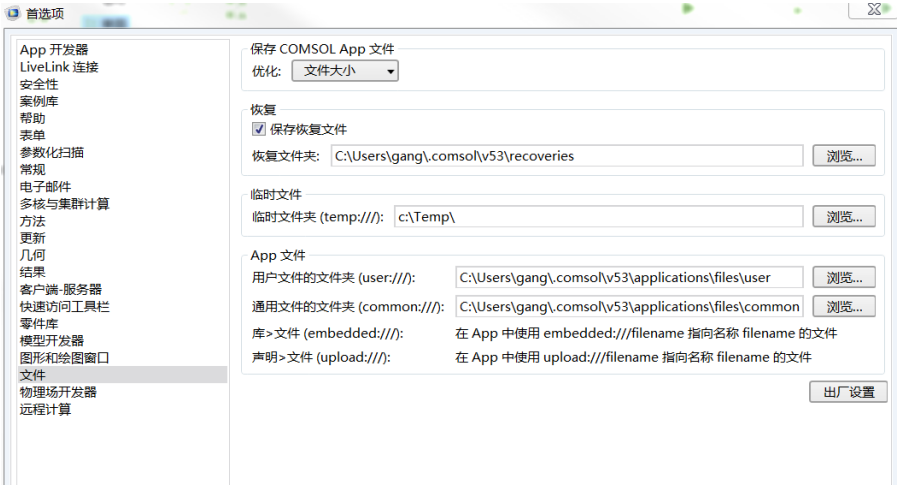
方案	指向	默认路径	典型用法
embedded:///	使用库 > 文件 内嵌在 App 中的文件	N/A	实验数据、CAD 文件、网格文件、插值数据
upload:///	用户运行时上传的文件	由文件声明的设置窗口中的目标路径决定	实验数据、CAD 文件、网格文件、插值数据
temp:///	对每个启动的 App 实例都是唯一的随机临时目录中的文件。当 App 关闭时，这些文件会被删除。	用于保存临时文件，由首选项 > 文件中的设置决定的随机子路径	由命令序列或方法产生，或者在客户端上保存输出 （用于 COMSOL Server）的临时文件
user:///	当前用户的所有 App 共享文件夹中的文件	由首选项 > 文件中的设置决定	会话之间从方法保存的输出
common:///	所有用户共享的文件夹中的文件	由首选项 > 文件中的设置决定	很多用户或 App 之间共享的文件

参见第 154 页 “库”，了解更多通过 embedded:/// syntax 提供库节点中可以访问文件的信息。

下表总结了不同文件方案的用法，表中的复选标记表明该方案可以使用，(r) 则表示这是一个推荐方案。

用法	嵌入	上传	临时	用户	公用
文件用作输入	√ (r)	√			√
文件用作输出			√ (r)	√	
方法读文件	√ (r)	√	√	√	√
方法写文件			√ (r)	√	
文件在客户端上	√	√	√ (r)	√	√

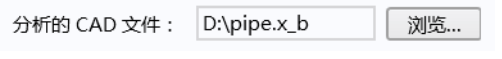
您可以在从**文件**菜单的**首选项**对话框的**文件**页面中设置临时、用户，以及公用文件的路径，如下图所示。



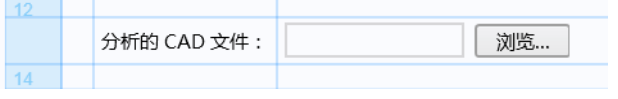
文件导入

使用模型树和文件导入对象导入 CAD

文件导入对象用来显示与输入框相关联的文件浏览器，可以浏览到文件或者输入它的路径和名称。它可以让 App 的用户在运行过程中，当所需的文件尚不能用于 App 时，启用文件导入。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。设想这样一个 App，可以在运行时选定和导入 CAD 文件，如下图所示。



下图显示对应的的**文件导入**对象。

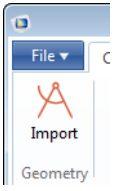


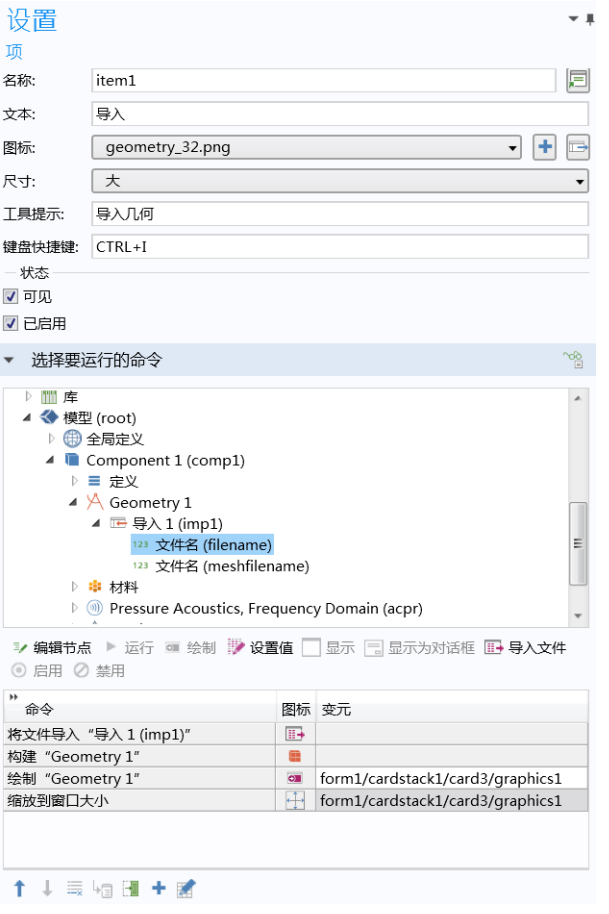
文件导入对象的设置窗口有**文件目标**栏，在此栏中，您可以选择任意允许输入文件名的树节点。如下图所示，选定了**导入**节点的**文件名**。



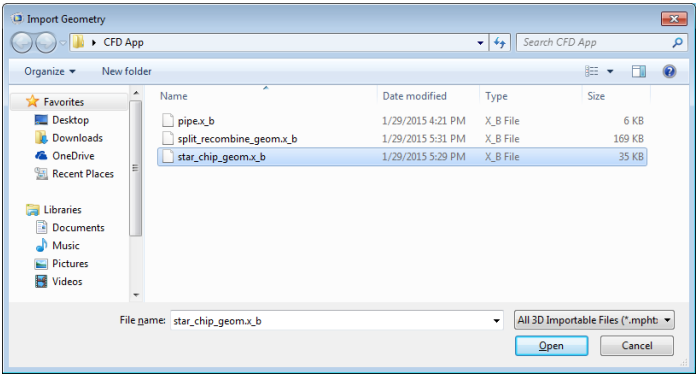
如果您不希望使用**文件导入**对象，可以从按钮、或者菜单项、功能区，或者工具栏等，直接引用一个**文件名**。

下图显示了用于几何导入的功能区项，以及它的**设置**窗口。





在上面的设置窗口中，命令**导入文件到导入 1** 将打开一个文件浏览器，供用户选择文件，如下图所示。



子命令序列构建和绘制几何，使用缩放到窗口大小进行缩放，最终设置字符串变量的值 （本例用来控制卡片堆）。参见第 201 页 “文件导入”，了解更多关于**文件导入**对象的信息。

方法中的文件导入

继续前一节的示例，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";
```

第一行表示使用内建方法 importFile。参见第 274 页 “文件方法”，了解更多方法 importFile 和其他文件处理方法的信息。

文件访问和文件声明

在**文件导入**对象的设置窗口下边，您可以看到使用了哪个文件方案语法来从方法访问导入的文件 （在**访问时使用的命令**旁边）。下图显示了一个用到**文件目**

标和**文件名**的示例。



访问这个文件需要文件方案语法 `upload:///geom1/imp1/filename`。

另一种方法是，您可以使用**声明**节点下的**文件**声明。（但是，**文件**声明主要用于方法代码的文件导入。）在这种情况下，可以在表单对象或方法中使用语法 upload:///file1， upload:///file2 等引用由用户选定的文件。文件名句柄 （file1，file2，等）可以用来引用由用户在运行时选择的实际文件名。参见第 122 页“文件声明”。

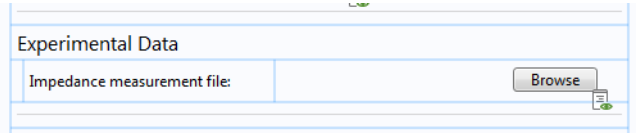
这种语法还可以用于模型开发器节点的很多文件浏览器文本框。下图显示了一个使用几何导入的模型中，用于模型树中**导入**节点的**文件名**框的文件引用。



然而，更快捷的方法是将文件导入对象直接链接到文件名框，如之前第 241 页“使用模型树和文件导入对象进行 CAD 导入”节中所述。

导入实验数据

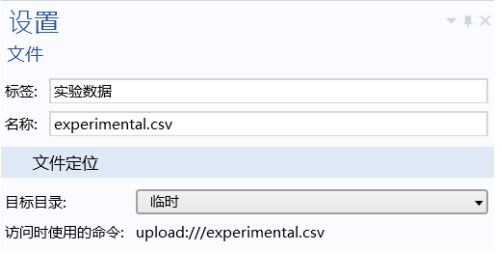
考虑这样一个 App，其中用户在运行时提供一个包含实验数据的文件。下图显示这样一个 App 在栅格布局模式下的文件导入对象。



下图显示对应的文件导入对象的设置窗口，它链接到一个文件声明。



在这个 App 中，文件类型表格指定只允许使用 CSV 文件。下图显示文件声明的设置窗口。

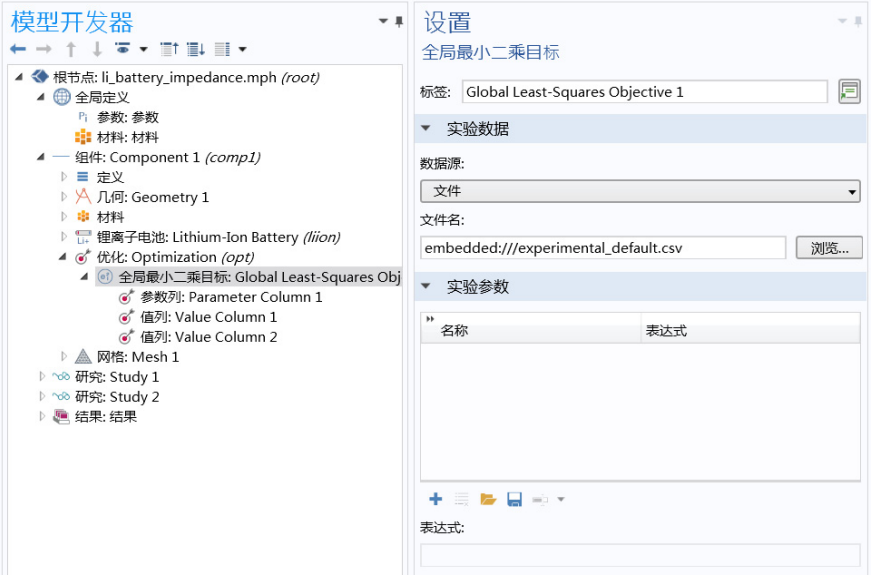


文件声明的作用是导入数据的 “目标”，写为 upload:///experimental.csv。

注意，声明中的文件扩展名 .csv 是可选项，用户在运行时挑选的文件可以是任意名称。例如，运行时挑选的文件名可以是 my_data.csv，但是当在方法代码中引用时，总是只能使用绝对的文件句柄名称 experimental.csv。

为了使得在运行 App 时不需要首先提供实验数据，在 App 中可以内嵌包含实验数据的文件。通过文件方案语法 embedded:/// 来访问这个默认的文件，如下图所示。

在这个使用优化模块的示例中， App 执行最小二乘法来匹配实验数据。



下面的方法处理一个逻辑关系，判断是否存在用户提供的实验数据文件，或者是否使用默认的数据集。

```
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，代码在物理场接口 glsobj1 中使用 `upload:///experimental.csv` 替换 `embedded:///experimental_default.csv`。

使用模型树导出文件

在命令序列中 （例如，按钮），您可以通过运行**导出**或**报告**节点的子节点来导出由内嵌模型生成的数据。

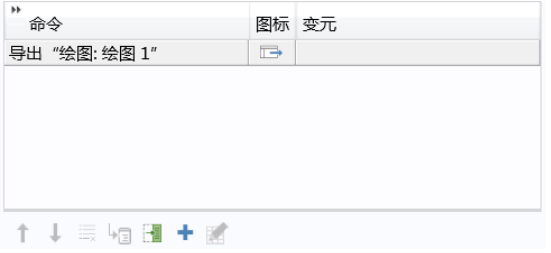
在模型树中，**导出**节点可能包含以下这些类型的可用于文件导出的子节点：

- **数据**
- **绘图**
- **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中都包含一个带有**文件名**框的输出栏。下图显示了一个**导出 > 绘图**节点的**设置**窗口。

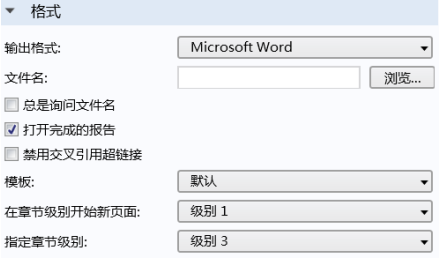


您可以保留**文件名**框为空，如上图所示。在命令序列中 （例如，按钮），您可以在运行时运行对应的导出 > 绘图节点，它将打开一个文件浏览器窗口，供用户选择位置和文件名，如下图所示。



当开发 App 时，您可能需要反复使用模型开发器来检查导出的数据。在这种情况下，您可以使用**文件名**框，并且勾选**总是询问文件名**复选框，这样将保持在运行时打开一个文件浏览器窗口。

与**导出**子节点的情况类似，**报告**子节点同样有**文件名**框和**格式**栏，如下图所示。



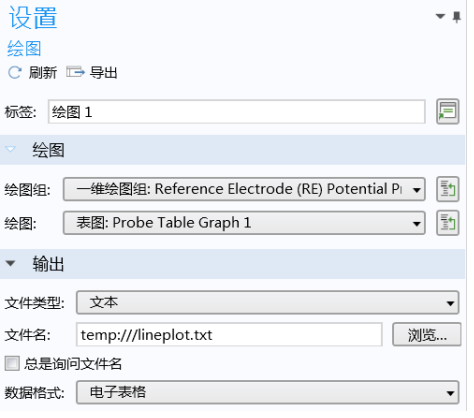
通过运行**报告**子节点，将打开一个文件浏览器窗口，用户可以选择报告的位置和文件名。

如果需要更加精细地控制文件导入和导出，您可以使用文件方案。

文件导出为临时文件

一些 App 可能需要产生临时文件，这可以通过使用 temp:/// 文件方案来实现。临时文件保存在随机的临时路径，对每个 App 线程具有唯一性。当 App 被关闭时，将自动删除这些文件。临时文件可以由命令序列或方法产生，或者当使用 COMSOL Server 时用于保存到客户端的输出文件。

下面的示例显示了用于将绘图数据导出为数值时的**导出** > **绘图**节点的设置窗口。

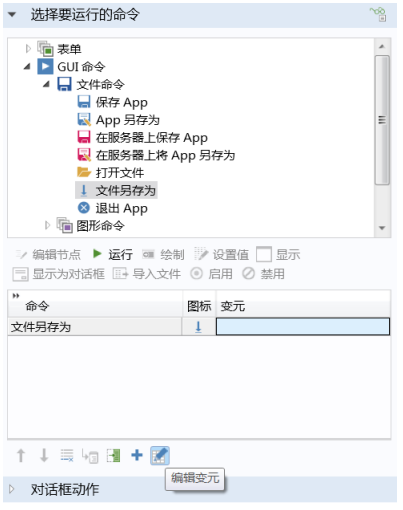


输出栏的**文件名**设置为 temp:///lineplot.txt。
为了在这个示例中将绘图保存到硬盘，创建了一个按钮。在按钮的**设置**窗口，在**选择运行的命令**栏中，首先通过选择上面创建的**导出** > **绘图**节点并单击**运行**，创建了一个输出图形文件。第二步，选择 **GUI 命令** > **文件命令** > **文件另存为**，并再次单击**运行**。

在按钮**设置**的**输出**栏，设置文件名为由**导出绘图**命令创建的临时文件，本例为`temp:///lineplot.txt`。



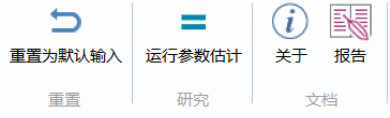
文件另存为命令提供了一个专门的编辑变元对话框，可以方便地访问所有内嵌的文件，以及所有文件方案的快捷方式。



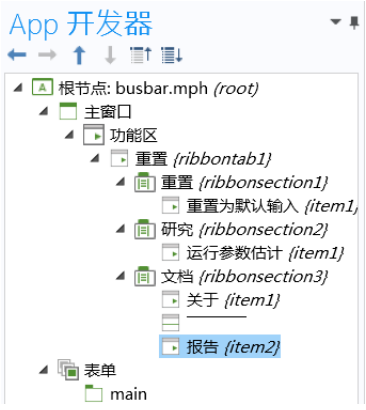
使用底层功能创建报告

本节描述使用底层功能创建报告，参见第 249 页 “文件导出”了解更多直接方法。

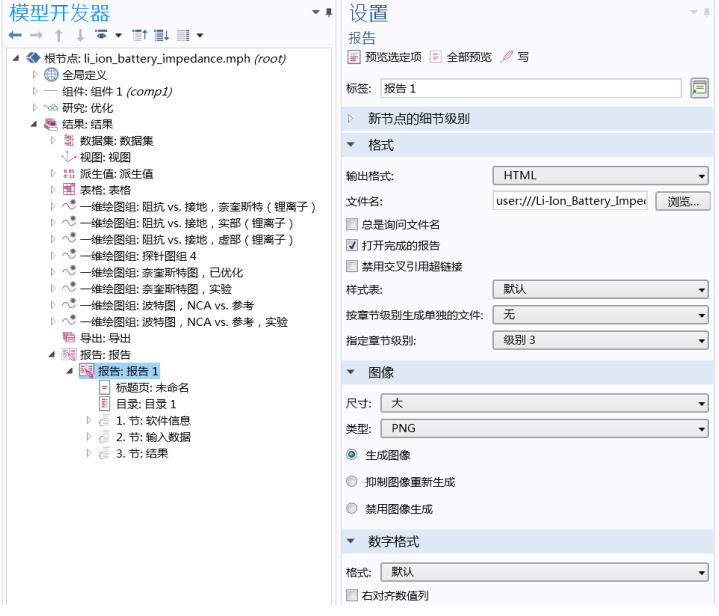
下面的示例显示了一个可以由用户保存 Microsoft® Word® 格式 （.docx）的报告的 App。下图显示这个 App 的功能区选项卡。在这个选项卡中，**文档**栏有一个**报告**按钮。



下图显示了关联的 App 树节点。



下图显示了语法 user:///file 如何用于模型开发器的**报告**节点的**设置**窗口中的**文件名**框。



在这个 App 中，选中了复选框**打开完成的报告**，这意味着当报告创建完成后将打开 Word® 文档。 App 的用户就可以从 Word® 文件的菜单来保存报告。

在这个示例中，可以通过类似的方式使用文件方案 common:///。 user 和 common 文件方案主要用于一个 App 会反复使用相同文件的情况。

下图显示了报告**功能区**项的**设置**窗口。



方法 b_report 包含了下列代码：

```
if(length(information_card)>0) {
   alert("New input data. Compute to update results first.");
}
else {
   model.result().report("rpt1").run();
}
```

还可以在方法中直接使用文件方案语法。下面的代码来自于一个导出 HTML 报告的方法。

```
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

下表列出了 App 开发器中可以使用的键盘快捷键。

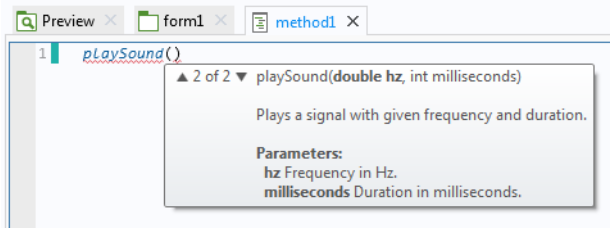
快捷键	操作	APP 开发器	表单编辑器	方法编辑器
Ctrl+A	选择全部	√	√	√
Ctrl+D	取消选择全部		√	
Ctrl+C	复制	√	√	√
Ctrl+V	粘贴		√	√
Del	删除	√	√	√
Ctrl+N	创建一个新 App	√	√	√
Ctrl+S	保存 App	√	√	√
Ctrl+F8	测试 App	√	√	√
Alt+Click	编辑确定的表单对象		√	
Ctrl+Pause	停止运行方法	√		
Ctrl+Shift+F8	应用修改	√	√	√
Ctrl+R	记录代码			√
F11	切换到节点			√
F12	提取变量			√
F1	显示帮助	√	√	√
F2	重命名适用节点	√		
F3	禁用适用节点	√		
F4	启用适用节点	√		
Ctrl+Up arrow	上移适用节点	√		
Ctrl+Down arrow	下移适用节点	√		
Ctrl+Z	复原	√	√	√
Ctrl+Y	重做 （在 Mac 上为 Control+Shift+Z）	√	√	√
F5	继续 （调试器中）			√
F6	步进 （调试器中）			√
F7	步进进入 （调试器中）			√
F8	检查语法			√

快捷键	操作	APP 开发器	表单编辑器	方法编辑器
Ctrl+F	在方法中查找和替换			√
Ctrl+Space	自动补完方法代码			√
Ctrl+U	将选定代码改为小写			√
Ctrl+Shift+U	将选定代码改为大写			√
Ctrl+B	在选定行切换中断点			√
Ctrl+M	在匹配的圆括号、方括号或花括号之间切换			√
Ctrl+Shift+M	选择匹配的圆括号、方括号或花括号之间的所有字符			√
Ctrl+Scroll wheel up	在方法代码窗口放大			√
Ctrl+Scroll wheel down	在方法代码窗口缩小			√
Ctrl+All arrow keys	微调选定表单对象的位置		√	
All arrow keys	微调选定表单对象的位置		√	
Ctrl+Alt+A	前往 App 开发器窗口		√	√
Ctrl+Alt+M	前往模型开发器	√	√	
Ctrl+Alt+Left-Click	创建一个局部方法		√	
Alt+F4	关闭窗口	√	√	√
Ctrl+F4	关闭文档		√	√
Ctrl+Shift+F4	关闭所有文档		√	√

附录 E — 内建方法库

本附录列出除了在模型对象和 App 对象上操作的方法之外的所有方法编辑器中可以使用的内建方法。如果要了解使用内建方法的详细信息以及所使用的语法的完整信息，请参见 *Application Programming Guide* 和 *Programming Reference Manual*。

作为学习这些方法的语法的一种替代方法，您可以使用代码完备，即输入方法的名称，然后使用 Ctrl+Space。此时将打开一个窗口，其中包含有关语法和方法的用法说明的信息。



模型实用程序方法

使用模型实用程序方法可以将 MPH 文件的模型对象零件加载到方法以进行进一步处理。

名称	描述
`createModel`	创建一个带有给定标记的新模型。
`removeModel`	删除模型。不能删除嵌入的模型。
`modelTags`	返回所有已加载模型 （包括嵌入模型）的模型标记数组。
`uniqueModeltag`	返回不在使用中的模型标记。
`getModel`	返回带有指定标记的模型。
`loadModel`	从文件加载带有指定标记的模型。
`loadProtectedModel`	从文件加载带有指定标记的受密码保护的模型。
`loadRecoveryModel`	从恢复目录 / 文件夹结构加载模型。
`saveModel`	将模型保存到文件。文件名可以是文件系统路径，如果安全设置允许，也可以是服务器文件路径。

文件方法

名称	描述
readFile	返回给定文件中的内容为字符串。
openFileStreamReader	返回可以用来从给定文件名逐行或逐字读取行的 CsReader 对象。
openBinaryFileStreamReader	返回可以用来从给定文件字节逐字节读取的 CsBinaryReader 对象。
readMatrixFromFile	读取给定文件内容到双精度矩阵。文件具有模型树导出节点可用的相同电子数据表类型的格式。
readStringMatrixFromFile	读取给定文件的内容到字符串矩阵。文件具有模型树导出节点可用的相同电子数据表类型的格式。
readCSVFile	读取用逗号隔开值的文件（CSV 文件）到字符串矩阵。期望文件使用 CSV 的 RFC 4180 格式。
writeFile	写数组数据到给定文件。如果使用电子数据表格式，则数据可被 readMatrixFromFile 或 readStringMatrixFromFile 读取。
openFileStreamWriter	返回可写到给定文件的 CsWriter 对象。
openBinaryFileStreamWriter	返回可以用来逐字节写到给定文件的 CsBinaryWriter 对象。
writeCSVFile	写给定双精度数或字符串数组到 CSV 文件，CSV 使用 RFC 4180。
exists	测试是否存在给定名称的文件。
deleteFile	删除具有给定名称的文件（如果存在）。文件在服务器上删除。
copyFile	在服务器上复制文件，源名称和目标名称都可以使用文件系统路径。
importFile	显示一个文件浏览器对话框，然后上传选定的文件到给定名称的文件声明。也可以上传选定的文件到给定模型对象实体的文件名文本框。
writeExcelFile	从 Excel 文件的指定工作簿的指定单元格开始写给定字符串数组数据。
readExcelFile	从 Excel 文件的指定工作簿，指定单元格开始读入二维字符串数组。
getFilePath	返回对应于确定的文件系统路径的服务器代理文件的绝对服务器文件路径，或者，如果不存在给定路径的服务器代理文件，返回 null。 例如，这个方法可以用来传递使用 temp:/// scheme 的文件的路径到外部代码或 App。

名称	描述
`getClientFileName`	返回客户端文件系统中上传的文件的原始名称 （或者，如果没有与给定文件系统路径匹配的上传文件，返回 null）。 这个方法只用于提供用户界面反馈。例如，获取哪个上传文件被使用的信息。并不保证原始文件仍然存在于客户端或者当前客户端是否为原始客户端。
`getClientFilePath`	返回客户端文件系统中上传的文件的原始路径 （或者，如果没有与给定文件系统路径匹配的上传文件，返回 null）。 这个方法只用于提供用户界面反馈。例如，获取哪个上传文件被使用的信息。并不保证原始文件仍然存在于客户端或者当前客户端是否为原始客户端。

操作系统方法

名称	描述
executeOSCommand	执行指定命令（完整路径）和参数的 OS 命令。当可行时，在服务器上运行命令。
fileOpen	在客户端上使用关联的程序打开文件。另请参见“文件方法”。
getUser	返回运行 App 的用户的用户名，如果不是通过 COMSOL Server 来运行 App，则返回首选项设置中通用 > 用户名 > 名称的值。
openURL	在客户端上的默认浏览器中打开 URL。
playSound	在客户端上播放声音。

电子邮件方法

名称	描述
emailFromAddress	返回来自 COMSOL Server 或首选项的电子邮件发信地址。
sendEmail	发送电子邮件给指定收信人，指定主题、正文，没有或附带多个从内嵌模型的报告、导出和表格节点创建的附件。
userEmailAddress	返回对应于当前登录用户的用户电子邮件地址，或者，如果用户还没有配置电子邮件地址，返回空字符串。

电子邮件类方法

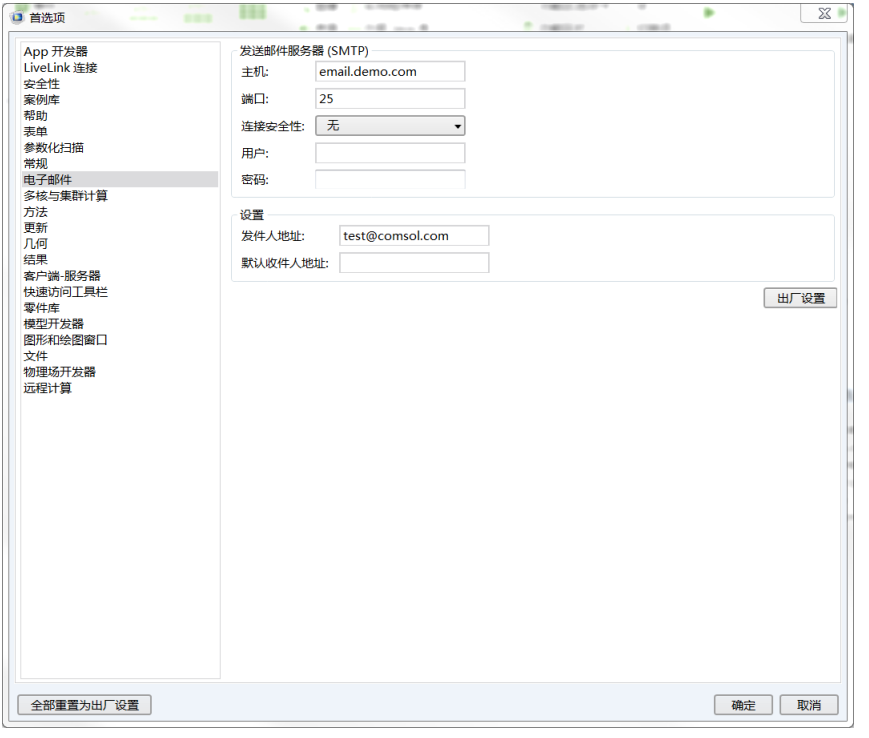
类 EmailMessage 可以用来创建定制的电子邮件信息。

名称	描述
EmailMessage	创建一个新的 EmailMessage 对象。
EmailMessage.setServer	设置电子邮件 (SMTP) 服务器主机和端口以用于该电子邮件信息。
EmailMessage.setUser	设置用户名和密码以用于电子邮件 (SMTP) 服务器验证。此方法必须在 setServer 方法后调用。
EmailMessage.setSecurity	设置电子邮件 (SMTP) 服务器通信的连接安全类型。
EmailMessage.setFrom	设置发信地址。
EmailMessage.setTo	设置收信地址。

名称	描述
`EmailMessage.setCc`	设置抄送地址。
`EmailMessage.setBcc`	设置密送地址。
`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。一封电子邮件可以包含文本和 HTML 主体。
`EmailMessage.setBodyHtml`	将电子邮件正文设为 HTML 文本。一封电子邮件可以包含文本和 HTML 主体。
`EmailMessage.attachFile`	从文件添加附件。附件 MIME 类型由文件扩展名决定。
`EmailMessage.attachFile`	从文件添加指定 MIME 类型的附件。
`EmailMessage.attachFromModel`	添加从模型中的报告、导出或表格特征创建的附件。
`EmailMessage.attachText`	添加指定子 MIME 类型的文本附件，例如纯文本或 HTML。
`EmailMessage.attachBinary`	从指定 MIME 类型的字节数组添加附件。
`EmailMessage.send`	发送电子邮件到电子邮件 (SMTP) 服务器。一个电子邮件主题只能发送一次。

电子邮件首选项

如果需要设置发送邮件的服务器 （SMTP）的首选项，请打开**首选项**对话框的**电子邮件**页面，如下图所示。



COMSOL Server 提供类似的电子邮件首选项设置。

GUI- 相关的方法

名称	描述
直接调用方法	使用名称从方法列表中调用方法，例如，`method1()`，`method2()`。
`callMethod`	另一种从方法列表中调用方法的方式，内部使用，以防名称冲突。
`useGraphics`	在第二个变元给定的名称或名称路径的图形表单对象中绘制给定实体（绘图组、几何、网格，或者显式选择）。
`useForm`	在当前主窗口显示给定名称的表单，相当于表单对象的 `use` 方法，见下文。
`closeDialog`	关闭给定名称的显示为对话框的表单。
`dialog`	以对话框形式显示给定名称的表单，相当于表单对象的 `dialog` 方法，见下文。
`alert`	停止执行，并显示给定文本的警告信息。
`alert`	停止执行，并显示给定文本和标题的警告信息。
`confirm`	停止执行并显示给定文本和标题的确认对话框。并且还显示两个或三个按钮，例如，“是”、“否”和“取消”。
`error`	停止执行并打开带给定信息的错误对话框。
`request`	停止执行并显示一个带需要用户输入的文本框的对话框。
`message`	如果在 App 中可以使用信息日志，则发送信息到信息日志。
`evaluateToResultsTable`	在第二个变元的名称或名称路径给定的表格对象中计算给定的派生值实体，然后作为派生值计算的默认目标。如果第三个变元为 true，则在添加新数据前清除表格，否则在后面增加数据。
`evaluateToDoubleArray2D`	计算给定的派生值实体，并返回生成为双精度数矩阵的实数表的非参数列部分。将考虑数值特征中的所有设置，但将忽略连接到数值特征的当前表中的设置。
`evaluateToIntegerArray2D`	计算给定的派生值实体，并返回生成为整数矩阵的实数表的非参数列部分。将考虑数值特征中的所有设置，但将忽略连接到数值特征的当前表中的设置。

名称	描述
evaluateToStringArray2D	计算给定的派生值实体，并返回生成为字符串矩阵的潜在复杂值表的非参数列部分。将考虑数值特征中的所有设置，但将忽略连接到数值特征的当前表中的设置。
useResultsTable	显示 resultsTable 表单对象中的 tableFeature 值。
getChoiceList	返回 ChoiceList 类型的对象，代表声明分支下的选择列表节点。 ChoiceList 类型有关联方法可以轻松地更改值和显示名称，参见 *Application Programming Guide* 第 305 页的“选择列表方法”。
setFormObjectEnabled	设置由名称或名称路径指定的表单对象的启用状态。
setFormObjectVisible	设置由名称或名称路径指定的表单对象的可见状态。
setFormObjectText	设置由第二个变元的名称或名称路径指定的表单对象的文本。如果不能设置指定表单对象的文本，则抛出错误。
setFormObjectEditable	设置由名称或名称路径指定的表单对象的可编辑状态。此功能只能用于文本框对象。
setMenuBarItemEnabled	设置由第一个变元的名称或名称路径 （来自菜单栏）指定的菜单栏项目的启用状态。
setMainToolbarItemEnabled	设置由第一个变元的名称或名称路径 （来自主工具栏）指定的主工具栏项目的启用状态。
setFileMenuItemEnabled	设置由第一个变元的名称或名称路径 （来自文件菜单）指定的文件菜单项目的启用状态。
setRibbonItemEnabled	设置由第一个变元的名称或名称路径 （来自主窗口）指定的功能区项目的启用状态。
setToolbarItemEnabled	设置由第一个变元的名称或名称路径指定的工具栏表单对象项目的启用状态。
useView	将视图应用到由第二个变元的名称或名称路径给定的图形内容。
resetView	将视图重置为由第二个变元的名称或名称路径给定的图形内容中的初始状态。
getView	返回当前由第二个变元的名称或名称路径给定的图形内容使用的视图。
setWebPageSource	设置由第一个变元的名称或名称路径指定的表单对象的源。

名称	描述
getScreenHeight	返回客户端系统上主屏幕的像素高度，如果使用的是网页客户端，则返回浏览器窗口的像素高度。
getScreenWidth	返回客户端系统上主屏幕的像素宽度，如果使用的是网页客户端，则返回浏览器窗口的像素宽度。

GUI 命令方法

名称	描述
clearAllMeshes	清除所有网格。
clearAllSolutions	清除所有解。
exit	退出 App。
fileOpen	在客户端上打开一个包含关联程序的文件。
fileSaveAs	下载文件到客户端。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	在另一名称下保存 App。（或者作为 MPH 文件保存。）
scenelight	切换给定图形对象的布景灯。
transparency	切换给定图形对象的透明度。
zoomExtents	在给定图形对象中使整个模型可见。

调试方法

名称	描述
debugLog	打印输入变元的值到调试日志窗口。输入变元可以是字符串、双精度数、整数或布尔变量的标量、一维数组或二维数组。

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 （原生）库的接口。外部类使用 Java Native Interface (JNI) 框架。请参见 *Application Programming Guide* 了解更多信息。

外部方法返回的方法

外部方法返回带有下列方法的外部类型的对象：

名称	描述
`invoke`	根据提供的变元调用库中的指定名称的原生方法。
`invokeWideString`	根据提供的变元调用库中的指定名称的原生方法。
`close`	释放库和自由资源。如果您不调用这个方法，当不再需要外部库时，它将被自动调用。

进度方法

名称	描述
`setProgressInterval`	设置用于顶级进度的进度间隔并显示该级别的消息。 调用这个方法将隐式地重置之前调用 `setProgress()` 所设置的所有手动进度。
`setProgress`	设置用户控制的进度水平的值。
`resetProgress`	删除所有进度水平，将进度重置为 0，消息重置为空字符串。
`showIndeterminateProgress`	通过不确定的进度条、给定的消息和可选的取消按钮显示进度对话框。
`showProgress`	显示一个带可选取消按钮、可选模型进度以及一级或两级进度信息的进度对话框。
`closeProgress`	关闭当前显示的进度对话框。
`startProgress`	将给定进度条表单对象名称的值重置为 0。
`setProgressBar`	将给定进度条表单对象名称的值设为 0 –100，以及关联的进度消息。

日期和时间方法

名称	描述
currentDate	以字符串形式返回当前日期 （格式根据服务器默认而定）。
currentTime	以字符串形式返回当前时间 （不包含日期，格式根据服务器默认而定）。
formattedTime	使用给定格式返回格式化的时间。格式可以是时间单位，也可以是描述较长格式的文本。
sleep	在指定的毫秒期间睡眠。
timeStamp	从 UTC 1970 年 1 月 1 日午夜到当前时间的毫秒数。
getExpectedComputationTime	返回描述 App 近似计算时间的字符串。字符串可通过 setExpectedComputationTime 方法更改。
setLastComputationTime	设置上一次计算时间，覆盖自动生成的时间。 您可以使用 timeStamp 方法来记录时间差，然后设置以 ms （长整数）位单位的测量时间。
getLastComputationTime	以给定格式返回上一次计算时间。格式可以是时间单位，或者描述为更长格式的文本。这是本地化格式，输出将翻译为当前的语言设置。

许可证方法

名称	描述
`getLicenseNumber`	为当前会话返回包含许可证编号的字符串。例如：`licensenumber=getLicenseNumber()`

转换方法

名称	描述
`toBoolean`	转换字符串和字符串数组为布尔变量。（‘true’ 返回 true，其他所有字符串返回 false）。
`toDouble`	转换字符串和字符串数组为双精度数。
`toInt`	转换字符串和字符串数组为整数。
`toString`	转换布尔变量、整型和双精度 （包括数组）为字符串。

数组方法

名称	描述
`getColumn`	返回二维数组 （矩阵）中指定列的字符串、双精度数、整数或布尔变量数组。例如，从文件读取值且只需要显示表格中的某些列时，该方法非常有用。
`getSubMatrix`	返回输入矩阵的矩形子矩阵，可用于字符串、双精度数、整数或布尔变量二维数组。
`insert`	在数组中插入一个或多个单元，并返回扩展后的数组。可用于字符串、双精度数、整数或布尔变量数组。
`append`	在数组后添加一个或多个单元，并返回扩展后的数组，可用于字符串、双精度数、整数或布尔变量数组。
`remove`	从数组中移除一个或多个单元，并返回缩减后的数组，可用于字符串、双精度数、整数或布尔变量数组。
`insertRow`	在矩形二维数组中插入一行或多行，并返回扩展后的数组，可用于字符串、双精度数、整数或布尔变量数组。
`appendRow`	在矩形二维数组后添加一行或多行，并返回扩展后的数组，可用于字符串、双精度数、整数或布尔变量数组。
`removeRow`	从二维数组中移除一行或多行，并返回缩减后的数组，可用于字符串、双精度数、整数或布尔变量数组。
`insertColumn`	在矩形二维数组中添加一列或多列，并返回扩展后的数组，可用于字符串、双精度数、整数或布尔变量数组。

名称	描述
`appendColumn`	在矩形二维数组后添加一列或多列，并返回扩展后的数组，可用于字符串、双精度数、整数或布尔变量数组。
`removeColumn`	从矩形二维数组中移除一列或多列，并返回较小的数组，可用于字符串、双精度数、整数或布尔变量数组。
`matrixSize`	以长度为 2 的整数数组形式返回矩阵的行数和列数，可用于字符串、双精度数、整数或布尔变量数组。

字符串方法

名称	描述
concat	使用给定分隔符将给定字符串数组或矩阵连接到一个字符串。
contains	如果给定字符串数组包含给定字符串，返回 true。
find	返回字符串数组中出现某个字符串的所有索引的数组。
findIn	返回在字符串数组中第一次出现某个字符串的索引，或字符串中第一次出现某个子字符串的索引。
length	返回字符串的长度。
replace	返回字符串被另一个字符串替换后的字符串。
split	通过以给定分隔符拆分给定字符串返回字符串数组。
substring	返回从给定位置开始的给定长度的子字符串。
unique	返回给定字符串数组中具有唯一值的字符串数组。

收集方法

名称	描述
copy	返回给定数组或矩阵的副本，可用于字符串、双精度数、整数或布尔变量数组。
equals	如果给定数组中的所有单元都相等且他们具有想用的单元数量，返回 true。可用于字符串、双精度数、整数或布尔变量数组，对于双精度数，使用相对容差进行比较。
sort	对给定数组进行排序。注：数组在原位排序。可用于字符串、双精度数或整数数组。
merge	返回合并给定数组中所有单元得到的数组，可用于字符串、双精度数或整数数组。

With、 Get 和 Set 方法

名称	描述
with	用于使代码更加紧凑。
endwith	with 语句的结尾。
set	设置布尔、整数、双精度数或字符串属性值。可用于标量、数组或矩阵属性。
setIndex	在给定索引处设置矩阵或矢量的字符串、双精度数或整数属性值。
getIntArray	获取整数矢量属性。

名称	描述
getIntMatrix	获取整数矩阵属性。
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	获取双精度数属性。
getString	获取字符串标量、矢量或矩阵属性。
getDoubleArray	获取双精度数矢量属性或参数。
getDoubleMatrix	获取双精度数矩阵属性或参数。
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	以字符串矩阵形式返回值。
getInt	获取整数属性。
get	返回变量表达式。
descr	返回变量描述。

附录 F— 开发 App 的准则

通用

• 在生成的报告文件中，包含输入数据和对应的输出数据。

• 直观易懂。如果有必要，提供帮助、提示，以及文档。

• 简单：“Safe I/O”，“Reset to default data”等。

• 保存模型的同时保存它的预览图片。

• 包含 App 描述 （会显示在 COMSOL Server 库中）。

• 在它将会被用到的计算机平台上进行测试。

• 风格简约。从开发者的视角出发，这样显得更加便于确保逻辑关系、组织、调试、维护，以及进一步开发 App。从使用者的角度，这样更加便于使用 App。风格简约需要在开发时更加仔细，但是后续可以花更少的精力来维护，并且适用于尽可能多的用户。

• 如果文件大小在一定范围内，将它们内嵌到模型的库。

• 显示可能的计算时间，以及完成计算后实际的计算时间。

• 当取消计算时，之前计算的输出数据应当被清除。

• 如果有必要，可以密码保护 （记住：如果你自己忘了密码，没人能够帮你。）

方法

• 不要创建不必要的方法。

方法越少，当查找一些东西时需要浏览的方法列表就越小。更少的方法通常意味着需要担心的代码行越少。

- 如果你编写的一些方法做的本质事务差不多，考虑将它们合并到一个方法，并通过输入变元来处理不同的情况。
- 如果只在某个地方使用就不要创建方法，将相关代码插入到这个地方就可以了。

• 如果只在一个表单中使用、被一个表单事件或一个表单对象事件触发，就创建局部方法。

• 给方法描述性的命名，对于类似的方法给予类似名称，以便于当通过字母排序时可以形成编组。你只需要记住少量的信息，并且可以很方便地找到它们。长文件名比短文件名更容易理解。

方法命令示例：

- 没有输出的方法以 p 开始 （p 表示过程）。
- 有输出的方法以 f 开始 （f 表示函数）。
- 所有的菜单项方法以 m 开始 （m 表示菜单）。
- 将经常查看的方法以 a 开始，这样它将显示在列表的前面。
- 所有的绘图方法加上 Plot，菜单项绘图方法 mPlotMesh、 mPlotResults。

• 以上这些要点同样可用于方法代码。风格简约，用到的代码行和变量越少越好。使用变量描述性名称。使用长名称代替较难理解的短名称。优化代码，提高运行效率。

• 上面的要点同样也可用于描述。使用好名称，而不是万不得已的名称。在用到它们的地方声明变量 （表单和方法，或者模型）。

表单

• 不要创建不必要的表单。

• 给表单描述性的名称，与命名方法的原因类似。

• 灵活运用各种不同类型的表单对象，一些表单适用于这类事务，另一些表单适用于其他事务。

• 尽量不要插入过多的表单对象。太多的输入数据选项将使得 App 难以使用。太多的输出数据则使得它很难找到重要信息。

• 插入文本框供用户保留备注，当保存 App 时可以与用户输入和输出数据同时记录下来。

• 考虑插入一个带方法的按钮，一键单击回复到默认值。

• • 应用 “安全的输入 / 输出”：

- 对于输入框，警告用户输入数据超出范围。你可以通过输入框的数据改变时运行事件触发警告，或者在表单对象设置窗口设置限制 （设置强制限制）。在生成警告的方法中，您可能只是警告用户，然后如果用户选择无论如何都运行时使用输入数据。
- 对于输出框，给定有意义的精度。如果当前结果与当前输入数据无关，显示它。如果计算失败，显示它。

• 包含工具提示、帮助、文档、提示，以及完整的报告。

• 提供关于使用默认输入数据在典型计算机上运行仿真需要花费的时间的用户信息。它可以是数秒、数小时，甚至数天，依赖于不同 App，这样一来，用户在单击计算按钮前就知道相关信息。考虑当计算完成时播放提示声音。用户可能去做别的事情，而不是坐等结果 （如果计算时间确实很长，当计算完成时，发送一个电子邮件消息，附上报告会是一个较好的方案）。

• 花一些时间来整理表单的布局。外观好的表单理容易使用，也更有兴趣使用。

• 考虑为按钮和菜单项设置键盘快捷键。

附录 G—App 库示例

在 App 库中，我们可以找到演示 App 开发器的各种功能的示例 App，它们集中在各个附加模块的 Applications 文件夹下。您可以编辑这些演示性 App，使用它们作为您自己的 App 设计的出发点。每个演示性 App 都包含文档 （PDF）描述 App，以及生成报告的选项。

下表列出了一些可供参考的 App 示例的列表，按照它们在 App 库树状结构中的顺序整理。

名称	案例库
受移动载荷的梁	COMSOL Multiphysics
螺旋式静态混合器	COMSOL Multiphysics
传输线计算器	COMSOL Multiphysics
管式反应器	COMSOL Multiphysics
音叉	COMSOL Multiphysics
等效非线性磁曲线计算器	AC/DC Module
钢棒感应加热	AC/DC Module
矿藏磁勘探	AC/DC Module
触摸屏仿真器	AC/DC Module
消音器设计器	Acoustics Module
声反射分析器	Acoustics Module
家庭住宅声学分析器	Acoustics Module
凤琴管设计	Acoustics Module, Pipe Flow Module[12]
小型音乐厅分析器	Acoustics Module
循环伏安法	Electrochemistry Module, Electrodeposition Module, Batteries & Fuel Cells Module, Corrosion Module
电化学阻抗谱	Electrochemistry Module, Electrodeposition Module, Batteries & Fuel Cells Module, Corrosion Module
锂电子电池的阻抗	Batteries & Fuel Cells Module[1]
喷墨打印机	CFD Module, Microfluidics Module

名称	案例库
NACA 机翼优化	CFD Module[8]
水处理池	CFD Module
生物传感器设计	Chemical Reaction Engineering Module
液相色谱分析	Chemical Reaction Engineering Module
膜透析	Chemical Reaction Engineering Module
船壳阴极保护	Corrosion Module
框架疲劳寿命	Fatigue Module[11]
参数化混凝土梁	Geomechanics Module[11]
同心管换热器	Heat Transfer Module[2]
翅片散热器	Heat Transfer Module
周期性微结构的等效属性	Heat Transfer Module
激光闪射法	Heat Transfer Module
散热器强制空气冷却	Heat Transfer Module
管线感应加热器	Heat Transfer Module[9]
阳伞和太阳辐射	Heat Transfer Module
热电冷却器	Heat Transfer Module
吸湿膨胀引起的 MEMS 压力传感器漂移	MEMS Module, Structural Mechanics Module
微电阻梁	MEMS Module
红血球分离	Microfluidics Module, Particle Tracing Module[3]
搅拌器	Mixer Module[4]
电荷交换单元仿真器	Molecular Flow Module, Particle Tracing Module[10]
离子注入机计算器	Molecular Flow Module
离心调节器仿真器	Multibody Dynamics Module[11]
车载起重机分析器	Multibody Dynamics Module[5,11]
压力容器应力分析	Nonlinear Structural Mechanics Module[11]
静态层流颗粒混合器设计器	Particle Tracing Module
气室仿真器	Pipe Flow Module
地热泵	Pipe Flow Module
容性耦合等离子体反应器	Plasma Module
分布式布拉格反射滤波器	Ray Optics Module

名称	案例库
圆形波纹喇叭状天线	RF Module
频率选择面仿真器	RF Module
微带贴片天线阵列合成器	RF Module
表面等离激元线光栅	RF Module, Wave Optics Module
可调波长 LED	Semiconductor Module
梁横截面计算器	Structural Mechanics Module[6]
车架分析器	Structural Mechanics Module[13]
过盈配合计算器	Structural Mechanics Module
桁架桥设计器	Structural Mechanics Module
桁架塔屈曲	Structural Mechanics Module
粘弹性结构阻尼器	Structural Mechanics Module
光纤模拟器	Wave Optics Module
表面等离激元线光栅	RF Module, Wave Optics Module[7]

[1] 需要 “电池与燃料电池模块” 和 “优化模块”。

[2] 通过 “CFD 模块”、“传热模块”、“微流体模块” 或 “等离子体模块” 运行。

[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” 和以下模块之一：“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] 可使用此 App 的扩展版本，此版本还需要 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”。

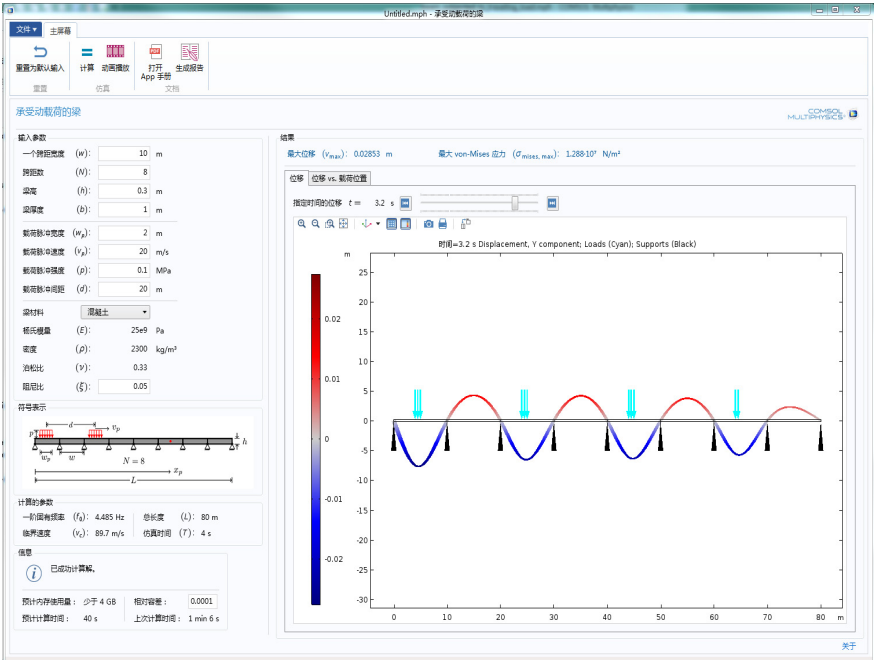
[13] 为了获取完整的参数化功能，此 App 需要 LiveLink™ for SOLIDWORKS®。

下面的章节着重说明这些演示性 App 的特点，这些特点演示一些重要的 App 开发器特征，包括使用动画、 email、优化、参数估计、表格、以及导入实验数据。

受移动载荷的梁

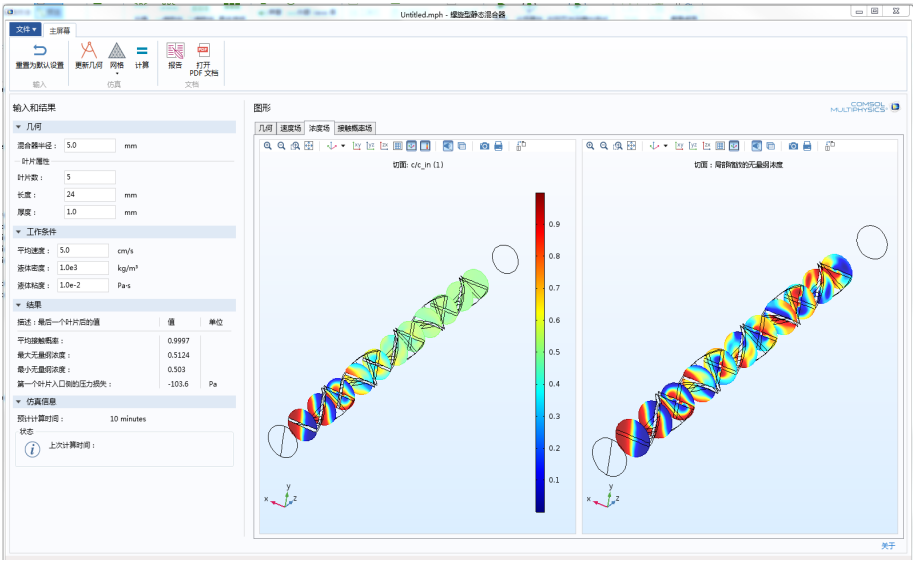
这个 App 仿真放置在一些均匀支撑上的梁受到沿着梁运动的载荷时的瞬态响应。目标是分析当汽车通过桥梁时的响应。它观察到，对于给定几何和材料属性的桥梁，确定的汽车速度会导致桥梁共振，产生很大幅度的振动。在 App 中，假设满足二维平面应力近似，可以简化为梁，它是由混凝土构成。

App 演示了动画和滑块的用法。第一个滑块显示随载荷位置的变形，第二个滑动显示位移随时间的演化。这个 App 不需要任何附加模块。



螺旋式静态混合器

这个 App 的目的是演示几何零件和参数化几何的用法。此外， App 可以用来估计在一个包含一到五个螺旋刀片的体系中的混合程度，通常这种装置用来混合聚合反应中的单分子和引发剂。 App 假定为牛顿流体，这是如果聚合反应对混合器本身的影响可以忽略情况下很好的假设。在 App 中还演示了**平铺或者选项卡**类型的表单集合的用法。不需要任何附加模块的许可证。



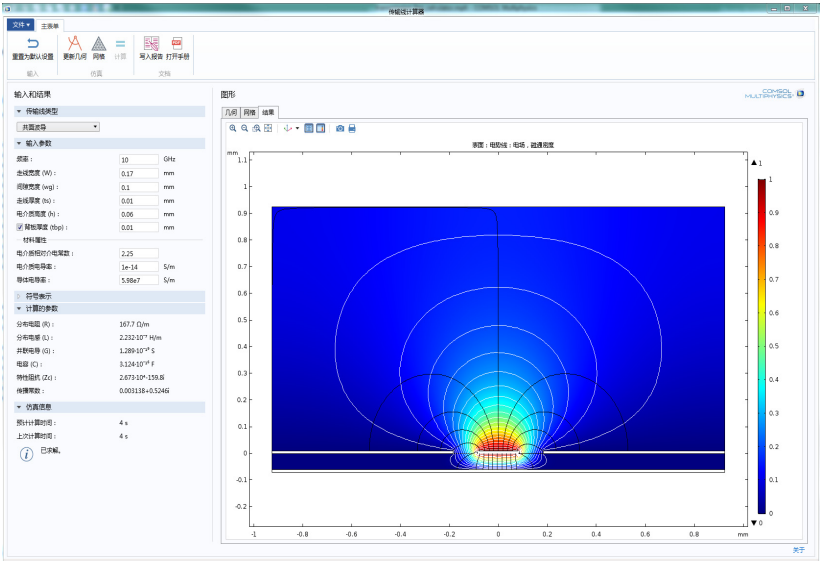
传输线计算器

传输线用于传输射频范围内的交变电流和电压。传输线有多种形式，其中很多适应于易加工和用于印刷电路板设计。它们是很多现代化电气设备中的关键元素，常用于携带信息，具有低损耗和低失真特性，可以在设备内和设备之间传输信号。

这个 App 提供了预定义的用户界面，计算一些常用传输线类型在不同参数化截面情况下的传输线参数 R、 L、 G 和 C，以及 γ 和 Z_0：

- 同轴线
- 双绞线
- 微带线
- 共面波导 （CPW）

界面中还提供了绘制几何、网格、电势、电场线，以及磁通量线等。不需要任何附加模块的许可证。



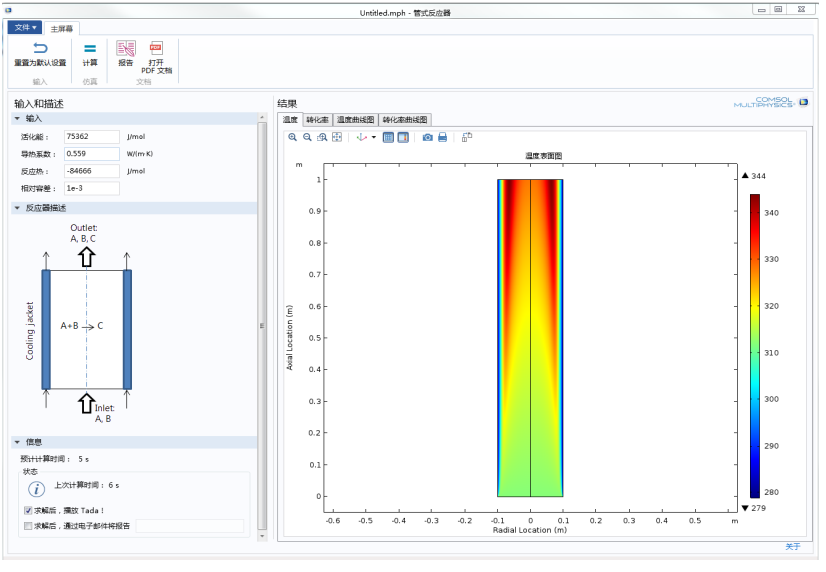
管式反应器

有了这个 App，化工系的学生可以模拟一个非理想管式反应器，包括径向和轴向温度和组成变化，并研究不同操作条件产生的影响。这个 App 描述的工艺是环氧丙烷与水进行放热反应生成丙二醇。

这个 App 还演示了教师如何开发专业的接口来加深学生对这种反应的映象。模型和练习出自于 Scott Fogler 的 Elements of Chemical Reaction Engineering。

数学模型包含轴对称坐标系下的能量守恒和物料守恒。学生可以改变反应器中的反应活化能、热导率，以及反应热。结果给出轴向和径向的转化率及温度分布。对于某些数据，仿真结果将不明确，这意味着对模型结果的解释还可以转变为解决问题的练习。

注意，当您勾选电子邮件复选框并输入电子邮件地址时，还可以让这个 App 在完成计算后发送电子邮件。它将发送包含设置和计算结果的报告。这个功能可以用于学生将结果发送给导师。对于需要较长计算时间的仿真，这个功能将非常实用。例如，您可以启动一个仿真，然后离开办公室或实验室，然后当完成计算后收到完整报告，您可以在路上，或者任何一个可以接收电子邮件的地方。这个 App 同样不需要任何附加模块。



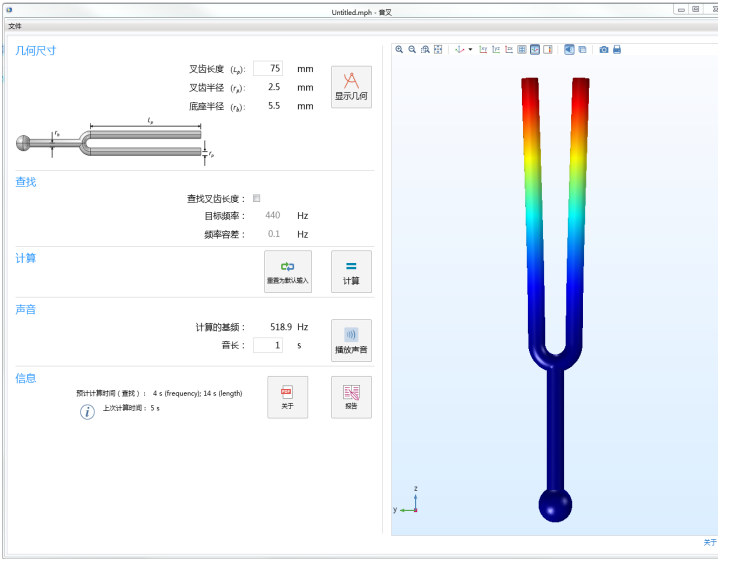
音叉

这个 App 计算用户定义叉齿长度的音叉的谐振频率。或者您可以指定需要的目标频率，让 App 找出对应的叉齿长度。其中叉齿和手柄的半径来自于商用音叉。

内嵌在这个 App 中的模型通过 COMSOL Multiphysics 模块的固体力学接口定义，不需要任何附加模块。叉齿长度的搜索算法采用割线法。

在计算完成后，内建方法 playSound 用来播放一个符合谐振频率的正弦波声音。

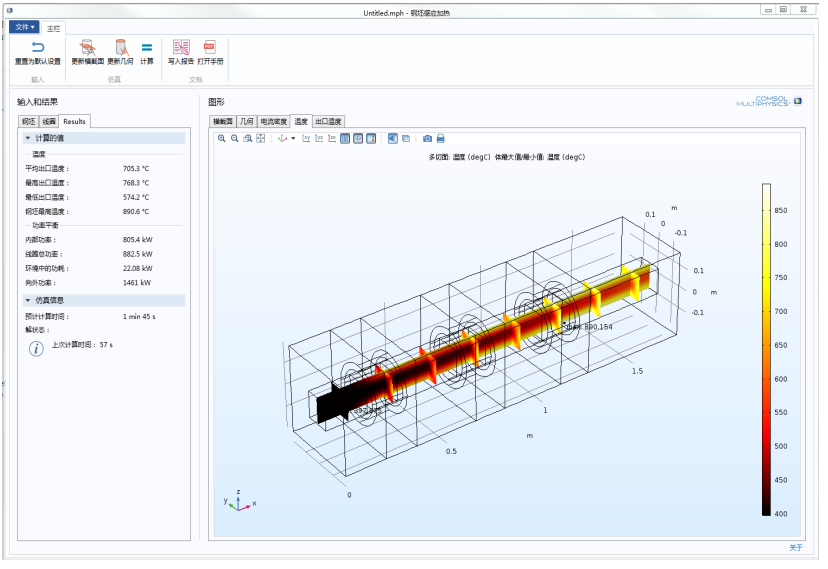
如果要了解更多背景理论知识，请查阅 App 库中 tuning_fork.mph 模型的文档。



钢棒感应加热

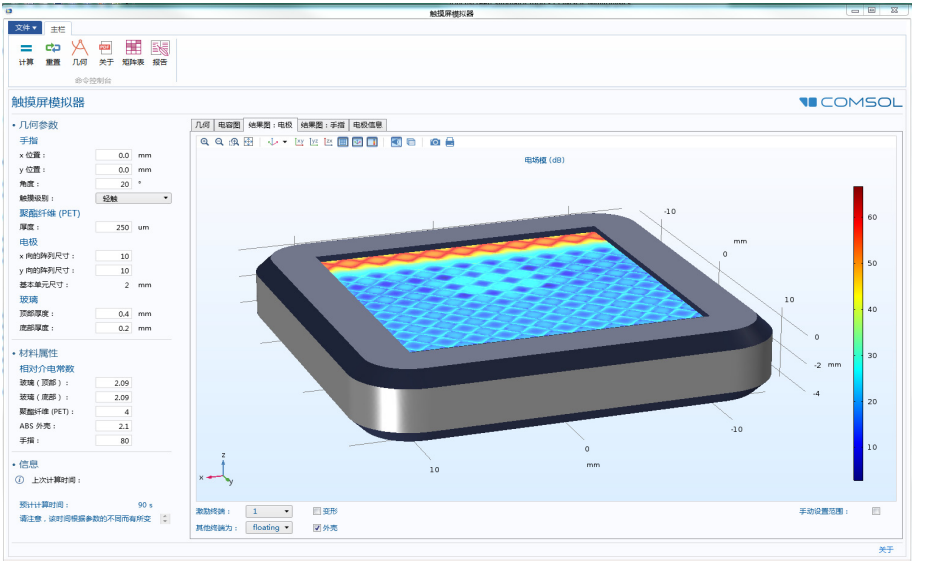
这个 App 可以用来设计一个钢棒的感应加热系统，其中包含一个或多个电磁线圈，钢棒从其中以恒定速度穿过。线圈由交变电流激励，在金属棒表面感应出涡电流，由于焦耳热效应产生热量。

钢棒的横截面、线圈数、位置、尺寸、初始温度和环境温度以及各个线圈的电流等都可以作为输入来指定。当计算结束后， App 中会显示钢棒在加工过程中的三维图，感应电流密度，以及出口末端截面上的二维温度分布。最后， App 计算钢棒中期望的温度范围数值，以及体系的能量平衡。



触摸屏仿真器

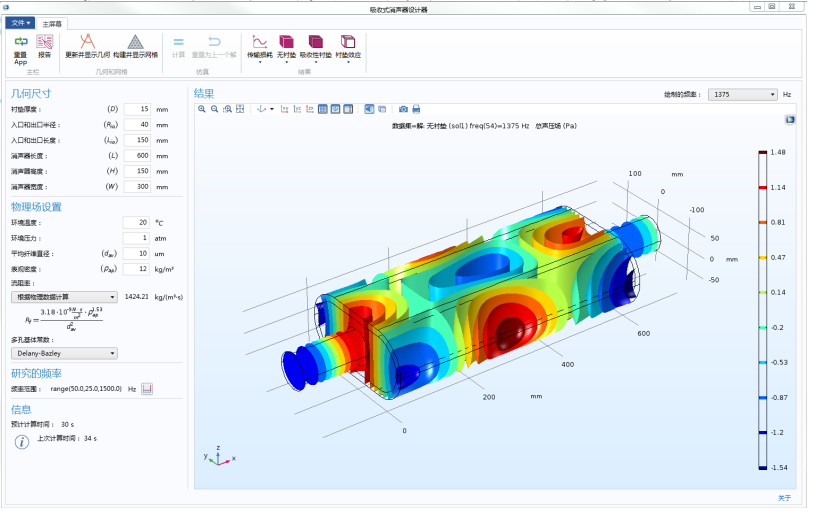
这个 App 计算了一个小型触摸屏与人手指接触时的电容响应。这些信息可以用于电路分析手指的位置。在 App 中，手指的位置和方向由输入参数控制，计算得到的电容矩阵作为输出显示。



消音器设计器

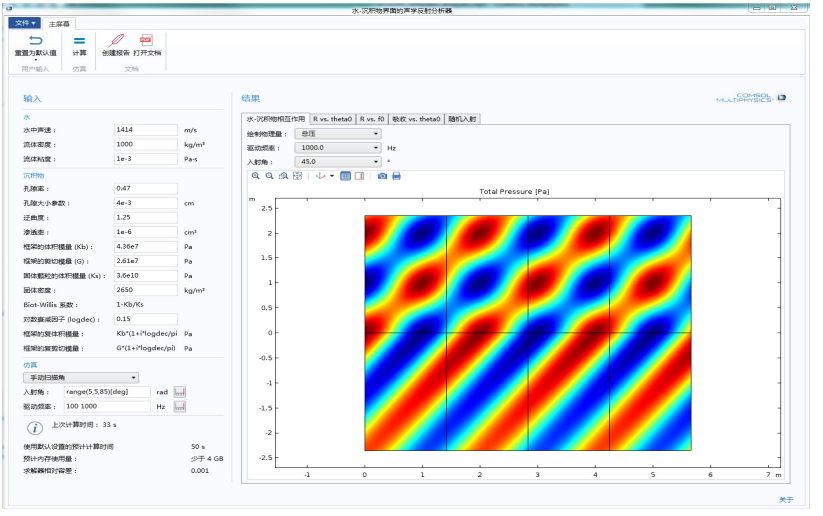
这个 App 的目标是研究和设计一个带多孔衬里的简易谐振式消音器，这种装置常用来衰减由内燃机或 HVAC 系统发出的噪音，通常在指定的频率范围内有很好的效果。衰减的测度称为传输损耗 （TL），以分贝的结果给出随频率变化的衰减函数关系。传输损耗依赖于消音器的几何结构，以及在系统中安放的多孔和纤维状材料的特征。这个 App 用来研究修改消音器的尺寸、环境工作条件，以及多孔衬里的材料属性等产生的影响。也就是说，这些变化如何影响系统的传输损耗。

这个 App 是一个给定的消音器模型的 “动态明细表”的典型示例。一个销售工程师可以携带这类 App 向客户展示为他们专门定制消音器的性能。例如，一个消音器可以是放置在汽车中，并受到空间限制。在这种情况下，它的性能可以很方便地图形化显示，并可以与客户一起研究各种不同的选项。



声反射分析器

这个 App 研究了平面波从水 - 河床界面的反射。以入射角和频率的函数形式确定了反射和吸收系数。而且，基于仿真数据，还可以计算随机入射吸收系数、或者扩散场吸收系数等。流体的材料属性 （本例中为水），以及多孔介质的材料属性 （本例中为半无限沉积层）等都可以进行修改。



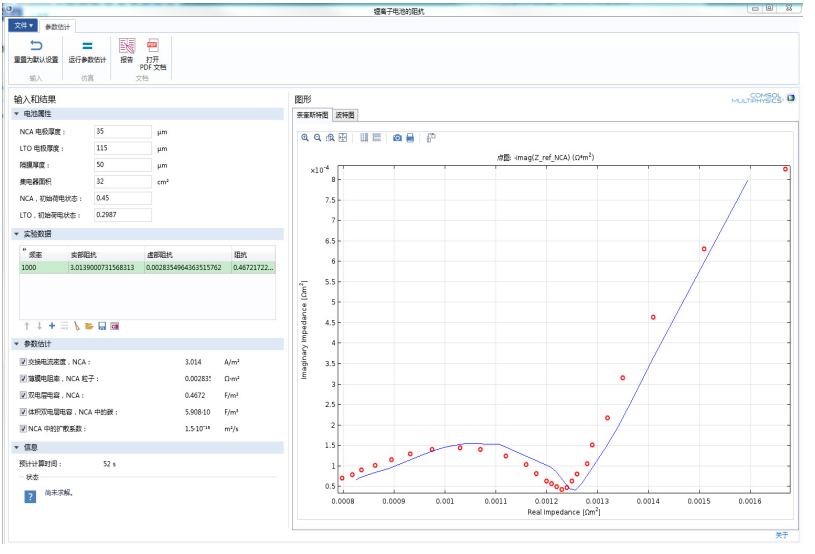
锂离子电池的阻抗

这个 App 的目标是解释实验的电化学阻抗谱 （EIS），并显示如何使用模型和测试结果来估计锂离子电池的属性。

在 App 中，读取 EIS 测试的实验结果作为输入，仿真这些测试，然后基于实验数据进行参数估计。

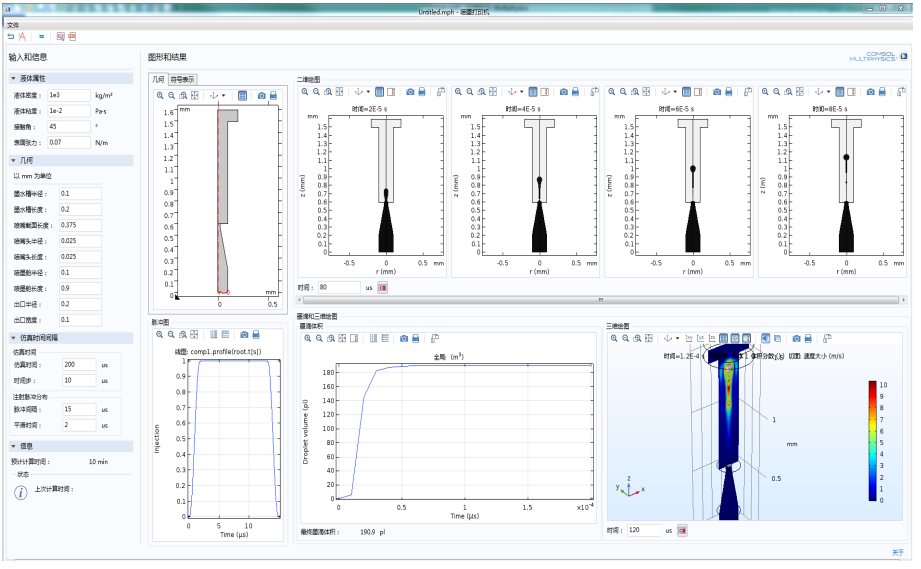
控制参数是交换电流密度、粒子阻抗层的阻抗， NCA 双电层电容，以及正极碳支撑材料的双电层电容。通过测试正极在 10 mHz 到 1 kHz 频率范围内的阻抗来进行拟合。

这个 App 演示加载逗号隔开值 （CSV）文件中的实验数据，并使用优化模块来进行参数估计。



喷墨打印机

这个 App 的目标是设计喷墨打印机的喷嘴形状和操作条件，达到所需的墨滴尺寸，通常这依赖于喷出的墨水的接触角、表面张力、粘度和密度等。结果还揭示了喷出的墨水是否会在最终接触到媒介之前发生破裂。

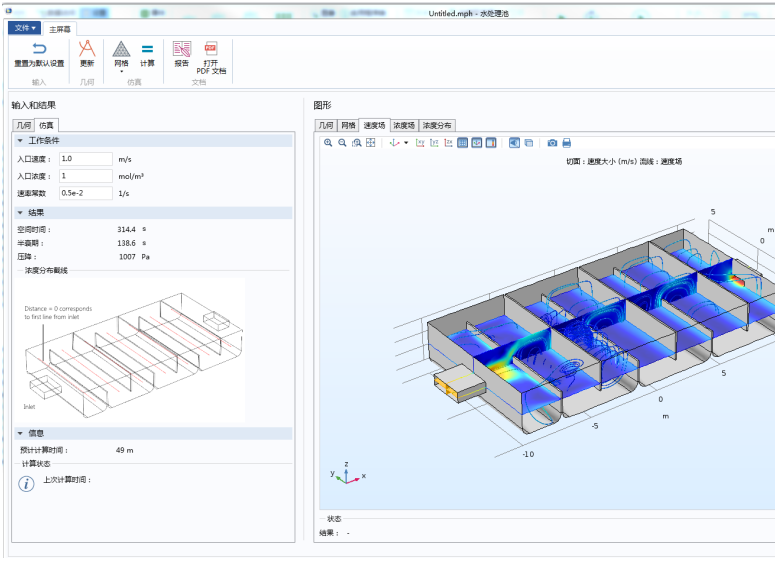


水处理池

这个水处理池 App 的目标是展示三维湍流建模的应用，有趣的是它还考虑了溶液中溶质的材料守恒。这种溶质会发生一级反应，这是一种常见的描述极稀化学物质的衰变过程的反应类型。 App 中还显示了如何使用参数化几何和累加选择来模拟湍流。

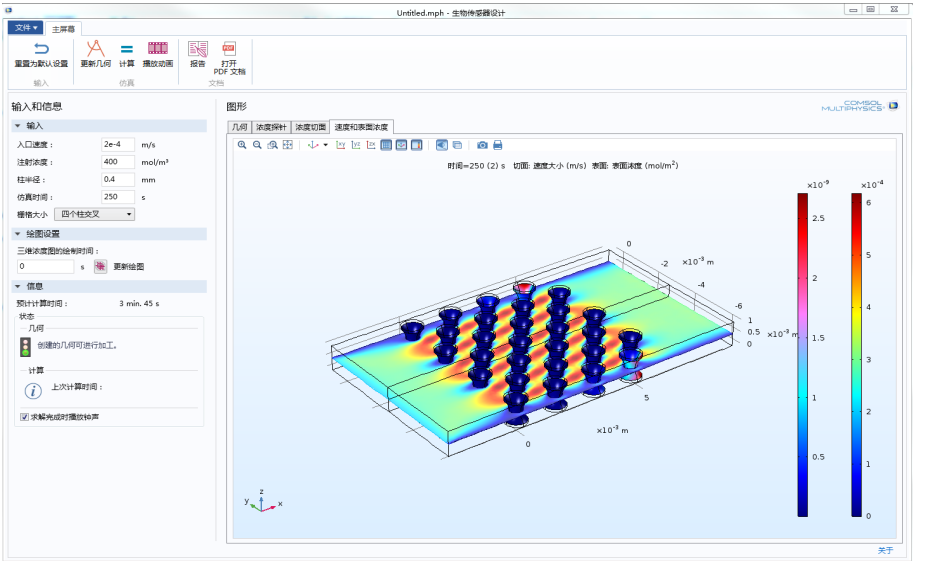
这个 App 可以用于您自己的一个建模稳态湍流和极稀溶质反应耦合的 App 的出发点。

示例中的体系是水处理过程中的加氯消毒池。



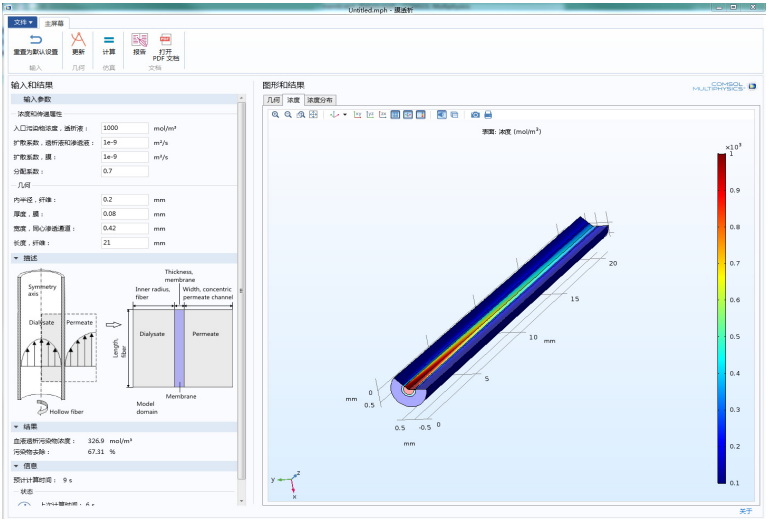
生物传感器设计

一个生物传感器的流动单元池包含一个微柱阵列，用于检测生物分子。这些微柱表面涂有活化材料，可以选择性地从样品流中吸附生物分子。这些生物分子随后会在表面上进行反应。这个 App 允许用户通过修改参数改变传感器的设计，例如微柱直径、间隔，以及入口流速，从而可以研究这些设计如何影响检测效果。几何和操作条件对信号强度和扩散性有很大的影响。此外，在 App 中还报告了由微柱之间的最小间距设置的生产限制。



膜透析

这个 App 模拟了血液中的不良成分通过膜透析装置进行净化。这种模拟的透析装置为中空纤维模块，其中的纤维壁作为透析膜。在纤维内部为透析液的流动，而外部则为渗透液流动。通过改变输入的参数，这个 App 可以检测装置内的不良成分的最大净化速率。

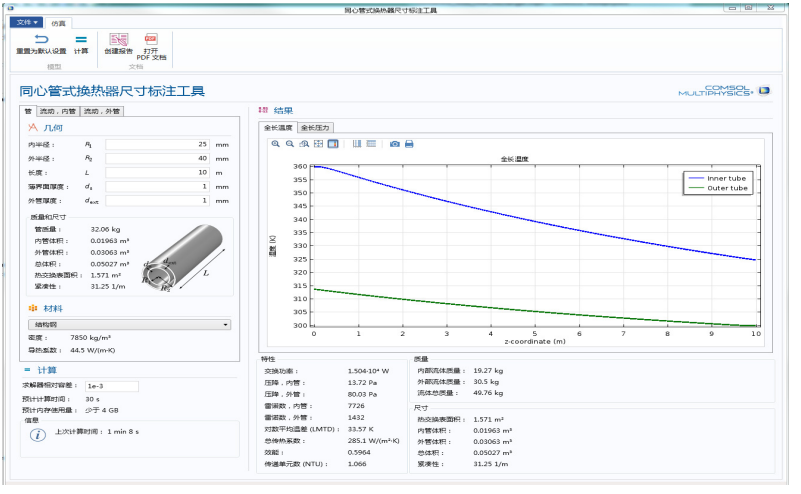


同心管换热器

尺寸是换热器的首要指标。这个 App 的目标是计算这些给定尺寸的数据。

这个示例 App 研究了两个同心圆管套在一起，将两种不同流体分开。这两种流体可以同向流动，或者反向流动。管和流体都可以在用户界面中进行定制。

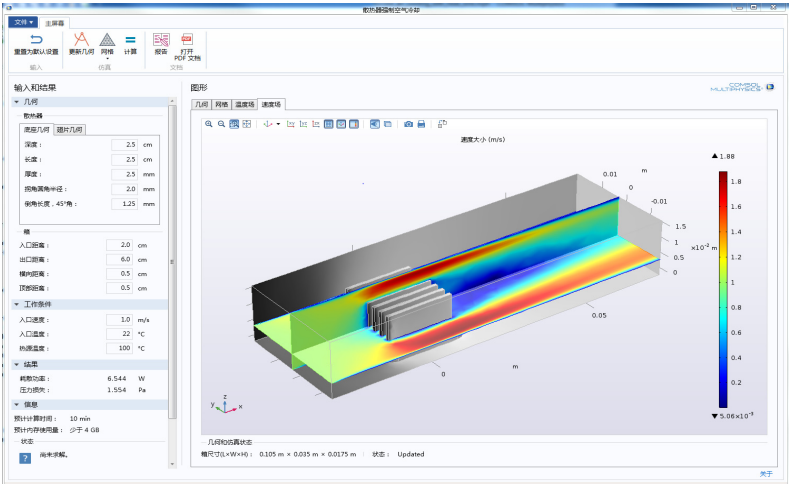
经过计算，可以显示出温度分布和其他一些数据。还可以输入流体属性，例如，体积和质量、可压缩率 （换热面积与换热器体积之比），以及材料属性。



翅片散热器

换热器的效率常常使用它对给定风扇曲线的散热程度来进行衡量。一种进行这种类型实验的可能做法是将散热器放置在周围绝热壁面的矩形流道中，然后测量流道的入口和出口处的温度和压力分布，以及保持散热器的底座维持给定温度所需的热量。在这些条件下，就可以估算出散热器散失的热量，以及流道中的压力降。

这个 App 的目标是通过建模和仿真实现这种标准实验的研究。例如，增加翅片数量，可能会增加散热，直至翅片的数量大到产生了很大的阻碍，使得流动下降并因此减少散热。这意味着在流道中给定总压力降的情况下，可能有一个最优的尺寸和翅片数量，给出最大的冷却效率。这个 App 可以让您实现这种研究。



周期性微结构的等效属性

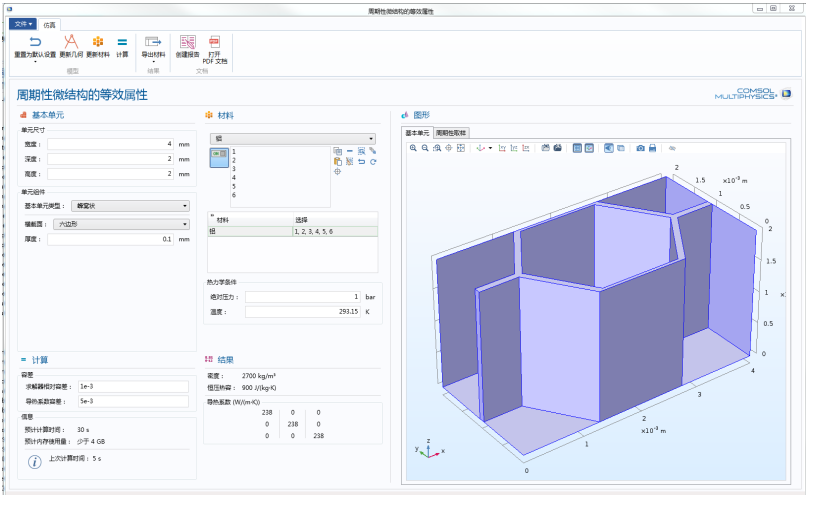
我们经常可以在复合材料中发现周期性微结构，例如碳纤维和蜂窝状结构。它们可以表征为沿着三个方向扩展的重复单元结构。为了减少计算成本，仿真可以将复合材料的所有细节使用等效属性来代替。这个 App 根据一个单元的几何结构和材料属性进行了等效属性的计算，提供了很多选择，分别是 9 种参数化单元和 13 种预定义的材料，可以直接将 App 扩展为平行管单元或添加其他材料。

在扩散类型的方程 （例如传热方程），等效扩散系数采取广义的张量形式。

在这个 App 中，可以在选定的单元形状下，在单元的不同区域使用给定材料计算下列材料属性：

- 密度
- 常压热容
- 传热速率

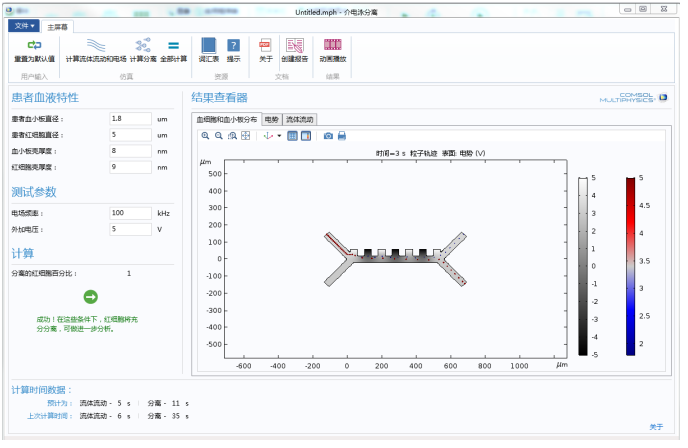
这个 App 中内建的单元库包括很多广泛应用的单元类型，例如，平行堆叠层、纤维增强复合材料，或者蜂窝状结构。当结构设置后，会自动设置物理场，包括单元对边界上的周期性传热条件。



红血球分离

介电泳 （DEP）是一种当溶液中的介电颗粒受到非均匀电场作用时受力的一种现象。粒子受电场感应而极化，然后受到介电泳力，其大小与电势成正比。

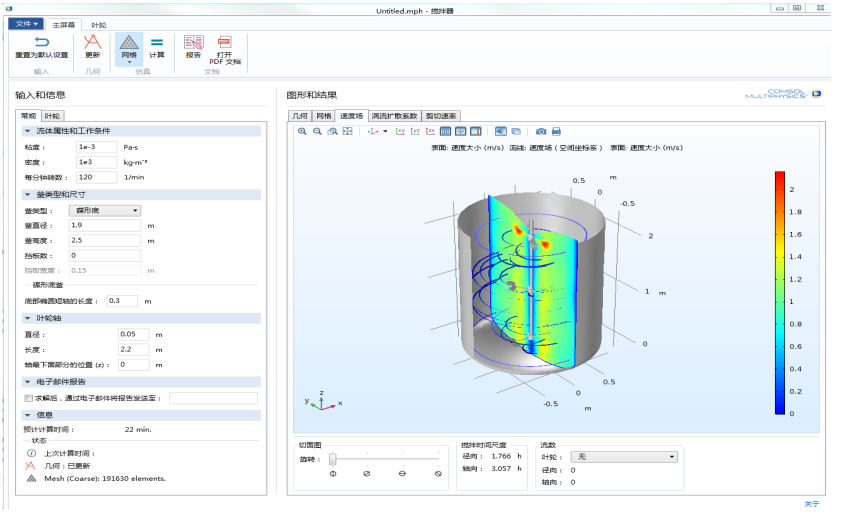
介电泳力受到颗粒的尺寸、形状以及介电属性的影响，从而可以让介电泳现象用于分离不同的颗粒。其中一种应用是在生物工程中，通过介电泳现象实现不同血液细胞的分离。这个示例 App 显示如何从血样中选择性地过滤出红细胞，从而将红细胞与血小板分离。这个过程很重要，因为血小板会使得血液凝固，导致血样受到污染，不利于进行后续的检测。



搅拌器

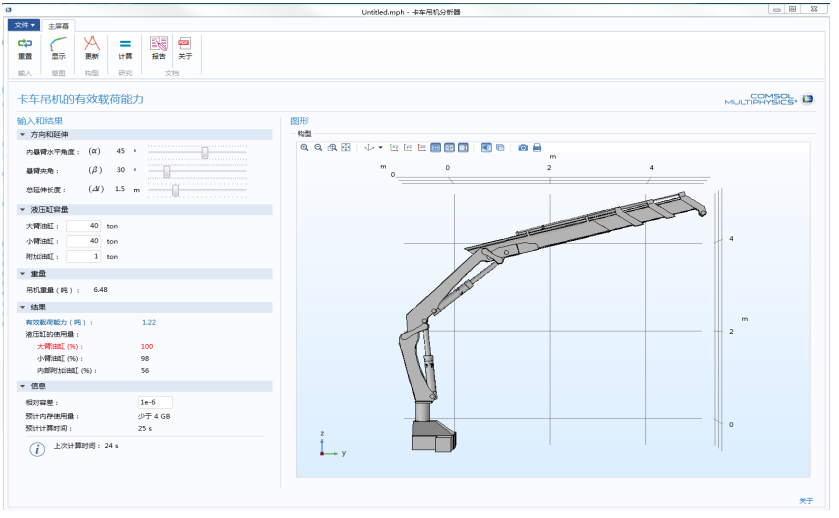
这个搅拌器 App 的目标是提供一个用户友好的界面，使得科学家、工艺设计师，以及工艺工程师可以研究容器、桨，以及工艺条件等对于搅拌效率和驱动桨转动所需功率的影响。 App 可以用来理解和优化搅拌器对特定液体的设计和工艺。不过，也许最重要的是，搅拌器 App 可以作为您自己开发用于建模和仿真搅拌器和反应器 App 的起点。

这个 App 演示了如何使用零件和累加选择来自动设置内嵌模型中的域和边界设置。甚至是当用户创建了很不同的几何结构，也可以自动创建这些设置。



车载起重机分析器

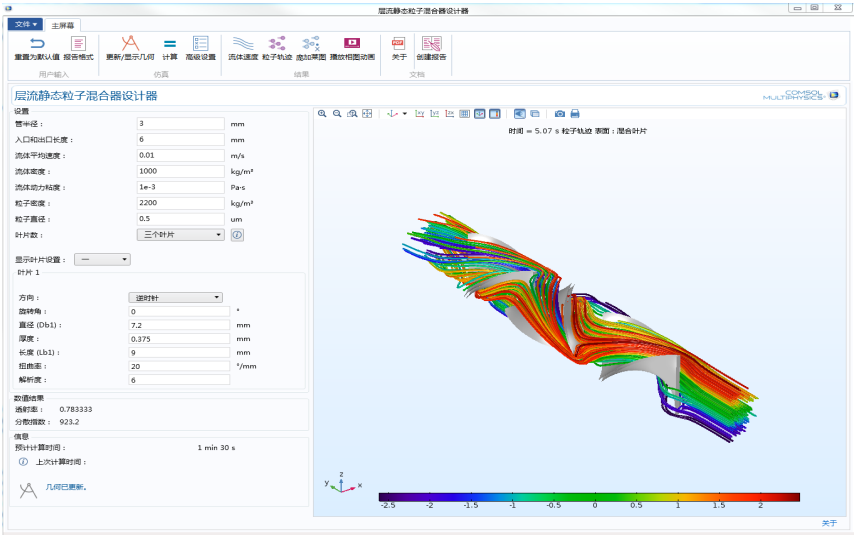
很多卡车装备了吊机 （起重机），用于处理装货。这样的吊机有很多液压缸来控制吊机的运动，以及其他机构。



在这个 App 中，对吊机进行刚体分析，以便找出指定方向和范围内的有效承载能力。其中还提供了所有液压缸的使用率，并高亮显示达到极限的液压缸。可以修改液压缸的有效承载，从而提高有效承载能力和液压缸的使用率。

静态层流颗粒混合器设计器

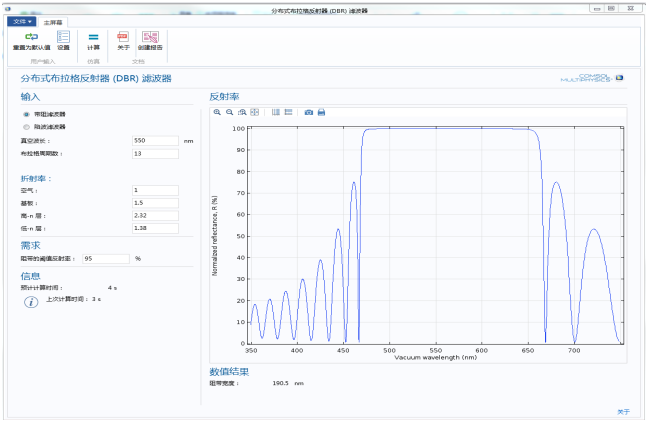
在静态混合器中 （又称无运动或在线混合器），液体通过管道泵入含有静态刀片的管道。这种混合技术特别适用于层流混合，因为在这种流动范畴，只会产生很小的压力损失。这个 App 研究了一个螺旋刀片式静态混合器，通过计算悬浮颗粒经过混合器的轨迹来评估混合效率。其中计算了一种室温下溶于溶剂的物质的静态混合效果。您可以研究颗粒混合过程中液体的影响和颗粒属性，以及静态刀片结构。



分布式布拉格反射滤波器

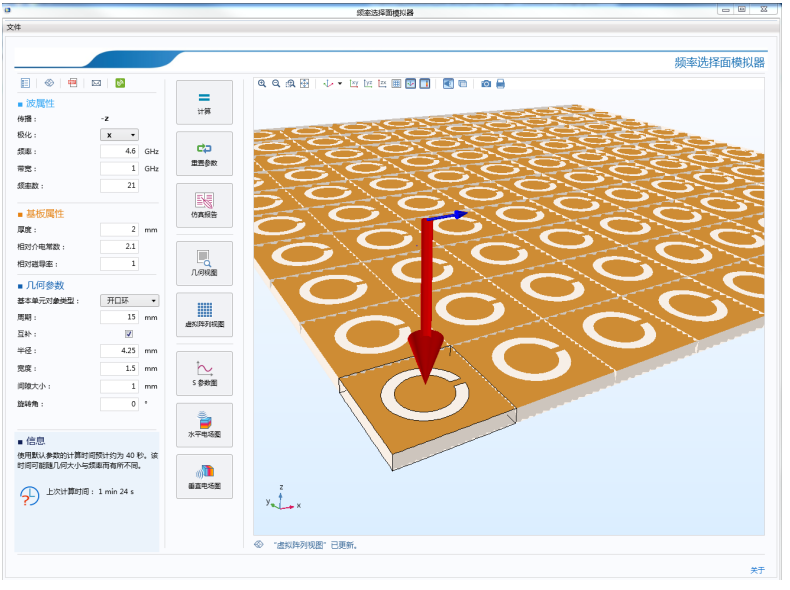
研究了一种由两种材料交替排布形成的分布式布拉格反射器 （DBR）。每种材料具有不同的折射率，在与 DBR 层垂直的方向上形成高和低折射率重复排列。当光经过这种结构传播时，在每个层之间的界面发生反射。多层反射波之间的干涉会导致 DBR 的反射率高度依赖于波长。与常用的金属镜相比， DBR 的主要优势在于它可以很方便地定制选定波长的反射率。

这个 App 可以用来研究一种简易光学缺口滤波器的反射率，它基于两个分布式布拉格反射器之间的三明冶腔体。



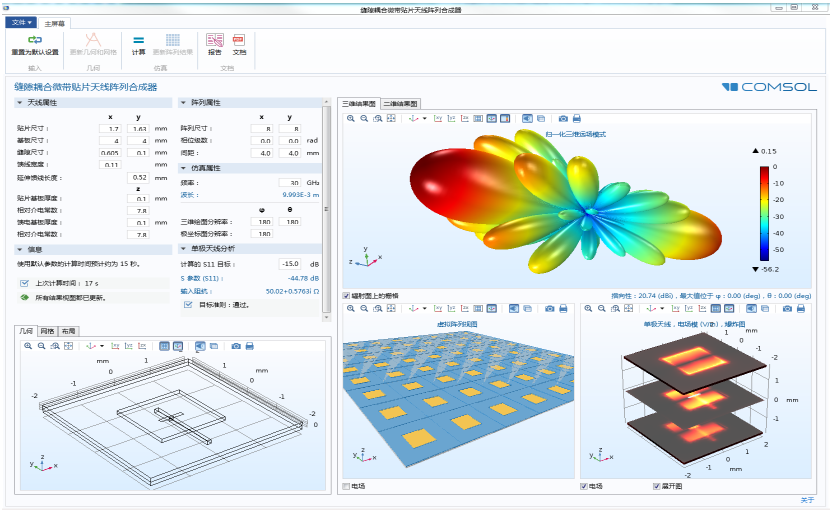
频率选择面仿真器

频率选择面 （FSS）是一种周期性结构，可以产生带通或带止频率响应。这个 App 模拟了一个用户从内建的单元类型中选择指定的周期性结构，其中提供了五种常用的 FSS 单元类型，以及两种预定义的法向入射的极化和传播。分析包括反射和透射谱，单元顶面上的电场模，以及在单元域的垂直截面上分贝尺度的电场模。



微带贴片天线阵列合成器

这个 App 仿真了一个单缝耦合微带贴片天线，它是在一个多层低温共烧陶瓷（LTCC）基板上加工形成。结果包括天线阵列的远场辐射图案，以及它的指向性。远场辐射图案通过阵列因子和单天线辐射图案的乘积近似得到，无需模拟复杂的全阵列模型就可以很有效地进行远场分析。使用默认的输入频率（30GHz），可以很方便地计算 5G 移动网络的相控阵天线原型。这个 App 还演示了当相机围绕天线运动时的动画。

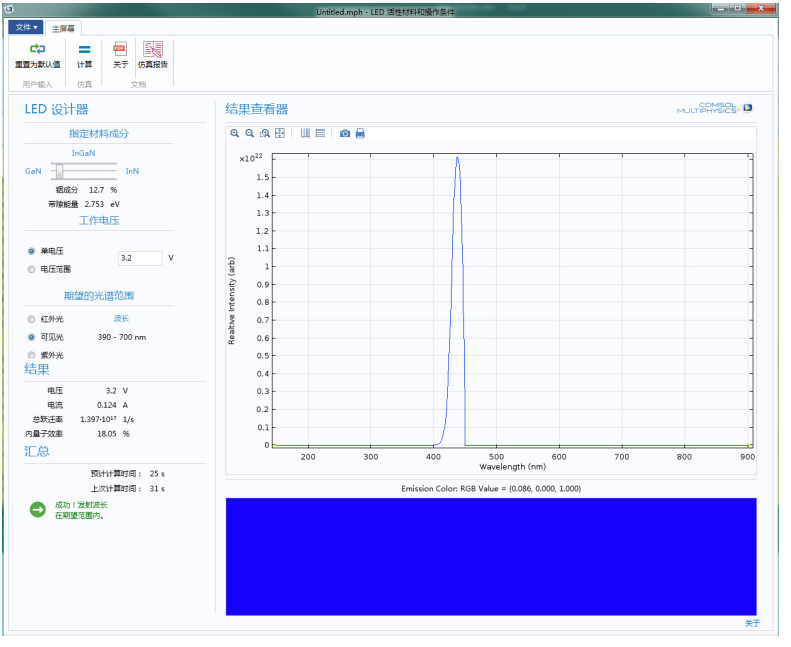


可调波长 LED

蓝光 LED 受到广泛关注，因为它们可以用于现代高效照明。由于具有较大的带隙能，氮化镓广泛用于生成蓝光。这个 App 仿真一个氮化镓基发光二极管的发光特性。

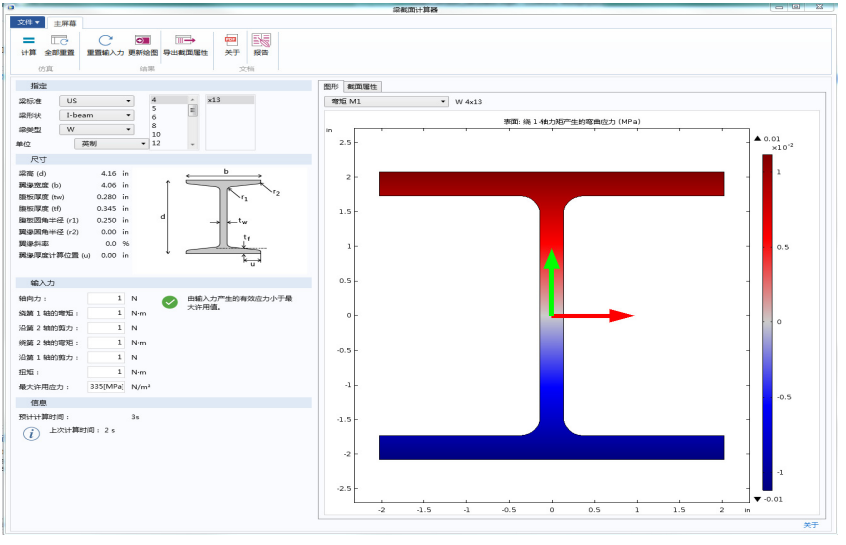
在器件的活性区使用的材料是 InxGa1−xN，即包含镓和铟的混合物，其中铟的成份由 x 给定。这种可变活化区的带隙可以通过改变材料的成份而变，即改变其中的铟成份。因为纯 InN 和 GaN 分别在红外和紫外波段发射，因此可以通过这种技术来调节 InxGa1−xN 在整个可见波段范围内的发射能量。

这个 App 可以控制器件的铟成份和工作电压。可以计算电流、发射强度、场致发光谱，以及内部量子效应。可以输入单个工作电压，或者电压范围。如果输入电压范围，还可以计算电流 - 电压曲线，从而确定器件的开关电压。这个 App 中大量地用到了方法。



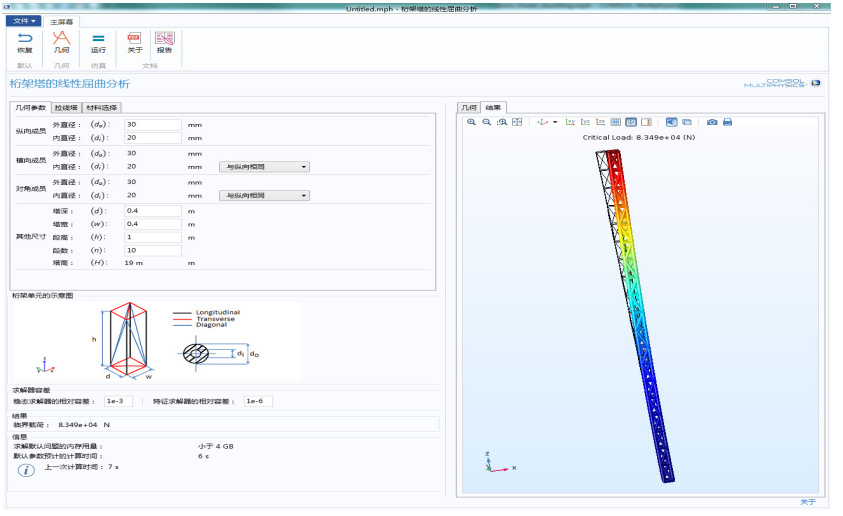
梁横截面计算器

这个 App 计算了一个设计的钢梁横截面的梁截面属性，还可以计算受到一组给定的力和扭矩条件时截面上的应力分布。可以选用美标和欧标梁。如果有 LiveLinkTM for Excel® 许可证，所有的输入和结果数据都以表格形式显示，并且可以导出成 Excel® 文件。还可以编辑包含梁尺寸数据的 Excel® 电子表格，并将它重新导入回 App 进行计算。



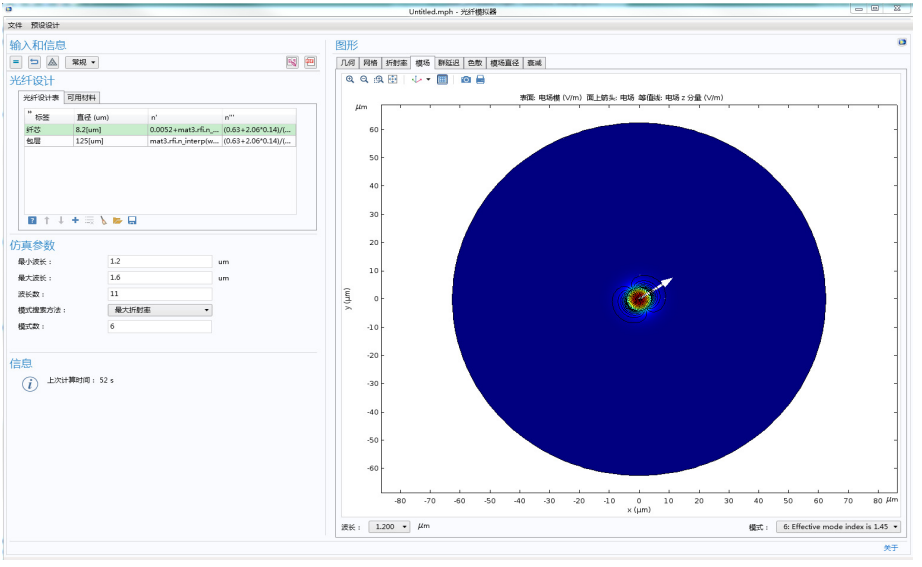
桁架塔屈曲

屈曲分析用于找出临界受压载荷，当载荷超出这个值时会使得结构失稳。这个 App 可以模拟桁架塔在垂直的压力载荷作用下的屈曲，其中塔有两种支撑形式，分别是支护和张力缆。这个 App 的目标是计算和分析塔在不同几何条件下的屈曲载荷，例如，不同的塔高，截面积，以及不同材料。其中还引入了死载荷效应 （桁架、张力缆的自重，以及缆线的预张力）。



光纤模拟器

这个 App 计算了一个同心圆介电层结构的模式分析，每层通过外直径来描述，可指定折射率的实部和虚部。折射率表达式可以包含随波长和径向距离的变化关系。因此，这个模拟器可以用来分析阶跃光纤和渐变光纤，这些光纤具有任意数量的同心圆层。计算结果包括群延迟和色散系数。



表面等离激元线光栅

这个 App 计算了在一个介电基质上的线光栅对于不同的透射和反射波 （m=0）和一阶及二阶衍射 （m = ±1 和 ±2）的衍射效率随入射角度变化的函数关系。平面波的入射角从法向入射扫描到切向入射。这个 App 还显示了选定入射角的多个光栅周期内的电场模结果图。

