

COMSOL Multiphysics

Application Builder Reference Manual

日本語訳版

Application Builder Reference Manual 日本語訳版

© 1998–2017 COMSOL

www.comsol.com/patents に掲載している米国特許第 7,519,518 号、第 7,596,474 号、第 7,623,991 号、第 8,457,932 号、第 8,954,302 号、第 9,098,106 号、第 9,146,652 号、第 9,323,503 号、第 9,372,673 号、および第 9,454,625 号で保護。特許出願中。

本書および本書に記載されているプログラムは、COMSOL のソフトウェア使用許諾契約 (www.comsol.com/comsol-license-agreement) に基づいて提供されており、使用許諾契約の条項に従う場合にのみ、使用または複製をすることができます。

COMSOL、COMSOL logo、COMSOL Multiphysics、COMSOL Desktop、COMSOL Server、および LiveLink は、COMSOL AB の登録商標または商標です。その他の商標は全て、それぞれの所有者の所有物であり、COMSOL AB およびその子会社、製品は、これらの商標の所有者とは無関係であり、推奨、後援または支援を受けていません。これらの商標の所有者のリストについては、www.comsol.com/trademarks を参照してください。

バージョン: COMSOL 5.3a

日本語訳: 計測エンジニアリングシステム株式会社

※ もし日本語訳と原文(英語版)に差違がある場合は、原文を正とします。
但し、原文に誤記などがある場合は、弊社として(注)を示しています。

お問い合わせ

一般的なお問い合わせ、技術サポートのお問い合わせ、住所および電話番号の検索については、www.comsol.com/contact にアクセスしてください。

住所および連絡先情報については、www.comsol.com/contact/offices でも確認できます。

サポート窓口へのお問い合わせについては、www.comsol.com/support/case の COMSOL Access ページにあるリクエストフォームを送信してください。

その他の有用なリンク:

- サポートセンター: www.comsol.com/support
- 製品のダウンロード: www.comsol.com/product-download
- 製品のアップデート: www.comsol.com/support/updates
- COMSOL ブログ: www.comsol.com/blogs
- フォーラム: www.comsol.com/community
- イベント: www.comsol.com/events
- COMSOL ビデオギャラリー: www.comsol.com/video
- サポートナレッジベース: www.comsol.com/support/knowledgebase

部品番号: CM020010

目次

Chapter 1: イントロダクション	7
アプリケーションビルダーについて	8
アプリケーションビルダーで何ができる?	8
ドキュメントへのアクセスは?	8
マニュアルの概要	10
Chapter 2: アプリケーションビルダーのツール	11
はじめに	12
アプリケーションビルダーの開始	13
アプリケーションビルダーの起動	13
COMSOL デスクトップからアプリケーションビルダーを開く	14
アプリケーションライブラリのアプリケーション例	14
アプリケーションビルダーとアプリケーションツリー	14
ホームツールバー	14
コンポーネントのカット、コピー、複製、削除、およびペースト	16
アプリケーション間のアプリケーションコンポーネントのコピー	17
メインウィンドウブランチ	18
メインウィンドウノード	18
メニューバーノード	19
ツールバーノード	19
メニューノード	19
ファイルメニューノード	20
リボンノード	20
リボンタブノード	20
リボンセクションノード	20
項目ノード	21
項目切替えノード	23
セパレーターノード	25
フォームブランチ	26
フォームノード	26
フォームノード(サブノード)	26
イベントブランチ	29
イベントノード	29
イベントノード(サブノード)	29
宣言ブランチ	32

配列シンタックス	33
文字列ノード	33
ブーリアンノード	33
整数ノード	34
倍精度ノード	34
配列1D文字列ノード	35
配列1Dブーリアンノード	35
配列1D整数ノード	36
配列1D倍精度ノード	36
配列2D文字列ノード	37
配列2Dブーリアンノード	37
配列2D整数ノード	38
配列2D倍精度ノード	39
選択リストノード	39
アクティベーション条件	40
ファイルノード	40
単位セットノード	41
グラフィックスデータノード	41
ショートカットの追加	42
宣言とコマンドシーケンスでの初期値と引数の編集	42
メソッドブランチ	46
メソッドブランチ	46
モデルメソッドブランチ	46
メソッドノード	46
モデルメソッドノード	46
ライブラリブランチ	48
画像	48
サウンド	48
ファイル	49
アプリケーションの計画と準備	50
アプリケーションの準備	50
モデルからのアプリケーション作成	51
コードをクリップボードにコピー	51
アプリケーションのテスト	52
キーボードショートカット	53
Chapter 3: フォームの操作	55
はじめに	56
フォームを作成するためのフォームとツールの概要	56
フォームの操作と新規フォームウィザードの使用	56
データアクセス	59

フォームツールバー	59
フォームウィンドウのレイアウトモード	61
スケッチモード	61
グリッドモード	62
フォームのプレビューとテスト	66
フォームオブジェクトでのメソッドの実行	66
フォームオブジェクト	67
フォームオブジェクトの概要	67
入力フィールド	68
ボタン	72
トグルボタン	75
チェックボックス	79
コンボボックス	81
テキストラベル	83
単位	85
方程式	86
ライン	87
データ表示	89
グラフィックス	91
ウェブページ	94
画像	95
ビデオ	96
進捗バー	98
ログ	99
メッセージログ	101
結果テーブル	102
フォーム	104
フォームコレクション	105
カードスタック	107
カード	109
ファイルインポート	111
情報カードスタック	113
配列入力	116
ラジオボタン	118
選択入力	121
テキスト	123
リストボックス	125
テーブル	128
カスタムツールバーアイテム編集ダイアログボックス	132
スライダー	134
ハイパーリンク	136
ツールバー	138
スペーサー	139

Chapter 4: メソッドの操作 141

概要	142
メソッドエディターウィンドウを開く	142
コーディングとメソッドの概要	142
アプリケーションビルダーウィンドウ	143
メソッドウィンドウ	143
メソッドツールバー	143
メソッドノードとメソッドエディターウィンドウ	145
ユーティリティクラスノード	146
外部 Java ライブラリノード	147
外部 C ライブラリノード	147
外部 C ライブラリの使用	148
ファイルスキームとファイル処理	151
クライアントファイルシステムとのファイルの送受信	152
メソッドの作成	153
シンタックスハイライトとコメント	153
コード補完とツールチップのヘルプ	154
コードの折り畳み	159
言語要素の追加	160
モデル表現の追加	160
モデルコードとフォームオブジェクトの追加	161
ソースコードがマップされるノードへの移動	163
コードの記録	164
ショートカットの使用	164
ローカル変数とそのタイプ宣言の作成	165
他のメソッドの直接的な呼び出し	166
変数宣言で定義されたプロパティの使用	166
テキストの検索	166
字下げと空白フォーマット	167
ブレースマッチング	167
アプリケーションのデバッグ方法	168
コンパイルエラーの表示	168
デバッグツール	168
エラーと警告ウィンドウ	169
メソッドのランタイムエラーの処理	170
実行中のメソッドの停止	170

イントロダクション

マルチフィジックスモデルに基づいてカスタムアプリケーションを作成するための一連のツールであるアプリケーションビルダーの使用方法については、このガイドを参照してください。アプリケーションビルダーは COMSOLDesktop®で直接利用でき、包括的なユーザーインターフェースとプログラミングツールが含まれています。アプリケーションビルダーとアプリケーションライブラリの概要については、*Introduction to the Application Builder* を参照してください。そこには、多くのサンプルアプリケーションが掲載されています。

以下に、この章の構成を示します。

- [アプリケーションビルダーについて](#)
- [マニュアルの概要](#)

アプリケーションビルダーについて

ここでは、以下が掲載されています。

- [アプリケーションビルダーで何ができる？](#)
- [ドキュメントへのアクセスは？](#)

アプリケーションビルダーで何ができる？

アプリケーションビルダーは、COMSOL Multiphysics®モデルからカスタムアプリケーションを作成するためのツールを提供します。グラフィカルなツールとエディター、および組み込みの言語要素と Java®コードを使用して、ユーザー入力、デザイン、および結果を含めるアプリケーションを調整できます。シミュレーションアプリケーションには、次のようなさまざまな用途があります。

- さまざまな設計パラメータの効率的なテスト
- 製品開発ワークフローの合理化

COMSOL ユーザーは、COMSOL マルチフィジックスでアプリケーションを実行することができます。COMSOL アプリケーションは、COMSOL マルチフィジックスを使用していない同僚や顧客に対して、ネットワーク上またはクラウド上の中央にインストールされた COMSOL Server™に接続させることで利用可能にすることもできます。同僚や顧客は、ウェブブラウザまたは COMSOL クライアントを介して、COMSOL サーバー上でアプリケーションを実行することができます。

ドキュメントへのアクセスは？

多くのインターネットリソースが、ライセンス情報や技術情報などの COMSOL に関する多くの詳細情報を提供しています。電子ドキュメント、トピックベース（またはコンテキストベース）のヘルプ、およびアプリケーションライブラリへのアクセスは、全て COMSOL Desktop®を介して行います。



ドキュメントをコンピュータ上の PDF ファイルとして閲覧している場合、[青色のリンク](#)は参照している別のガイドのコンテンツを開くことはできません。但し、COMSOL マルチフィジックス内のヘルプシステムを使用している場合には、これらのリンクは他のモジュール（ライセンスを持っている限り）、アプリケーションのサンプルやドキュメントに接続してくれます。

E メールによる COMSOL への連絡

一般的な製品情報については、COMSOL (info@comsol.com) までお問い合わせください。

COMSOL から COMSOL 製品の技術サポートを受けるには、お近くの COMSOL 担当者に連絡するか、support@comsol.com までご質問ください。自動通知とケース番号が電子メールで送信されます。

COMSOL WEBSITES

COMSOL website	www.comsol.com
Contact COMSOL	www.comsol.com/contact
Support Center	www.comsol.com/support
Product Download	www.comsol.com/product-download
Product Updates	www.comsol.com/support/updates
Discussion Forum	www.comsol.com/community
Events	www.comsol.com/events
COMSOL Blog	www.comsol.com/blogs
COMSOL Video Gallery	www.comsol.com/video
Support Knowledge Base	www.comsol.com/support/knowledgebase

マニュアルの概要

Application Builder Reference Manual には、アプリケーションビルダーのツールと機能に関する包括的な情報が記載されています。この紹介に加えて、さらに次の章があります。

- **アプリケーションビルダーのツール:** アプリケーションビルダーのツールの概要、アプリケーションビルダーツリーのノード、およびホームリボンツールバーのツールの詳細。
- **フォームの操作:** アプリケーションのフォームを作成および設計するための、アプリケーションビルダー内のツールに関する情報。
- **メソッドの操作:** アプリケーションの機能拡張のためにメソッドとクラスのコードを作成および記述するための、アプリケーションビルダー内のツールに関する情報。

次のドキュメントも参照してください。

- *Introduction to the Application Builder:* アプリケーションビルダーの概要について記載されています。
- *Application Programming Guide:* メソッドエディターを使用するアプリケーション用のコードを記述するためのガイドです。
- *COMSOL Multiphysics Programming Reference Manual:* アプリケーションにメソッドを追加してプログラミングすることができます。この場合のモデルオブジェクトのコマンドについての情報が記載されています。

アプリケーションビルダーのツール

この章では、カスタムアプリケーションの作成に役立つ、アプリケーションビルダーで使用可能なツールの概要を説明しています。このツールには、フォームとウィンドウを設計するためのエディターとユーザーインターフェイスコンポーネント、およびアプリケーション固有のアクションとメソッドを追加するためのメソッドエディターやコーディングとデバッグツールが含まれています。

以下に、この章の構成を示します。

- [はじめに](#)
- [アプリケーションビルダーの開始](#)
- [メインウィンドウブランチ](#)
- [フォームブランチ](#)
- [イベントブランチ](#)
- [宣言ブランチ](#)
- [メソッドブランチ](#)
- [ライブラリブランチ](#)
- [アプリケーションの計画と準備](#)
- [モデルからのアプリケーション作成](#)
- [キーボードショートカット](#)

はじめに

アプリケーションビルダーには、COMSOL モデルに基づいてアプリケーションを作成および展開するための包括的なツールセットが含まれています。アプリケーションの作成に使用するアプリケーションビルダーの主な部分は次の通りです。

- COMSOL デスクトップのアプリケーションビルダーウィンドウでは、アプリケーションを定義するノードのあるツリーと、アプリケーションを作成するツールを備えたリボンツールバーが含まれています。[アプリケーションビルダーとアプリケーションツリー](#)、および、アプリケーションビルダーツリー内のブランチやノードについてさらに詳細に説明しているこの章のセクションを参照してください。
- アプリケーションに適したさまざまなフォームコンポーネント（ユーザーインターフェースコントロール）によってフォーム（ユーザーインターフェース）を作成および設計するためのツールが備わっています。詳細は、[フォームの操作](#)の章を参照してください。
- ユーザーインターフェースイベントに接続可能なカスタムコードを含めるためのメソッドとクラスを作成および編集するためのツールが備わっています。詳細については、[メソッドの操作](#)の章を参照してください。

開始するにはまず、COMSOL マルチフィジックスや多くのアドオン製品に含まれているアプリケーションを検索して実行することができます。[アプリケーションライブラリ](#)ウィンドウでは、**Applications** の下に表示されるアプリケーションは実行可能なアプリケーションのサンプルであり、フォーム（ユーザーインターフェース）とメソッド（アプリケーションの中に機能を追加するコード）を作成する方法を学習するために開いて調べることができます。[アプリケーションライブラリ](#)ウィンドウの詳細は、*COMSOL Multiphysics Reference Manual* を参照してください。

アプリケーションビルダーの開始

アプリケーションビルダの起動

アプリケーションビルダーを起動するには、ホームツールバーのアプリケーションビルダーのボタン (A) をクリックします。その後、COMSOL デスクトップが切り替えられてアプリケーションビルダーのツールバーが表示されます (または、環境設定ダイアログボックスのアプリケーションビルダーのページにあるアプリケーションビルダー用に別のデスクトップウィンドウを使用のチェックボックスを選択した場合は、別のデスクトップウィンドウで開かれます)。アプリケーションの開発が完了したら、ファイルメニューからアプリケーションを実行を選択し、アプリケーションの MPH ファイルを選択して実行することができます。メインウィンドウでアプリケーションをブラウズして実行することもできます。下図では、COMSOL デスクトップでアプリケーションにアクセスして実行する方法についても示しています。アプリケーションビルダーからモデルビルダーに戻るには、アプリケーションビルダーのホームツールバー (H) 上でモデルビルダーをクリックします。

アプリケーションビルダーウィンドウは、アプリケーションツリー、コンテキストメニュー、ツールバー、および全てのアプリケーションツリーノードの設定ウィンドウに関して、モデルビルダーと同じように機能します。アプリケーションツリーのノードでは、実行可能なアプリケーションのフォーム、イベント、メソッド、およびその他の部分を表しています。フォームのコンテキストリボンツールバーからグラフィカルユーザーインターフェース (GUI) コンポーネント (フォームオブジェクト) を追加し、アプリケーションビルダーのグラフィックツールを使用してインタラクティブに配置することができます (フォームでの操作を参照)。また、アプリケーションの実行時にフォームまたはアプリケーション全体がどのように見えるかをプレビューすることもできます。アプリケーションビルダーは Microsoft Windows® オペレーティングシステムインストールで使用できますが、アプリケーションビルダーを使用して作成されたアプリケーションは、COMSOL サーバーを利用してウェブブラウザで実行するなど含め、全てのオペレーティングシステムに展開して使用することが可能です。

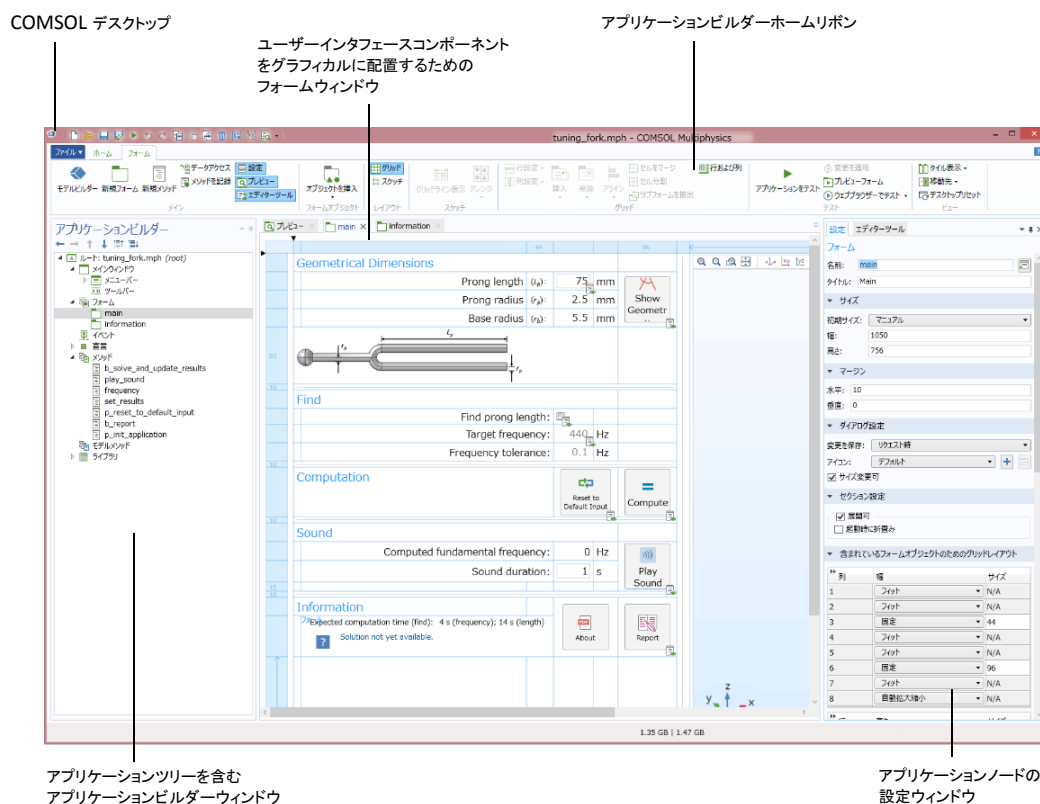




Figure 2-1: アプリケーションビルダーでは、COMSOL デスクトップをアプリケーション設計向けに適合させています。

COMSOL デスクトップからアプリケーションビルダーを開く

COMSOL デスクトップでは、アプリケーションビルダーとマルチフィジックスを切り替えることができます。ホームツールバーで **アプリケーションビルダー**  をクリックしてアプリケーションエディターを開きます。これは、Ctrl + Alt + A を押すことによっても可能です。その後、アプリケーションのユーザーインターフェースを変更したり、アプリケーションのコードの作成や編集をしたりすることができます。

逆に、アプリケーションビルダーでホームツールバーの **モデルビルダー**  をクリックして COMSOL マルチフィジックスに戻ることができます。Ctrl + Alt + M を押すことによっても可能です。


アプリケーションライブラリのアプリケーション例

アプリケーションライブラリウィンドウでは、各 COMSOL 製品のアプリケーションライブラリを参照できます。それらのほとんどには **Applications** フォルダが含まれており、そのフォルダにはアプリケーションビルダーで開いて実行できる実行可能なアプリケーションのサンプルが入っています。これらのアプリケーションを、独自のアプリケーションを設計する場合の出発点、あるいはアイデアの創造に利用することができます。例えば、ユーザーインターフェースの設計でフォームオブジェクトを使用する方法や、アプリケーションの機能を拡張するためのメソッドを作成する方法などがあります。

アプリケーションビルダーとアプリケーションツリー

COMSOL デスクトップの **アプリケーションビルダー**ウィンドウには、アプリケーションツリーがあります。このツリーには、アプリケーションビルダーのデータ構造のノードと埋め込みモデルのノードが表示されています。埋め込みモデルのノードの全ては、モデルビルダーを使用している場合のモデルツリーのノードと全く同じように動作します。また、アプリケーションツリーには、アプリケーションの root ノードと、その root ノードの下にある以下のブランチが含まれています。

- [メインウィンドウブランチ](#)
- [フォームブランチ](#)
- [イベントブランチ](#)
- [宣言ブランチ](#)
- [メソッドブランチ](#)
- [ライブラリブランチ](#)




アプリケーションビルダーの root ノード () はモデルビルダーの root ノードに似ていますが、アプリケーション特有の設定を含む **アプリケーション** というセクションもあります (詳細は *COMSOL Multiphysics Reference Manual* の [The Root Settings and Properties Windows](#) を参照してください) 。







ホームツールバー

ホームツールバーは常に使用可能で、アプリケーションビルダーで最も一般的な機能にアクセスするためのボタンが含まれています。

メインセクション


このセクションは、さまざまなウィンドウに移動して新しいフォームやメソッドを作成するための以下のボタンを含んでいます。

- **モデルビルダー**のボタン () : アプリケーションビルダーからモデルビルダーのウィンドウと標準の COMSOL デスクトップに切り替えます。
- **新規フォーム**のボタン () : 新規フォームのウィザードを使用して新しいフォームを作成します。 [フォームの操作と新規フォームウィザードの使用](#) を参照してください。
- **新規メソッド**のボタン () : モデル内に新しい **メソッド** または **モデルメソッド** のノードを作成し、そのコードを新しいエディタータブで開きます。 [メソッドノードとメソッドエディターウィンドウ](#) を参照してください。

- **データアクセスボタン** (): モデルに依存するデータとプロパティ、およびデータアクセス機能を通して実行中のアプリケーションで変更可能なアプリケーション固有のプロパティを追加します。[データアクセス](#)を参照してください。
- **メソッド記録のボタン** (): 埋め込みモデルで操作の記録セッションを開始し、そのメソッドで後でコードとして使用できる新しいメソッドを作成します。記録が開始されると、そのボタンは**記録停止のボタン** () に変わり、クリックすると記録が終了します。[コードの記録](#)を参照してください。
- **設定ボタン** (): **設定**ウィンドウを移動したり、開いたり、閉じたりします。
- **プレビューボタン** (): アプリケーションのフォームとメソッドのライブプレビュー用の**プレビュー**ウィンドウを表示または非表示にします。**プレビュー**ウィンドウでは、アプリケーションのフォームとメソッドのプレビューをスクロールして行うことができます。例えば、フォームとやりとりするメソッドを操作している場合などに便利です。**プレビュー**ウィンドウにフォームのプレビューを表示するには、**アプリケーションビルダー**ウィンドウでフォームまたはメソッドノードを選択します。
- **エディターツールボタン** (): **エディター**ツールウィンドウを表示または非表示にします。ここでは、よく使う COMSOL マルチフィジックスのモデル操作を選択してメソッドに挿入したり、それらに基づいてフォームオブジェクトを生成したりすることができます。[モデルコードとフォームオブジェクトの追加](#)を参照してください。









イベントセクション

このセクションには、イベントを作成するためのボタンが一つ含まれています。

- **イベントボタン** (): イベントを定義するための**イベント**ノードが追加されます。[イベントブランチ](#)を参照してください。







宣言セクション

このセクションは、変数、ファイル、選択リストの宣言を作成するための以下に示すボタンを含んでいます。[宣言ブランチ](#)を参照してください。

- **スカラーボタン** (): 文字列、ブーリアン、整数、または倍精度のスカラーを追加選択するメニューを開きます。
- **配列 1D ボタン** (): 文字列、ブーリアン、整数、または倍精度の 1D 配列を追加選択するメニューを開きます。
- **配列 2D ボタン** (): 文字列、ブーリアン、整数、または倍精度の 2D 配列 (行列) を追加選択するメニューを開きます。
- **More 宣言** () の下では、以下の項目から選択します。
 - **ファイル** (): ファイルにアクセスするためのファイル宣言を追加します。
 - **選択リスト** (): 許可された値のリストを定義するための選択リストを追加します。
 - **単位セット** (): 単位セットを追加します。
 - **グラフィックスデータ** (): グラフィックスウィンドウから抽出されるデータにアクセスするためのグラフィックスデータを追加します。










ライブラリセクション

このセクションには、画像、サウンド、ファイルのライブラリを開き、外部コードとユーティリティクラスを追加するための以下のボタンが含まれています。

- **画像ボタン** (): 画像ライブラリを開きます。[画像](#)を参照してください。
- **サウンドボタン** (): サウンドライブラリを開きます。[サウンド](#)を参照してください。
- **ファイルボタン** (): ファイルライブラリを開きます。[ファイル](#)を参照してください。
- **ユーティリティクラスボタン** (): Java ユーティリティクラスを追加します。[ユーティリティクラスノード](#)を参照してください。
- **外部 Java ライブラリボタン** (): 外部 Java ライブラリを追加します。[外部 Java ライブラリノード](#)を参照してください。
- **外部 C ライブラリボタン** (): 直接リンクしたネイティブコードライブラリを追加します。[外部 C ライブラリノード](#)を参照してください。




メインウィンドウセクション

このセクションには、アプリケーションのメインウィンドウにメニューとツールバーを追加するためのボタンが含まれています。ボタンは、それぞれのメニューまたはツールバー(使用するメニュータイプとメインウィンドウの下の現在のノードによる)を追加することが可能な場合にのみ使用できます。

- **メニューバーボタン** (): メニューバーを追加します。 [メニューバーノード](#) を参照してください。
- **ツールバーボタン** (): ツールバーを追加します。 [ツールバーノード](#) を参照してください。
- **ファイルメニューボタン** (): ファイルメニューを追加します。 [ファイルメニューノード](#) を参照してください。
- **リボンタブボタン** (): リボンにリボンタブを追加します。 [リボンタブノード](#) を参照してください。
- **リボンセクションボタン** (): リボンにリボンセクションを追加します。 [リボンセクションノード](#) を参照してください。
- **メニューボタン** (): メニューを追加します。 [メニューノード](#) を参照してください。
- **項目ボタン** (): メニュー項目を追加します。 [項目ノード](#) を参照してください。
- **項目切替えボタン** (): 項目切替えを追加します。 [項目切替えノード](#) を参照してください。
- **セパレーターボタン** (): メニュー内の関連するメニュー項目のグループを区切るためのセパレーターを追加します。 [セパレーターノード](#) を参照してください。




テストセクション

このセクションには、アプリケーションをテストするための以下のツールが含まれています。

- **アプリケーションをテストのボタン** (): 別のウィンドウでアプリケーションを起動してテストすることができます。 [アプリケーションのテスト](#) を参照してください。
- **変更を適用のボタン** (): 実行中のアプリケーションにコード変更をコンパイルして適用します(いわゆるホットコードスワップ)。 [実行中のアプリケーションへの変更の適用](#) を参照してください。
- **ウェブブラウザでテストのボタン** (): ウェブブラウザでアプリケーションをテストします。 [アプリケーションのテスト](#) を参照してください。

ビューセクション

ビューセクションには、アプリケーションビルダーのデスクトップウィンドウでビューを再配置する以下のボタンがあります。

- **タイル表示** () と **移動先** () ボタンを使用して、アプリケーションビルダーのウィンドウを再配置します。
- **デスクトップリセットボタン** (): デスクトップのレイアウトをデフォルトの状態にリセットします。

コンポーネントのカット、コピー、複製、削除、およびペースト

フォーム、メソッド、宣言など、アプリケーションビルダーツリーに追加するノードは、カット、コピー、複製、削除、およびペーストが可能です(アプリケーション間のコピーについては、以下のセクションを参照してください)。クイックアクセスツールバーのボタンを使用するか、ノードを右クリックして、次のいずれかのオプションを選択します。

- **カット** (ショートカットがサポートされている場合は **Ctrl + X**)
- **コピー**
- **複製** **Ctrl + Shift + D**
- **削除** **Del**

ノードをカットまたはコピーする場合、**宣言ノード**のコンテキストメニューの**選択リストをペースト**など、親ノードのコンテキストメニューを使用してノードを貼り付けることができます。

アプリケーション間のアプリケーションコンポーネントのコピー

アプリケーションの多くの部分をコピーして、別のアプリに貼り付けることができます。例えば、一つのフォームまたはいくつかのフォームオブジェクトをコピーした場合、COMSOL マルチフィジックスの二つの実行中のインスタンス間、または新しいアプリケーションを開いた後で一つの実行中のインスタンスに貼り付けることができます。メソッド、ユーティリティメソッド、外部ライブラリ、ファイル宣言、選択リスト宣言、メニュー項目、メニューディバイダ、メニュー、リボンセクション、およびリボンタブをコピーすることもできます。コピー操作は、アプリケーション内のさまざまなオブジェクトへの参照を処理することが可能です。アプリケーション間でコピー＆ペーストする場合、メソッドやファイルへの参照に関する以下の情報が役立ちます。

- コピー＆ペースト操作の対象には、ローカルメソッドも含まれています。但し、メソッド内のコードは変更されません。
- 必要に応じて、貼り付け中に全ての参照画像がコピーされ追加されます。同じ名前の既存の画像がある場合には、コピーされるバージョンの代わりに使用されます。
- ファイルやサウンドはコピーされません。

貼り付け操作に問題がある場合は、メッセージウィンドウが表示されます。そこで、貼り付け操作をキャンセルすることができます。

メインウィンドウブランチ

メインウィンドウノードから、以下のノードを追加することができます。

- **メニューバーとツールバー:** メインウィンドウノードの**設定**ウィンドウの**メニュータイプ**リストから**メニューバー**を選択した場合。
- **ファイルメニュー:** メインウィンドウノードの**設定**ウィンドウの**メニュータイプ**リストから**リボン**を選択した場合。**リボン**ノードは常に使用可能で、**リボンタブ**と**リボンセクション**ノードを追加することができます。

メインウィンドウノード

メインウィンドウノード(□)はアプリケーションのメインウィンドウを表し、ユーザーインターフェースの最上位のノードです。これには、ウィンドウのレイアウト、メインメニューの設定、オプションのリボンの設定が含まれています。

メインウィンドウの**設定**ウィンドウには、以下のセクションがあります。

一般

タイトルのフィールドには、アプリケーションのタイトルが表示されます。タイトルは、メインウィンドウの上部にあるテキストです。デフォルトでは、アプリケーションの作成に使用されたモデルのタイトルと同じです。**タイトルでファイル名表示**のチェックボックスがデフォルトで選択されています。これをクリアすると、アプリケーションのファイル名がタイトルから除外されます。

デフォルトの COMSOL アイコン以外のアプリケーションのアイコンを追加するには、**画像ライブラリ**内の全ての画像を含む**アイコン**リストから使用する画像ファイルを指定するか、**ライブラリに対する画像を追加してここで使用**のボタン(+)をクリックしてファイルシステム上で使用する画像を検索します。そこで選択した画像は、**画像ライブラリ**の一部となり、アプリケーションのメインウィンドウのアイコンとして設定されます。**デフォルト**設定では cube.png をロードします。**エクスポート**ボタン(→)をクリックすると、画像がファイルシステムにエクスポートされます(例えば、別のアプリケーションで使用する場合)。

メニュータイプのリストから、アプリケーションのメニューのタイプを選択します。

- アプリケーションの上部でメインメニューを使用するには、**メニューバー**(デフォルト)を選択します。**メインウィンドウ**のブランチには**メニューバー**ノードと**ツールバー**ノードがあり、そこにメニュー項目、アクション、セパレーターを含むメニューとサブメニューを追加することができます。
- アプリケーションの上部にある Windows スタイルのリボンツールバーを使用するには、**リボン**を選択します。**メインウィンドウ**の下に**ファイルメニュー**ノード(例えば、**終了**コマンド)と**リボン**ノードを追加し、**リボンタブ**ノードを追加することができます。**リボンタブ**ノードの下では、**リボンセクション**ノードを追加して、メニュー項目、アクション、セパレーターを含むメニューとサブメニューを追加することができます。

ステータスバーのリストは、ステータスバーへの表示を制御します。実行時に**進捗**または**なし**を選択します。(デフォルトでは、アプリケーションの右下隅に COMSOL の進捗バーが追加されています。カスタム進捗バーを追加するには、**進捗バー**を参照してください。)

メインフォーム

このセクションには、メインウィンドウに表示されるフォームへの参照が含まれています。**フォーム**のリストから、表示するフォームを選択します。

サイズ

初期サイズのリストから、アプリケーションウィンドウの初期サイズを設定する方法を選択します。

- **最大化**を選択すると、アプリケーションウィンドウがスクリーンいっぱいに最大化された状態で開かれます。

- **メインフォームのサイズを使用** (デフォルト) を選択すると、アプリケーションウィンドウのサイズがメインフォームのサイズに合わせられます。このフォームとは、上記の**メインフォーム**のセクションで選択したフォームのことです。
- **マニュアル**を選択した場合は、表示される**幅** (デフォルト: 1280 ピクセル) と**高さ** (デフォルト: 800 ピクセル) のフィールドにアプリケーションウィンドウの初期サイズを指定します。

詳細ダイアログ

このセクションでは、詳細ダイアログボックスの内容と、そのダイアログボックスへのリンクの配置を指定できます。


配置のリストから、リンクの配置として以下のオプションから選択します。

- **自動** (デフォルト)。これにより、詳細ダイアログボックスへのリンクが次の場所に表示されます。
 - ファイルメニュー: **メニュータイプ**のリストが**リボン**に設定され、**ファイルメニュー**が追加されている場合。
 - ツールバー: **メニュータイプ**のリストが**メニューバー**に設定され、ツールバーはあるがメニューはない場合。
 - それ以外の場合は、右下隅。
- **ファイルメニュー**または**メニューバー** (**メニュータイプ**リストの設定に依る)。
- **リボン**または**ツールバー** (**メニュータイプ**リストの設定に依る)。
- **右下コーナー** (アプリケーションのメインフォームの右下コーナーにある**詳細ハイパーリンク**)。アプリケーションにフォームがない場合には、代わりに自動配置が使用されます。


標準情報 (ロゴタイプ、バージョン番号、使用されている製品に関する情報) を含めたくない場合は、**COMSOL 情報表示**のチェックボックスをオフにします。

カスタムテキストのフィールドに、詳細ダイアログボックスに含めたいテキストを追加します。このテキストは、ライセンス、特許、および商標情報の上に表示されます。そこにウェブアドレスを含めると、詳細ダイアログボックスにはハイパーリンクとして表示されます。ウェブアドレスは、http://または www で始まる有効な URL である必要があります。


メニューバーノード

メニューバーのノード () は、アプリケーションのメインウィンドウのトップレベルのメインメニューを表します。各メニューの子ノードは、メインウィンドウのタイトルバー上のメニューを表します。**メニューバー**ノードを右クリックして、**メニュー**ノードをメインメニューに追加します。メニューバーに複数のメニューを作成した場合、一番左のメニューには**アプリケーションを閉じる**の項目がデフォルトで含まれます。

ツールバーノード

ツールバーノード () は、アプリケーションのメインウィンドウの上部 (メインメニューの下側) にあるツールバーを表します。ここには、ドロップダウンツールバーのメニューを表す**メニュー**の子ノード、それらに接続されたアクションを受け持つツールバーのボタンを表す**項目**ノード、および関連するツールバーボタンのグループを区切る**セパレーター**ノードを追加することができます。他のノードをツールバーに追加するには、このノードを右クリックします。

メニューノード

メニューノード () は、親メニューの下に別のメニューレベルを追加します。他のメニュータイプも可能です。**メニュー**ノードを右クリックして、別の**メニュー**ノードをサブメニューとして追加することができます。また、右クリックして、アクションを持つメニュー項目である**項目**ノードを、またはメニューにセパレーターを挿入する**セパレーター**ノードを追加することもできます (**項目ノード**と**セパレーターノード**を参照)。**メニュー**ノード (および**項目**と**セパレーター**ノード) を追加レベルとしてさらに追加し、この追加レベルのサブメニューを作成することができます。メニューオブジェクトの名前は、**名前**フィールドに入力します。

タイトルフィールドには、新しいメニューレベルのタイトルを指定します。

ファイルメニューノード、ツールバーノード、またはリボンセクションノードの直下にあるメニューノードでは、アイコンをメニューに追加することができます。これを行うには、画像ライブラリ内の全ての画像を含んでいるアイコンリストから使用する画像ファイルを指定するか、ライブラリに対する画像を追加してここで使用するボタン(+)をクリックし、使用する画像をファイルシステムで検索します。この画像は画像ライブラリの一部として取り込まれ、このメニューのアイコンとして選択されます。アイコンを使用しない場合は、アイコンリストからなしを選択します。エクスポートボタン(📁)をクリックして、画像をファイルシステムにエクスポートすることができます(例えば、別のアプリケーションで使用する場合)。

リボンセクションの直下にあるメニューノードでは、サイズのリストから大(デフォルト)または小を選択することもできます。このサイズによって、リボン内のボタンのサイズが制御されます。

ファイルメニューノード

ファイルメニューノード(📁)は、リボンツールバーを持つアプリケーションの左上隅にあるファイルメニューを表します。ファイルメニューノードを右クリックして、メニューノードをサブメニューとして追加することができます。また、右クリックして、アクションを持つメニュー項目である項目ノードを、または関連するメニュー項目のグループを区切るセパレーターを追加することもできます。ファイルメニューには、アプリケーションの保存や終了などの標準的な項目を含めておく必要があります。

リボンノード

リボンノード(📁)は、アプリケーションのメインウィンドウの上部にあるリボンツールバーを表します。リボンツールバーには、リボンタブを表す子ノードのリボンタブを追加することができます。リボンタブには、ドロップダウンメニューとボタンを持ったリボンセクションを追加することができます。このリボンノードを右クリックして、リボンにリボンタブノードを追加します。

リボンタブノード

リボンタブノード(📁)は、リボンタブをアプリケーションウィンドウの上部にあるリボンに追加します。リボンタブには、リボンセクションのサブノードにメニューを追加して含めることによって、そこからさらに展開される項目を含んだセクションを持たせることができます。そのようなリボンセクションを追加するには、このノードを右クリックしてリボンセクションを選択します。リボンタブオブジェクトの名前は、名前フィールドに入力します。

リボンタブのタイトルを指定します。

リボンタブにアイコンを追加するには、画像ライブラリ内の全ての画像を含んでいるアイコンリストから使用する画像ファイルを指定するか、ライブラリに対する画像を追加してここで使用するボタン(+)をクリックし、使用する画像をファイルシステムで検索します。この画像は画像ライブラリの一部として取り込まれ、このメニューのアイコンとして選択されます。アイコンを使用しない場合は、アイコンリストからなしを選択します。エクスポートボタン(📁)をクリックして、画像をファイルシステムにエクスポートすることができます(例えば、別のアプリケーションで使用する場合)。

リボンセクションノード

リボンセクションノード(📁)は、リボンにセクションを追加します。リボンセクションノードを右クリックして、メニューノード、直接的なアクションを持つボタンとしての項目ノード、リボンセクションのコンテンツを定義するセパレーターノードを追加することができます。リボンセクションオブジェクトの名前は、名前フィールドに入力します。



リボンのメニューボタンでサブメニューを使用すると、そのメニューにヘッダーが表示されます。このサブメニューを利用して、標準のメニューバーのようにいくつかの項目をグループ化して表示することができます。サブメニューの下に高い優先度のメニュー項目を追加しないのであれば、この方法はリボンに最適です。

リボンセクションの**タイトル**を指定します。

リボンセクションにアイコンを追加するには、画像ライブラリ内の全ての画像を含んでいる**アイコン**リストから使用する画像ファイルを指定するか、**ライブラリに対する画像を追加してここで使用**のボタン(+)をクリックし、使用する画像をファイルシステムで検索します。この画像は**画像**ライブラリの一部として取り込まれ、このリボンセクションのアイコンとして選択されます。

アイコンを使用しない場合は、**アイコン**リストから**なし**を選択します。**エクスポート**ボタン(📁)をクリックして、画像をファイルシステムにエクスポートすることができます(例えば、別のアプリケーションで使用する場合)。

項目ノード

項目ノード(📁)は、メソッドを親メニューのメニュー項目として、またはツールバーのボタンとして実行するメニューオプションです。**メニュー**ノードの下に、あるいはフォームの**ツールバー**オブジェクトに**項目ノード**を追加することができます。**項目**オブジェクトの名前は、**名前**フィールドに入力します。

テキストフィールドに、メニュー項目に表示するテキストを入力します。メニュー項目にアイコンを追加するには、画像ライブラリ内の全ての画像を含んでいる**アイコン**リストから使用する画像ファイルを指定するか、**ライブラリに対する画像を追加してここで使用**のボタン(+)をクリックし、使用する画像をファイルシステムで検索します。この画像は**画像**ライブラリの一部として取り込まれ、このメニュー項目のアイコンとして選択されます。アイコンを使用しない場合は、**アイコン**リストから**なし**を選択します。**エクスポート**ボタン(📁)をクリックして、画像をファイルシステムにエクスポートすることができます(例えば、別のアプリケーションで使用する場合)。

リボンセクションの直下にある**項目ノード**では、**サイズ**のリストから**大**(デフォルト)または**小**を選択することもできます。このサイズによって、リボン内のボタンのサイズが制御されます。

リボンセクションの直下にある**項目ノード**の場合は、**ツールチップ**のフィールドへの入力によって、ツールチップを追加することもできます。ツールチップは、リボンボタンの上にポインタを置くと表示される説明テキストです。

キーボードショートカットのフィールドへの入力によって、項目のショートカットを定義することができます。キーボードショートカットを追加するには、**キーボードショートカット**のフィールドをアクティブにしてから、キーボード上でそのキーボードショートカットを入力操作します。

キーボードショートカットでは、普通の文字だけでなく、例えば CTRL + SHIFT + D などの修飾キーを使用する必要があります。ショートカットには、Ctrl キー(CTRL)、Alt キー(ALT)、Shift キー(SHIFT)を使用できます。Ctrl キーは OS X のコマンドとして解釈されることに注意してください。ショートカットでは次のキーを使用しないでください。

- Backspace: ショートカットを消去するために使用できるため
- Delete: ショートカットを消去するために使用できるため
- Escape
- Alt 単独使用: ファイルメニューのショートカットとの競合を避けるため



他のキーボードショートカットをオーバーライドすることが可能ですので、使用するショートカットキーの組み合わせを選択する際には注意してください。

また、**設定**ウィンドウには、以下のセクションがあります。

実行コマンド選択

このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーがフィルタリング表示されたツリーがあります。ノードはコマンドをサポートしているか、または子ノードを持つかのいずれかです。一つまたは複数のコマンドをサポートするノードを選択すると、ツリーの下ツールバー内にある対応するコマンドツールバーのボタンが有効になります。また、ノードを右クリックして、そのノードの使用可能なコマンドのリストを取得することもできます。選択されたノードのコマンドをクリック(または Enter キーを押して、**実行**、**プロット**、または**値設定**などのデフォルトコマンドでコマンドを追加)する

と、設定ウィンドウの**実行コマンド選択**セクションのテーブルの最後の行にそのコマンドとノードが表示されます。このテーブルには実行される全てのノードが含まれており、このテーブルの下にあるツールバーを使用してコマンドを削除したり移動することができます。

モデルブランチでは、**パラメータ**ノードの下にあるパラメータのように、何らかの種類のデータ値を表すノードの全ては、**値設定**コマンドをサポートしています。テーブルに**値設定**コマンドが追加されると、その3列目の**引数**が有効になります。その列に、設定する値を入力します。配列を表すデータの場合には、中括弧とカンマを使って配列要素を入力します。例えば、{1,2,3}を入力して、値 1,2,3 の3要素配列を設定します。配列と行列の入力方法の詳細については、**配列 1D 文字列ノード**を参照してください。**補間**の関数ノードの下にある**ファイル名**ノードなど、ファイルのインポートを表すノードの場合は、**インポートファイル**のコマンドが使用できます。また、全ての**ビュー**ノードに対して**プロット**コマンドを追加して、グラフィクスオブジェクトの名前を引数として指定することもできます。

ツリーには、**モデル**ブランチに加えて、アプリケーションツリーからのいくつかのブランチが含まれています。



- **フォーム**ブランチ: **フォーム**ノードは、フォームをアプリケーションのメインフォームとして設定する(つまり、アプリケーションウィンドウのコンテンツがこのフォームになる) **表示**のコマンドと、フォームを別のダイアログウィンドウとして開く**ダイアログ**として**表示**のコマンドをサポートしています。
- **GUI** コマンドブランチ: このブランチの下にあるコマンドは、三つのサブカテゴリにグループ化されています。
 - **ファイル**コマンド: これには、**アプリケーションを保存**(現在の名前ですべてのアプリケーションを保存する); **名前を付けてアプリケーションを保存**(ユーザーが適切な場所にアプリケーションを保存できるようにするファイルブラウザダイアログを開く); **サーバー上でアプリケーションを保存**; **サーバー上でアプリケーションを別名で保存**; **ファイルを開く**(引数列の有効な URI パスを使用して指定されたアプリケーションファイルリソースを開く); **名前を付けてファイルを保存**(同様に、ユーザーが引数列で指定した名前ですべてのファイルを保存できるようにする); および**アプリケーションを終了**(実行中のアプリケーションを閉じる)が含まれています。アプリケーションが COMSOL サーバー上で実行されている場合、**サーバー上でアプリケーションを保存**と**サーバー上でアプリケーションを別名で保存**のコマンドは、現在の状態を COMSOL サーバーアプリケーションライブラリに新しいアプリケーションとして保存します。
 - **グラフィックス**コマンド: これには、**画面にわたってズーム**、**現在のビューをリセット**、**シーンライト**、**透明度**、**印刷**、**全選択**、および**選択対象消去**のコマンドがあります。全てのグラフィックコマンドで、コマンドを適用したいグラフィクスオブジェクトの名前を引数として追加します。
 - **モデル**コマンド: これには、**全解除**と**全メッシュ消去**のコマンドがあります。

上記のいずれかのノードをダブルクリックするか右クリックして、**実行**コマンドを追加します。


- **宣言**ブランチ: このブランチには、**アプリケーションビルダー**ウィンドウの**宣言**ブランチの下で追加されている変数宣言がタイプ別にグループ化されています。パラメータと同様に、**値設定**コマンドがサポートされています。
- **メソッド**ブランチ: ここでは、**メソッド**ノードが**実行**コマンドをサポートしています。
- **ライブラリ**ブランチ: ここでは、**サウンド**の下で、コマンドシーケンスで再生するサウンドファイルを選択できます。



ツリーの下にあるコマンドの一つをクリックすると、そのコマンドが下のリストの**コマンド**の列に表示されます。また、**アイコン**の列と**引数**の列があります。引数の列には、コマンドで使用される任意の引数を入力できます。そこには、コマンドが期待する引数のタイプを示すツールチップが表示されます。例えば、downloadtoclient コマンドの場合、引数は embedded:///myfile.txt のようなファイル名とそのパスの両方です。このような引数を定義するのに役立つ別のダイアログボックスがあります。詳細については、**ファイルスキーム**と**ファイル処理**を参照してください。

コマンドリストの下にある**ローカルメソッド作成**のボタン(+)をクリックすると、テーブル内のコマンドリスト全体を同等のコードを含む新しいローカルメソッドに変換し、そのローカルの onClick メソッドのエディターウィンドウを開きます。ローカルメソッドを作成していないときに**新しいメソッドに変換**のツールバーボタン(🔄)をクリックすると、テーブル内のコマンドのリスト全体を同等のコードを含む新しい**メソッド**ノードに変換します。この操作をすると、コマンドのリストには作成された**メソッド**ノードが1回だけ**実行**される操作しか含まれなくなります。**コマンド**の下でメソッドを選択するか、リストにメソッドが一つしかない場合は、**メソッド**へのボタン(🔍)をクリックすると、そのメソッドのエディターウィンドウに移動します。**引数編集**のボタン(🔍)の詳細については、**宣言**と**コマンドシーケンス**での**初期値**と**引数の編集**を参照してください。**上へ移動**(↑)、**下へ移動**

()、**削除**()のツールバーボタンを使用して、リストからコマンドを整理したり削除します(削除の場合はローカルメソッドも削除されます。)

項目切替えノード

項目切替えノード()は、親メニューのメニュー項目として、またはツールバーのボタンとして状態を切り替えるメニューオプションです。例えば、項目切替えは、例えば、フォームコレクションのタブとタイトルの表示を切り替えるのに便利です。**メニューノード**の下に、あるいはフォームの**ツールバー**オブジェクトに**項目切替えノード**を追加することができます。項目切替えオブジェクトの名前は、**名前**フィールドに入力します。

テキストフィールドに、項目切替えメニューに表示するテキストを入力します。項目切替えメニューにアイコンを追加するには、**画像ライブラリ**内の全ての画像を含んでいる**アイコン**リストから使用する画像ファイルを指定するか、**ライブラリ**に対する**画像を追加してここで使用**のボタン()をクリックし、使用する画像をファイルシステムで検索します。この画像は**画像ライブラリ**の一部として取り込まれ、このメニュー項目のアイコンとして選択されます。アイコンを使用しない場合は、**アイコン**リストから**なし**を選択します。**エクスポート**ボタン()をクリックして、画像をファイルシステムにエクスポートすることができます(例えば、別のアプリケーションで使用する場合)。

リボンセクションの直下にある**項目切替えノード**では、**サイズ**のリストから**大**(デフォルト)]または**小**を選択することもできます。このサイズによって、リボン内のボタンのサイズが制御されます。

リボンセクションの直下にある**項目切替えノード**の場合は、**ツールチップ**のフィールドへの入力によって、ツールチップを追加することもできます。ツールチップは、リボンボタンの上にポインタを置くと表示される説明テキストです。

キーボードショートカットのフィールドへの入力によって、項目のショートカットを定義することができます。キーボードショートカットを追加するには、**キーボードショートカット**のフィールドをアクティブにしてから、キーボード上でそのキーボードショートカットを入力操作します。

キーボードショートカットでは、普通の文字だけでなく、例えば CTRL + SHIFT + D などの修飾キーを使用する必要があります。ショートカットには、Ctrl キー (CTRL)、Alt キー (ALT)、Shift キー (SHIFT) を使用できます。Ctrl キーは OS X のコマンドとして解釈されることに注意してください。ショートカットでは次のキーを使用しないでください。



- Backspace: ショートカットを消去するために使用できるため
- Delete: ショートカットを消去するために使用できるため
- Escape
- Alt 単独使用: ファイルメニューのショートカットとの競合を避けるため



他のキーボードショートカットをオーバーライドすることが可能ですので、使用するショートカットキーの組み合わせを選択する際には注意してください。

また、**設定**ウィンドウには、以下のセクションがあります。


ソース

このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーがフィルタリング表示されたツリーがあります。ノードは何らかの種類のデータを表すか、または子ノードを持つかのいずれかです。項目切替えの場合、**宣言**の下にある文字列とブーリアン変数宣言(項目切替えのオン/オフの状態を表す)はソースとして使用できます。データを表すノードを選択すると、ツリーの下にある**ソースとして使用**のツールバーボタンが有効になります。ノードをクリックするか、Enter キーを押すか、ダブルクリックするか、ノードを右クリックして**ソースとして使用**を選択して、選択ソースとして追加します。**ソース**セクションのヘッダーにある**作成**ボタン()をクリックして、項目切替えの新規変数宣言を作成してソースとして使用することもできます。後者の場合、**変数を作成および使用**のダイアログボックスが開き、ソースのデータタイプ(該当する場合)とその名前を選択することができます。名前は、既存の変数宣言と競合することはできません。ツリーの下の**ノード編集**のツールバーボタン()をクリックすると、対応するノードに移動します。

ソースとしてノードを選択すると、**選択ソース**に選択されたソースとしてそのノードが表示されます。



複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースのデフォルト値は期待通りとは限りません。あるフォームオブジェクトのデフォルト値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。

通常、**パラメータノード**の下で使用可能なパラメータ、**変数ノード**の下の変数、およびアプリケーションビルダーのアプリケーションツリー（**イベントブランチ**の下）にある**宣言ブランチ**の下に定義されているデータノードのみが表示されます。使用可能なデータノードのリストを拡張するには、**ソースセクション**のヘッダーにある**モデルビルダーへスイッチしデータアクセスをアクティブ**のボタン（)をクリックし、自動的にモデルビルダーに移動します。次に、アクセスしたいデータがある**モデルビルダーブランチ**のノードを選択します。そのボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることができる設定の横に**データソース選択**のチェックボックス（緑色の四角形）が表示されます。このチェックボックスをオンにすると、そのデータをイベントの使用可能なソースノードとして含めることができます。

データソースに対する制限は、アプリケーションのユーザーインターフェースに渡されます。例えば、構造力学の**境界荷重ノード**の**荷重タイプ**リストでは、三つの値しか使用できません。このデータをソースとして使用するフォームオブジェクトは、それらの値のサブセットのみをサポートできます。

項目切替えオブジェクトの場合、初期値を選択 (on) するか、クリア (off) するかを指定することもできます。初期値のリストから、**カスタム値** (デフォルト) または **データソース参照** を選択します。**カスタム値** の場合は、**初期状態** のリストから **選択** または **クリア** を選択します。

実行コマンド選択

このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーがフィルタリング表示されたツリーがあります。ノードはコマンドをサポートしているか、または子ノードを持つかのいずれかです。一つまたは複数のコマンドをサポートするノードを選択すると、ツリーの下ツールバー内にある対応するコマンドツールバーのボタンが有効になります。また、ノードを右クリックして、そのノードの使用可能なコマンドのリストを取得することもできます。選択されたノードのコマンドをクリック（または Enter キーを押して、**実行**、**プロット**、または**値設定**などのデフォルトコマンドでコマンドを追加）すると、**設定**ウィンドウの**実行コマンド選択**セクションのテーブルの最後の行にそのコマンドとノードが表示されます。このテーブルには実行される全てのノードが含まれており、このテーブルの下にあるツールバーを使用してコマンドを削除したり移動することができます。

モデルブランチでは、**パラメータノード**の下にあるパラメータのように、何らかの種類のデータ値を表すノードの全ては、**値設定**コマンドをサポートしています。テーブルに**値設定**コマンドが追加されると、その 3 列目の**引数**が有効になります。その列に、設定する値を入力します。列を表すデータの場合には、中括弧とカンマを使って配列要素を入力します。例えば、{1,2,3}を入力して、値 1,2,3 の 3 要素配列を設定します。配列と行列の入力方法の詳細については、**配列 1D 文字列ノード**を参照してください。**補間の関数ノード**の下にある**ファイル名**ノードなど、ファイルのインポートを表すノードの場合は、**インポートファイル**のコマンドが使用できます。また、全ての**ビューノード**に対して**プロット**コマンドを追加して、グラフィクスオブジェクトの名前を引数として指定することもできます。

ツリーには、**モデルブランチ**に加えて、アプリケーションツリーからのいくつかのブランチが含まれています。

- **フォームブランチ**: **フォームノード**は、フォームをアプリケーションのメインフォームとして設定する（つまり、アプリケーションウィンドウのコンテンツがこのフォームになる）**表示**のコマンドと、フォームを別のダイアログウィンドウとして開く**ダイアログ**として**表示**のコマンドをサポートしています。
- **GUI コマンドブランチ**: このブランチの下にあるコマンドは、三つのサブカテゴリにグループ化されています。
 - **ファイルコマンド**: これには、**アプリケーションを保存**（現在の名前でアプリケーションを保存する）；**名前を付けてアプリケーションを保存**（ユーザーが適切な場所にアプリケーションを保存できるようにするファイルブラウザダイアログを開く）；**サーバー上でアプリケーションを保存**；**サーバー上でアプリケーションを別名で保存**；**ファイルを開く**（引数列の有効な URI パスを使用して指定されたアプリケーションファイルリソースを開く）；**名前を付けてファイルを保存**（同様に、ユーザーが引数列で指定した名前でファイルを保存できるようにする）；および**アプリケーションを終了**（実行中の

アプリケーションを閉じる)が含まれています。アプリケーションが COMSOL サーバー上で実行されている場合、**サーバー上でアプリケーションを保存**と**サーバー上でアプリケーションを別名で保存**のコマンドは、現在の状態を COMSOL サーバーアプリケーションライブラリに新しいアプリケーションとして保存します。

- **グラフィックスコマンド**: これには、**画面にわたってズーム**、**現在のビューをリセット**、**シーンライト**、**透明度**、**印刷**、**全選択**、および**選択対象消去**のコマンドがあります。全てのグラフィックコマンドで、コマンドを適用したいグラフィックスオブジェクトの名前を引数として追加します。
- **モデルコマンド**: これには、**全解除**と**全メッシュ消去**のコマンドがあります。

上記のいずれかのノードをダブルクリックするか右クリックして、**実行コマンド**を追加します。

- **宣言ブランチ**: このブランチには、**アプリケーションビルダー**ウィンドウの**宣言ブランチ**の下で追加されている変数宣言がタイプ別にグループ化されています。パラメータと同様に、**値設定**コマンドがサポートされています。
- **メソッドブランチ**: ここでは、**メソッド**ノードが**実行コマンド**をサポートしています。
- **ライブラリブランチ**: ここでは、**サウンド**の下で、コマンドシーケンスで再生するサウンドファイルを選択できます。


ツリーの下にあるコマンドの一つをクリックすると、そのコマンドが下のリストの**コマンド**の列に表示されます。また、**アイコン**の列と**引数**の列があります。引数の列には、コマンドで使用される任意の引数を入力できます。そこには、コマンドが期待する引数のタイプを示すツールチップが表示されます。例えば、downloadtoclient コマンドの場合、引数は embedded:///myfile.txt のようなファイル名とそのパスの両方です。このような引数を定義するのに役立つ別のダイアログボックスがあります。詳細については、[ファイルスキームとファイル処理](#)を参照してください。

コマンドリストの下にある**ローカルメソッド作成**のボタン(+)をクリックすると、テーブル内のコマンドリスト全体を同等のコードを含む新しいローカルメソッドに変換し、そのローカルの onClick メソッドのエディターウィンドウを開きます。ローカルメソッドを作成していないときに**新しいメソッドに変換**のツールバーボタン(📄)をクリックすると、テーブル内のコマンドのリスト全体を同等のコードを含む新しい**メソッド**ノードに変換します。この操作をすると、コマンドのリストには作成された**メソッド**ノードが1回だけ**実行**される操作しか含まれなくなります。**コマンド**の下でメソッドを選択するか、リストにメソッドが一つしかない場合は、**メソッドへ**のボタン(📄)をクリックすると、そのメソッドのエディターウィンドウに移動します。**引数編集**のボタン(🔍)の詳細については、[宣言とコマンドシーケンスでの初期値と引数の編集](#)を参照してください。**上へ移動**(↑)、**下へ移動**(↓)、**削除**(✖)のツールバーボタンを使用して、リストからコマンドを整理したり削除します(削除の場合はローカルメソッドも削除されます)。


セパレーターノード

メニューノードの下に**セパレーター**ノード(☰)を追加することによって、親メニューにセパレーター(水平ラインまたはデバイダー)を追加し、関連するメニュー項目のグループを区切ることができます。また、フォーム内のツールバーオブジェクトにセパレーターを追加することもできます。

フォームブランチ

フォームブランチ()の下には、アプリケーションのユーザーインターフェースを定義する全てのフォームがあります。フォームをダイアログとして表示するコマンドを実行することにより、ユーザーインターフェースにフォームを含めることができます。これは、**ボタン**、**項目**、**イベント**、または**メソッド**ノードから行うことができます。フォーム、フォームオブジェクト、およびフォームを作成するために利用できるツールの詳細については、[フォームの操作](#)を参照してください。

フォームノード

フォームノード()は、サブノードとして追加する全てのフォームノードのプレースホルダーです。設定ウィンドウには、フォームの外観に関する以下の設定があります。

外観

このセクションの設定では、追加するフォームで使用するテキストを定義しますが、フォーム内の特定のスタイルを指定する設定を継承しないように選択することもできます。


テキスト色のリストから、**システム**(オペレーティングシステムで定義されているデフォルト)；予め定義された色の一つ；カラーパレットからカスタムテキストカラーを選択する**カスタム**のいずれかを選択します。



背景色のリストから、フォームの背景として使用する色を選択します。**システム**(オペレーティングシステムで定義されているデフォルト)；予め定義された色の一つ；カラーパレットからカスタムの背景色を選択する**カスタム**のいずれかです。

フォントのリストから、テキストのフォントを選択します。**システム**(オペレーティングシステムで定義されているデフォルト)、またはリスト内の別のフォントを選択します。利用者のローカル言語の文字をサポートするフォントを選択する必要があるかもしれません。必要に応じて、**フォントサイズ**のコンボボックスからフォントサイズを選択するか、ポイント単位で入力します。デフォルトでは、**システム**サイズのフォントが使用されます。デフォルトでは、フォームオブジェクトはこれらの設定からフォントとフォントサイズを継承します。

新しいフォームオブジェクトへ適用の下で、**ボールド**、**イタリック**、**アンダーライン**のいずれかのチェックボックスをオンにすると、フォームに追加するフォームオブジェクトに対してデフォルトで、ボールド体フォント、イタリック体、または下線付きの任意の組み合わせによってテキスト表示されます。

フォームノード(サブノード)

フォームノード()を追加して、アプリケーションの一般的なユーザーインターフェース領域である *form* を作成します。例えば、デスクトップウィンドウ、ダイアログウィンドウ、またはタブ付きペインのコンテンツを表すことができます。フォームは、別のノードによってユーザーインターフェースの一部として参照される必要がありますが、ユーザーがボタンをクリックした後でフォームをダイアログとして表示することもできます。フォームを編集またはテストするには、**フォームノード**のコンテキストメニューで次のオプションを使用します。

- **フォームノード**を右クリックして**編集**()を選択し、フォームウィンドウを開きます。そこでインタラクティブにフォームを作成してデザインすることができます([フォームの操作](#)を参照)。
- **フォームノード**を右クリックして**プレビューフォーム**()を選択し、フォームを検査可能な別のウィンドウでプレビューとして開きます。そこでフォームを確認することができます。

フォームオブジェクトの名前は、**名前**フィールドに入力します。

フォームの**タイトル**を指定します。デフォルトのタイトルは、アプリケーションの最初のフォームであれば**フォーム1**です。

サイズ

サイズプロパティは、フォームがダイアログボックスに表示されたときに適用されます。デフォルトでは、アプリケーションビルダはフォームの内容に基づいて初期サイズを自動的に決定します。**初期サイズ**のリストから**自動**(デフォルト)、または**マニュアル**を選択して幅フィールドと高さフィールドに初期サイズを指定します(デフォルト: 40 ピクセル)。

マージン

このセクションでは、必要に応じてフォームの**水平**と**垂直**の余白を調整することができます(デフォルト: 20 ピクセル)。

ダイアログ設定

例えば、ユーザーが OK ボタンまたは適用ボタン(キャンセルボタンで保留中の変更を取り消すことが可能)をクリックしたときにデータ変更を保存するには、**変更を保存**のリストから**リクエスト時**(デフォルト)を選択します。変更が行われたときに直ちにデータを保存する場合には、**すぐに**を選択します。この設定は、フォームがダイアログとして表示される場合に適用されます。**すぐに**の設定を使用して、例えばスライダーと入力フィールド間を直接接続するような動的なダイアログを作成することができます。

ダイアログボックスとして使用したときにフォームのアイコン(デフォルトの COMSOL アイコン以外)を追加するには、**画像ライブラリ**内の全ての画像を含んでいる**アイコン**リストから使用する画像ファイルを指定するか、**ライブラリに対する画像を追加してここで使用**のボタン(+)をクリックし、使用する画像をファイルシステムで検索します。この画像は**画像ライブラリ**の一部として取り込まれ、ダイアログボックスとして使用されるフォームのアイコンとして選択されます。**デフォルト**のオプションでは、cube.png がロードされます。**エクスポート**ボタン(📄)をクリックして、画像をファイルシステムにエクスポートすることができます(例えば、別のアプリケーションで使用する場合)。

サイズ変更のチェックボックスを選択すると、必要に応じてユーザーがダイアログのサイズを変更できるようになります。

セクション設定

展開可のチェックボックスは、デフォルトで選択されています。ユーザーがセクションを展開したり折り畳んだりしないようにしたい場合は、それをクリアしておきます。もう一つのオプションである**起動時に折り畳み**のチェックボックスを選択すると、セクションの状態が最初に折り畳まれた状態になります。

スケッチグリッド



スケッチグリッドセクションは、フォームのスケッチモードを選択した場合にのみ使用できます。

このセクションでは、スケッチモードで表示できるグリッドの設定とそのグリッドへのフォームオブジェクトのスナップについて説明します([グリッド線の表示とグリッドへのスナップ](#)を参照)。

列幅(デフォルト: 100 ピクセル)と**行高さ**(デフォルト: 20 ピクセル)のフィールドに値を入力して、グリッドサイズを指定します。

グリッドラインを左余白および上余白に揃えるには、**マージンに対するアライングリッド**のチェックボックスをオンにします。

スナップゾーンのスライダーでは、グリッドにスナップするためにフォームオブジェクトのサイズを変更する場合に必要な正確さの程度を制御します。デフォルトでは、スナップゾーンは最大値に設定されているため、オブジェクトはグリッドにスナップするようにすばやくサイズ変更されます。必要に応じてスナップゾーンを小さくするには、スライダーを**大**から**小**に移動します。

グリッドに対してのみスナップのチェックボックスをオンにすると、フォームオブジェクトのサイズ変更がグリッドにのみスナップされ、例えば他のフォームオブジェクトの境界にはスナップされません。

含まれているフォームオブジェクトのためのグリッドレイアウト



含まれているフォームオブジェクトのためのグリッドレイアウトセクションは、フォームのグリッドモードを選択した場合にのみ使用できます。

設定ウィンドウのこのセクションには、グリッド内の列用と行用の二つのテーブルがあります。各テーブルの列および行という名前の列には、それぞれ列番号と行番号がつけられています。そのそれぞれの番号は、フォーム上の左から1から始まり、上から1から始まる対応となっています。各行と各列がフォーム内のスペースをどのように埋めるかを制御することができます。各テーブルには、幅(列)または高さ(行)という名前の列があり、そこにはフィット(デフォルト)、自動拡大縮小、および固定のオプションを含むリストがあります。

- **フィット**オプションを使用すると、列または行に対して、そこに格納されるオブジェクトが収まるのに必要なスペースを使用させるようになります。この設定による列と行のサイズは自動変更されません。
- **自動拡大縮小**オプションを使用すると、例えば、アプリケーションのユーザーがアプリケーションウィンドウのコーナーをドラッグしてフォームのサイズを大きくしたときに、フォーム内の使用可能なスペースを使用して、グリッドの列または行を拡大させることができます。
- **固定**オプションでは、グリッドレイアウトの列または行の幅または高さが、テーブルの3番目の**サイズ**という名前の列に指定されます。他のオプションが選択された場合には、**サイズ**の列は適用されず、N/Aが表示されます。ピクセル単位で追加された幅または高さは、フォームの編集ウィンドウの列ヘッダーまたは行ヘッダーに表示されます。この設定の列と行のサイズは自動変更されません。

列を継承のリストからは、列の設定を継承するフォームオブジェクトを選択します。デフォルトはなしです。つまり、列の設定は継承されません。

外観

このセクションでは、フォームのテキストと背景の外観を制御することができます。

テキスト色のリストから、**システム**(オペレーティングシステムで定義されているデフォルト); 予め定義された色の一つ; カラーパレットからカスタムテキストカラーを選択する**カスタム**のいずれかを選択します。

背景色のリストから、フォームの背景として使用する色を選択します。**システム**(オペレーティングシステムで定義されているデフォルト); 予め定義された色の一つ; カラーパレットからカスタムの背景色を選択する**カスタム**のいずれかです。

背景画像のリストからは、フォーム内でそのような背景の画像を使用したい場合に選択します。デフォルトでは、背景画像を持たないなしが選択されています。画像を画像ライブラリに追加して背景画像として使用するには、**ライブラリに対する画像を追加してここで使用**のボタン(+)をクリックします。**エクスポート**のボタン(📄)をクリックすると、背景画像をPNGファイルに保存することができます。

背景画像を使用する場合は、**画像位置およびサイズ**で次のアライメント設定を指定することもできます。

- **水平アライメント**のリストから、**左**、**中心**、**右**、**全幅**、**繰り返し**を選択します。
- **垂直アライメント**のリストから、**上**、**中間**、**下**、**全幅**、**繰り返し**を選択します。

全幅を選択すると、水平または垂直方向(または両方向)にフォームウィンドウの全幅となるように、背景画像が自動的に引き伸ばされます。画像を水平方向、垂直方向、または両方向に繰り返して表示(タイル表示)するには、**繰り返し**を選択します。

イベント

このセクションでは、フォームをロードしたり閉じたりするときにトリガされるイベントにローカルメソッドを接続することができます。このメソッドでは、フォームをロードしたり閉じたりするときに、例えば初期化やクリーンアップを実行することができます。

ロード時と**クローズ時**のリストのデフォルトは、なしです。つまり、フォームのロードやクローズのイベントがトリガされたときに、メソッドは実行されません。**ロード時**や**クローズ時**のローカルメソッドを作成するには、**ローカルメソッド作成**のボタン(+)をクリックします。すると、それに対応するリストの選択が**ローカルメソッド**に変更されます。**ソース**へのボタン(📄)をクリックすると、メソッドエディターウィンドウが開きます。そこでは、**ロード時**または**クローズ時**のメソッドのコードを作成または変更することができます。**ローカルメソッドを削除**のボタン(X)をクリックすると、ローカルメソッドが削除されます。

イベントブランチ

イベントとは、アプリケーションが一つまたは複数のアクションを実行する必要があることを通知するアクティビティです（例えば、ボタンをクリックする、キーボードショートカットを入力する、フォームをロードする、変数の値を変更するなど）。各アクションは、前述したタイプのコマンドシーケンスであってもよく、実行中のメソッドも含まれます。イベントブランチ(📁)には、グローバルパラメータや文字列データなど、さまざまなデータエンティティの変更を検出する全てのイベントが含まれます。イベントノードを右クリックしてイベント(📌)を選択し、イベントをアプリケーションに追加します。メインイベントノードには、アプリケーションの開始と終了に関連するイベントのオプションが含まれています。

イベントノード

イベントノード(📁)は、その下でイベントノードを追加してイベントを定義できる最上位のノードです。このノードの設定ウィンドウでは、以下のイベントを追加することができます。

イベント

スタートアップ時のリストから、アプリケーションウィンドウが開く前に実行されるメソッドを選択します。従って、例えば、プロットや他のユーザーインターフェース関連のコードを呼び出すことはできません（そのようなメソッドの場合は、フォームのロード時のイベントを使用します）。アプリケーションのためのいくつかの特別な設定が可能です。このイベントにローカルメソッドを追加するには、**ローカルメソッド作成**のボタン(+)をクリックします。

まもなくシャットダウンしますのリストから、アプリケーションが閉じる前に実行されるメソッドを選択します。例えば、ファイルを消去や削除するために利用できます。このイベントにローカルメソッドを追加するには、**ローカルメソッド作成**のボタン(+)をクリックします。参照するメソッドは、ブーリアン値を返します。閉じられた際に戻り値が false の場合、シャットダウンはキャンセルされます。

イベントノード(サブノード)

イベントノード(📌)では、ランタイムモデルの変更を検出するイベントを追加します。変更が発生すると、アクションが実行されます。データフィールドの変更、フィーチャの作成、フィーチャの削除を検出することができます。名前フィールドには、イベントオブジェクトの名前を入力します。

設定ウィンドウには、以下のセクションがあります。

データ変更イベントのソース


このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーがフィルタリング表示されたツリーがあります。ノードは何らかの種類のデータを表すか、または子ノードを持つかのいずれかです。イベントの場合、宣言の下の変数宣言はソースとして使用できます。さらに、**モデル**では、グローバルパラメータ、コンポーネント内の定義の下の変数、および明示的選択ノードをソースとして使用できます。明示的選択をソースとして使用すると、明示的な選択が変更されたときに実行されるメソッドを持つことができます（その変更は、あるメソッドのコードによって、または選択入力やフォームオブジェクトとのインタラクティブな操作によってトリガーされます）。例えば、ユーザーがグラフィックス内の境界をクリックしたときに、アプリケーションがメソッドを実行するようにすることができます。

データを表すノードを選択すると、ツリーの下にある**ソースとして使用**のツールバーボタンが有効になります。それをクリックするか、Enter キーを押すか、ダブルクリックするか、ノードを右クリックして**ソースとして使用**を選択して、選択ソースとして追加します。また、**データ変更イベントのソース**セクションのヘッダーにある**作成**ボタン(+)をクリックして、フォームオブジェクトの新規変数宣言を作成してソースとして使用することもできます。後者の場合、**変数を作成および使用**のダイアログボックスが開き、ソースのデータタイプ（該当する場合）とその名前を選択することができます。名前は、既存の変数宣言と競合することはできません。ツリーの下**ノード編集**のツールバーボタン(✎)をクリックすると、対応するノードに移動します。

ノードをソースとして選択すると、そのノードが選択されたソースとして**選択ソース**の下に表示されます。



複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースのデフォルト値は期待通りとは限りません。あるフォームオブジェクトのデフォルト値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。

通常、**パラメータ**ノードの下で使用可能なパラメータ、**変数**ノードの下の変数、およびアプリケーションビルダーのアプリケーションツリー部分の**宣言**ブランチ(**イベント**ブランチの下)で定義されたデータノードのみが表示されます。使用可能なデータノードのリストを拡張するには、**データ変更イベント**のソースセクションのヘッダーにある**モデルビルダーへスイッチしデータアクセスをアクティブ**のボタン()をクリックし、自動的にモデルビルダーに移動します。次に、アクセスしたいデータがある**モデルビルダー**ブランチのノードを選択します。そのボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることができる設定の横に**データソース**選択のチェックボックス(緑色の四角形)が表示されます。このデータをイベントのために使用可能なソースノードとして含めるには、そのチェックボックスをオンにします。

データソースに対する制限は、アプリケーションのユーザーインターフェースに渡されます。例えば、構造力学の**境界荷重**ノードの**荷重タイプ**リストでは、三つの値しか使用できません。このデータをソースとして使用するフォームオブジェクトは、それらの値のサブセットのみをサポートできます。

実行コマンド選択

このセクションでは、イベントに対して実行するコマンドを選択します。このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーがフィルタリング表示されたツリーがあります。ノードはコマンドをサポートしているか、または子ノードを持つかのいずれかです。一つまたは複数のコマンドをサポートするノードを選択すると、ツリーの下ツールバー内にある対応するコマンドツールバーのボタンが有効になります。また、ノードを右クリックして、そのノードの使用可能なコマンドのリストを取得することもできます。選択したノードのコマンドをクリックする(または Enter キーを押すか、ダブルクリックして、**実行**、**プロット**、**値設定**などのデフォルトコマンドでコマンドを追加する)ことによって、ツリーの下テーブルの最後の行にコマンドとノードが表示されます。このテーブルには実行される全てのノードが含まれており、このテーブルの下にあるツールバーを使用してコマンドを削除したり移動することができます。

モデルブランチでは、**パラメータ**ノードの下にあるパラメータのように、何らかの種類のデータ値を表すノードの全ては、**値設定**コマンドをサポートしています。テーブルに**値設定**コマンドが追加されると、その 3 列目の**引数**が有効になります。その列に、設定する値を入力します。配列を表すデータの場合には、中括弧とカンマを使って配列要素を入力します。例えば、{1,2,3}を入力して、値 1,2,3 の 3 要素配列を設定します。配列と行列の入力方法の詳細については、**配列 1D 文字列ノード**を参照してください。**補間**の関数ノードの下にある**ファイル名**ノードなど、ファイルのインポートを表すノードの場合は、**インポートファイル**のコマンドが使用できます。

ツリーには、**モデル**ブランチに加えて、アプリケーションツリーからのいくつかのブランチが含まれています。

- **フォーム**ブランチ: **フォーム**ノードは、フォームをアプリケーションのメインフォームとして設定する(つまり、アプリケーションウィンドウのコンテンツがこのフォームになる)**表示**のコマンドと、フォームを別のダイアログウィンドウとして開く**ダイアログ**として**表示**のコマンドをサポートしています。
- **GUI** コマンドブランチ: このブランチの下にあるコマンドは、三つのサブカテゴリにグループ化されています。
 - **ファイル**コマンド: これには、**アプリケーションを保存**(現在の名前でアプリケーションを保存する); **名前を付けてアプリケーションを保存**(ユーザーが適切な場所にアプリケーションを保存できるようにするファイルブラウザダイアログを開く); **サーバー上でアプリケーションを保存**; **サーバー上でアプリケーションを別名で保存**; **ファイルを開く**(引数列の有効な URI パスを使用して指定されたアプリケーションファイルリソースを開く); **名前を付けてファイルを保存**(同様に、ユーザーが**引数**列で指定した名前でファイルを保存できるようにする); および**アプリケーションを終了**(実行中のアプリケーションを閉じる)が含まれています。アプリケーションが COMSOL サーバー上で実行されている場合、**サーバー上でアプリケーションを保存**と**サーバー上でアプリケーションを別名で保存**のコマンドは、現在の状態を COMSOL サーバーアプリケーションライブラリに新しいアプリケーションとして保存します。

- **グラフィックスコマンド**: これには、画面にわたってズーム、現在のビューをリセット、シーンライト、透明度、印刷、全選択、および選択対象消去のコマンドがあります。全てのグラフィックコマンドで、コマンドを適用したいグラフィックスオブジェクトの名前を引数として追加します。
- **モデルコマンド**: これには、全解除と全メッシュ消去のコマンドがあります。

上記のいずれかのノードをダブルクリックするか右クリックして、実行コマンドを追加します。


- **宣言ブランチ**: このブランチには、アプリケーションビルダーウィンドウの宣言ブランチの下で追加されている変数宣言がタイプ別にグループ化されています。パラメータと同様に、値設定コマンドがサポートされています。
- **メソッドブランチ**: ここでは、メソッドノードが実行コマンドをサポートしています。
- **ライブラリブランチ**: ここでは、サウンドの下で、コマンドシーケンスで再生するサウンドファイルを選択できます。

ツリーの下にあるコマンドの一つをクリックすると、そのコマンドが下のリストの**コマンド**の列に表示されます。また、**アイコン**の列と**引数**の列があります。引数の列には、コマンドで使用される任意の引数を入力できます。そこには、コマンドが期待する引数のタイプを示すツールチップが表示されます。例えば、downloadtoclient コマンドの場合、引数は embedded:///myfile.txt のようなファイル名とそのパスの両方です。このような引数を定義するのに役立つ別のダイアログボックスがあります。詳細については、[ファイルスキームとファイル処理](#)を参照してください。

コマンドリストの下にある**ローカルメソッド作成**のボタン(+)をクリックすると、テーブル内のコマンドリスト全体を同等のコードを含む新しいローカルメソッドに変換し、そのローカルの onEvent メソッドのエディターウィンドウを開きます。ローカルメソッドを作成していないときに**新しいメソッドに変換**のツールバーボタン(📄)をクリックすると、テーブル内のコマンドのリスト全体を同等のコードを含む新しい**メソッドノード**に変換します。この操作をすると、コマンドのリストには作成された**メソッドノード**が1回だけ**実行**される操作しか含まれなくなります。コマンドの下でメソッドを選択するか、リストにメソッドが一つしかない場合は、**メソッドへ**のボタン(📄)をクリックすると、そのメソッドのエディターウィンドウに移動します。**引数編集**のボタン(🔍)の詳細については、[宣言とコマンドシーケンスでの初期値と引数の編集](#)を参照してください。**上へ移動**(↑)、**下へ移動**(↓)、**削除**(✖)のツールバーボタンを使用して、リストからコマンドを整理したり削除します(削除の場合はローカルメソッドも削除されます)。

使用可能なノードのリストを拡張するには、**実行コマンド選択**セクションのヘッダーにある**モデルビルダーへスイッチしデータアクセスをアクティブ**のボタン(🔗)をクリックし、自動的にモデルビルダーに移動します。次に、アクセスしたい**モデルビルダーブランチ**のノードを選択します。そのボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることができる設定の横に**データソース選択**のチェックボックス(緑色の四角形)が表示されます。このデータをイベントのために使用可能なソースノードとして含めるには、このチェックボックスをオンにします。


宣言ブランチ


宣言ブランチ()の下全てのノードは、ある種の宣言を指定します。通常は新しいデータフィールドで、さまざまなフォームオブジェクトにバインドすることができます。例えば、**文字列**ノードは、文字列値を格納する一つ以上のデータフィールドを宣言します。その意味ではグローバルパラメータと同じですが、グローバルパラメータの場合には文字列値は有効な式でなければならないのに対し、文字列データフィールドにはそのような制限はありません。次の宣言ノードの詳細については、以降の該当するセクションを参照してください。所望の宣言は、**宣言**ノードを右クリックして選択追加することができます。



- **スカラー**の下:
 - 文字列ノード
 - ブーリアンノード
 - 整数ノード
 - 倍精度ノード
- **配列 1D**の下:
 - 配列1D文字列ノード
 - 配列1Dブーリアンノード
 - 配列1D整数ノード
 - 配列1D倍精度ノード
- **配列 2D**の下:
 - 配列2D文字列ノード
 - 配列2Dブーリアンノード
 - 配列2D整数ノード
 - 配列2D倍精度ノード
- 選択リストノード
- ファイルノード
- 単位セットノード
- グラフィックスデータノード



スカラー、**配列 1D**、および**配列 2D**の下には、各データ型宣言のうちの一つだけを追加することができます。そのように宣言を追加した後には、それらの宣言はコンテキストメニューから消えます。

さらに、フォームオブジェクトまたはメニュー項目にショートカットを追加した場合は、**ショートカット**ノード()が表示されません。[ショートカットの追加](#)を参照してください。

変数のリストの下にある  をクリックすると開くウィンドウを使用して、スカラーノードと配列ノードの初期値を編集するためのツールについては、[宣言とコマンドシーケンスの初期値と引数の編集](#)を参照してください。

スカラー、**配列 1D**、および**配列 2D**のノード内の全ての変数リストに於いて、**ファイルに保存**のボタン()をクリックすると、**ファイルに保存**ダイアログボックスに**ファイル名**を拡張子.txtを含めて入力することができます。そこで**保存**をクリックすれば、そのテキストファイルを保存できます。その情報は、画面に表示されているのと同じ順序でスペースで区切られた列に格納される形式となっています。**ファイルからロード**のボタン()と**ファイルからロード**のダイアログボックスを使用して、例えばスプレッドシートプログラムなどで生成されたテキストファイルのデータを読み込むことができます。データはスペースまたはタブで区切られている必要があります(ライセンスに LiveLink™ for Excel® が含まれている場合は、Microsoft® Excel® ワークブックのスプレッドシートに記載する必要があります)。

配列シンタックス

配列のデフォルト値は、特別な構文を使用して入力して任意の長さの配列にすることができます。配列の定義は、中括弧（{ }）で始まり中括弧で終わりでなければならず、各要素はコンマで区切られる必要があります。文字列を示すには、シングルクォーテーション（'）で囲みます。配列要素の中に特殊な文字（例えばスペースやコンマを含む）が必要な場合は、要素をシングルクォーテーション（'）で囲む必要があります。文字列自体にアポストロフィ（'）が含まれている場合は、二つのシングルクォーテーション（"）を使用します。次の例を参照してください。

配列シンタックス	結果として生じる配列
{1, 2, 3}	要素が 1, 2, 3 の 3 要素の配列
{}	空の配列
{'one, two', 'three by four'}	特殊なシンタックスの要素を持つ 2 要素の配列
{{1, 2, 3}, {4, 5, 6}}	3 要素の配列を含んだ 2 要素の配列（2 行 3 列の行列）
{{1, 2},{'one, two', 'Poisson's ratio'}}	二つの 2 要素の配列を含んだ 2 要素の配列（2 行 2 列の行列）

非矩形配列は可能ですが、アプリケーションのコンテキストではあまり役に立ちません。

構文解析によって、中括弧のレベルが一つの場合には単一レベルの文字列配列が返され、中括弧の二つのネスト化したレベルがある場合には二つの配列（または文字列行列）が返されます。

文字列ノード

文字列ノード^(abc) は、フォームオブジェクトとメソッドからアクセス可能な一つ以上の名前付き文字列を宣言します。入力フィールド、コンボボックス、チェックボックスなど、多くのフォームオブジェクトで文字列をソースとして使用できます。その**設定**ウィンドウには、以下のセクションがあります。

変数リスト

このセクションには一つのテーブルがあり、そこでは行ごとに一つの格納する文字列を指定します。**名前**の列に名前を、**初期値**の列に文字列の初期値を指定します。また、**説明**の列にはオプションの文字列の説明を追加することもできます。



文字列の初期値を指定することはできませんが、その文字列を異なる値を指定するフォームオブジェクトのソースとして使用すると、その値が上書きされることがあります。

上へ移動 (↑)、**下へ移動** (↓)、**削除** (✖) のツールバーボタンを使用して、リストから選択肢を整理したり削除します。

コードアクセス

メソッドのコードでは、通常の Java String 変数のように、string1 といったような名前の文字列にアクセスします。

```
String s = string1 ;  
string1 = "newValue" ;
```

ブーリアンノード

ブーリアンノード^(d) は、フォームオブジェクトとメソッドからアクセス可能な一つ以上の名前付きスカラーブーリアン変数を宣言します。例えば、ブーリアン変数はチェックボックスのソースとして使用できます。その**設定**ウィンドウには、以下のセクションがあります。




変数リスト

このセクションには一つのテーブルがあり、そこでは行ごとに一つの格納するブーリアン変数を指定します。**名前**の列に名前を、**初期値**の列に文字列の初期値 (true または false; デフォルトは false) を指定します。また、**説明**の列にはオプションのブーリアン変数の説明を追加することもできます。



ブーリアン変数の初期値を指定することはできませんが、そのブーリアン変数を異なる値を指定するフォームオブジェクトのソースとして使用すると、その値が上書きされることがあります。

文字列として、on または off、true または false、yes または no をブーリアン値として使用できます (全て大文字と小文字を区別しません)。

上へ移動 ()、下へ移動 ()、削除 () のツールバーボタンを使用して、リストから選択肢を整理したり削除します。

コードアクセス

メソッドのコードでは、通常の Java boolean 変数のように、bool1 といったような名前のブーリアン変数にアクセスします。

```
boolean b = bool1 ;
bool1 = newValue ;
```

整数ノード




整数ノード (^{1.23}) は、フォームオブジェクトとメソッドからアクセス可能な一つ以上の名前付きスカラー整数を宣言します。整数でなければならない値に対して、入力のソースとして整数を使用することができます。その設定ウィンドウには、以下のセクションがあります。

変数リスト

このセクションには一つのテーブルがあり、そこでは行ごとに一つの格納する整数を指定します。**名前**の列に名前を、**初期値**の列に初期値 (デフォルトは 0) を指定します。また、**説明**の列にはオプションの整数の説明を追加することもできます。



整数の初期値を指定することはできませんが、その整数を異なる値を指定するフォームオブジェクトのソースとして使用すると、その値が上書きされることがあります。

上へ移動 ()、下へ移動 ()、削除 () のツールバーボタンを使用して、リストから選択肢を整理したり削除します。

コードアクセス

メソッドのコードでは、通常の Java int 変数のように、int1 といったような名前の整数にアクセスします。

```
int i = int1 ;
int1 = newValue ;
```

倍精度ノード

倍精度ノード (^{1.23}) は、フォームオブジェクトとメソッドからアクセス可能な一つ以上の名前付きスカラー倍精度浮動小数点値を宣言します。浮動小数点数 (いくつかのスカラー値) である入力では、そのソースとして倍精度を使用することができます。その設定ウィンドウには、以下のセクションがあります。

変数リスト

このセクションには一つのテーブルがあり、ここでは行ごとに一つの格納する倍精度を指定します。**名前**の列に名前を、**初期値**の列に初期値(デフォルトは 0.0)を指定します。また、**説明**の列にはオプションの倍精度の説明を追加することもできます。



倍精度の初期値を指定することはできませんが、その倍精度を異なる値を指定するフォームオブジェクトのソースとして使用すると、その値が上書きされることがあります。

上へ移動(↑)、**下へ移動**(↓)、**削除**(☒)のツールバーボタンを使用して、リストから選択肢を整理したり削除します。

コードアクセス

メソッドのコードでは、通常の Java double 変数のように、dbl1 といったような名前の倍精度にアクセスします。

```
double d = dbl1 ;
dbl1 = newValue ;
```

配列 1D 文字列ノード

配列 1D 文字列ノード(☒)は、フォームオブジェクトとメソッドからアクセス可能な一つ以上の名前付き文字列配列を宣言します。文字列配列の要素数には制限がありません。通常、文字列配列は、テーブルの列を格納する用途で利用されます。**設定**ウィンドウには、次のセクションがあります。

変数リスト

このセクションには一つのテーブルがあり、ここでは行ごとに一つの格納する文字列配列を指定します。**名前**の列に名前を、**初期値**の列に配列の初期値を、**新規要素値**の列に各要素の新しい要素値を指定します。新規要素値には、文字列配列の新しい要素が特定の追加操作(例えばテーブルなどで)で取得される値を指定します。また、**説明**の列にはオプションの文字列配列の説明を追加することもできます。



文字列配列の初期値を指定することはできませんが、その配列を異なる値を指定するフォームオブジェクトのソースとして使用すると、それらの値が上書きされることがあります。

上へ移動(↑)、**下へ移動**(↓)、**削除**(☒)のツールバーボタンを使用して、リストから選択肢を整理したり削除します。

コードアクセス

メソッドのコードでは、通常の Java String[] 変数のように、array1 といったような名前の文字列配列にアクセスします。

```
String[ ] sa = array1 ;
array1 = new String[ ] { "element1", "element2" } ;
```

配列 1D ブーリアンノード

配列 1D ブーリアンノード(☒)は、フォームオブジェクトとメソッドからアクセス可能な一つ以上の名前付きブーリアン配列を宣言します。ブーリアン配列の要素数には制限がありません。通常、ブーリアン配列は、ブーリアン値のリストを指定する用途で利用されます。**設定**ウィンドウには、次のセクションがあります。

変数リスト

このセクションには一つのテーブルがあり、ここでは行ごとに一つの格納するブーリアン配列を指定します。**名前**の列に名前を、**初期値**の列に初期値を、**新規要素値**の列に各要素の新しい要素値(デフォルト値は false で、最初の値を表します)を

指定します。新規要素値には、ブーリアン配列の新しい要素が特定の追加操作（例えばテーブルなどで）で取得される値を指定します。また、**説明**の列にはオプションのブーリアン配列の説明を追加することもできます。



ブーリアン配列の初期値を指定することはできますが、その配列を異なる値を指定するフォームオブジェクトのソースとして使用すると、それらの値が上書きされることがあります。

ブーリアン配列が文字列として宣言されている場合は、on または off、true または false、yes または no をブーリアン値として使用できます（全て大文字と小文字を区別しません）。


上へ移動(↑)、下へ移動(↓)、削除(✖)のツールバーボタンを使用して、リストから選択肢を整理したり削除します。

コードアクセス

メソッドのコードでは、通常の Java boolean [] 変数のように、booleanArray1 といったような名前のブーリアン配列にアクセスします。

```
boolean[] ba = booleanArray1 ;
booleanArray1 = new boolean[] { value1, value2 } ;
```

配列 1D 整数ノード

配列 1D 整数ノード()は、フォームオブジェクトとメソッドからアクセス可能な一つ以上の名前付き整数配列を宣言します。整数配列の要素数には制限がありません。通常、整数配列は、整数だけの値しかない配列を指定する用途で利用されます。**設定**ウィンドウには、次のセクションがあります。

変数リスト

このセクションには一つのテーブルがあり、そこでは行ごとに一つの格納する整数配列を指定します。**名前**の列に名前を、**初期値**の列に初期値を、**新規要素値**の列に各要素の新しい要素値（デフォルト値は 0 で、最初の値を表します）を指定します。新規要素値には、整数配列の新しい要素が特定の追加操作（例えばテーブルなどで）で取得される値を指定します。また、**説明**の列にはオプションの整数配列の説明を追加することもできます。



整数配列の初期値を指定することはできますが、その配列を異なる値を指定するフォームオブジェクトのソースとして使用すると、それらの値が上書きされることがあります。


上へ移動(↑)、下へ移動(↓)、削除(✖)のツールバーボタンを使用して、リストから選択肢を整理したり削除します。

コードアクセス

メソッドのコードでは、通常の Java int [] 変数のように、intArray1 といったような名前の整数配列にアクセスします。

```
int[] ia = intArray1 ;
intArray1 = new int[] { value1, value2 } ;
```

配列 1D 倍精度ノード




配列 1D 倍精度ノード()は、フォームオブジェクトとメソッドからアクセス可能な一つ以上の名前付き倍精度浮動小数点配列を宣言します。倍精度配列の要素数には制限がありません。通常、倍精度配列は、浮動小数点値を持つ配列入力を指定する用途で利用されます。整数だけの値しかない配列を指定します。**設定**ウィンドウには、次のセクションがあります。

変数リスト

このセクションには一つのテーブルがあり、ここでは行ごとに一つの格納する倍精度配列を指定します。**名前**の列に名前を、**初期値**の列に初期値を、**新規要素値**の列に各要素の新しい要素値(デフォルト値は 0.0 で、最初の値を表します)を指定します。新規要素値には、倍精度浮動小数点配列の新しい要素が特定の追加操作(例えばテーブルなどで)で取得される値を指定します。また、**説明**の列にはオプションの倍精度配列の説明を追加することもできます。



倍精度配列の初期値を指定することはできますが、その配列を異なる値を指定するフォームオブジェクトのソースとして使用すると、それらの値が上書きされることがあります。


上へ移動()、**下へ移動**()、**削除**()のツールバーボタンを使用して、リストから選択肢を整理したり削除します。

コードアクセス

メソッドのコードでは、通常の Java double [] 変数のように、dblarray1 といったような名前の倍精度配列にアクセスします。

```
double[] da = dblarray1 ;
dblarray1 = new double[] { value1, value2 } ;
```

配列 2D 文字列ノード

配列 2D 倍精度ノード()は、フォームオブジェクトとメソッドからアクセス可能な一つ以上の名前付き 2D 文字列配列(行列)を宣言します。2D 文字列配列の要素数には制限がありませんが、列数を指定できます。**設定**ウィンドウには、次のセクションがあります。




変数リスト

このセクションには一つのテーブルがあり、ここでは行ごとに一つの格納する 2D 文字列配列を指定します。

名前の列に名前を、**列数**の列のリストから 2D 文字列配列の列数を(配列に含まれる列数がわからない場合には**未定義**を使用)、**初期値**の列に初期値を、**新規要素値**の列に新しい要素値を指定します。新規要素値には、2D 文字列配列の新しい要素が特定の追加操作で取得される値を指定します。また、**説明**の列にはオプションの 2D 文字列配列の説明を追加することもできます。



2D 文字列配列の初期値を指定することはできますが、その配列を異なる値を指定するフォームオブジェクトのソースとして使用すると、それらの値が上書きされることがあります。


上へ移動()、**下へ移動**()、**削除**()のツールバーボタンを使用して、リストから選択肢を整理したり削除します。

コードアクセス

メソッドのコードでは、通常の Java String [][] 変数のように、matrix1 といったような名前の 2D 文字列配列にアクセスします。

```
String[][] sm = matrix1 ;
matrix1 = new String[][] { { "element11", "element12" }, { "element21", "element22" } } ;
```

配列 2D ブーリアンノード

配列 2D ブーリアンノード()は、フォームオブジェクトとメソッドからアクセス可能な一つ以上の名前付き 2D ブーリアン配列(行列)を宣言します。2D ブーリアン配列の要素数には制限がありませんが、列数を指定できます。**設定**ウィンドウには、次のセクションがあります。

変数リスト




このセクションには一つのテーブルがあり、そこでは行ごとに一つの格納する 2D ブーリアン値配列を指定します。

名前の列に名前を、列数の列のリストから 2D ブーリアン配列の列数を (配列に含まれる列数がわからない場合には**未定義**を使用)、初期値の列に初期値を、新規要素値の列に新しい要素値を指定します。新規要素値には、2D ブーリアン配列の新しい要素が特定の追加操作で取得される値を指定します。また、説明の列にはオプションの 2D ブーリアン配列の説明を追加することもできます。



2D ブーリアン配列の初期値を指定することはできませんが、その配列を異なる値を指定するフォームオブジェクトのソースとして使用すると、それらの値が上書きされることがあります。

文字列として、on または off、true または false、yes または no をブーリアン値として使用できます (全て大文字と小文字を区別しません)。


上へ移動 ()、下へ移動 ()、削除 () のツールバーボタンを使用して、リストから選択肢を整理したり削除します。

コードアクセス

メソッドのコードでは、通常の Java boolean [][] 変数のように、boolmatrix1 といったような名前の 2D ブーリアン配列にアクセスします。

```
boolean[ ][ ] bm = boolmatrix1 ;  
boolmatrix1 = new boolean[ ][ ] { { value11, value12 }, { value21, value22 } } ;
```

配列 2D 整数ノード

配列 2D 整数ノード () は、フォームオブジェクトとメソッドからアクセス可能な一つ以上の名前付き 2D 整数配列 (行列) を宣言します。2D 整数配列の要素数には制限がありませんが、列数を指定できます。設定ウィンドウには、次のセクションがあります。




変数リスト

このセクションには一つのテーブルがあり、そこでは行ごとに一つの格納する 2D 整数配列を指定します。

名前の列に名前を、列数の列のリストから 2D 整数配列の列数を (配列に含まれる列数がわからない場合には**未定義**を使用)、初期値の列に初期値を、新規要素値の列に新しい要素値を指定します。新規要素値には、2D 整数配列の新しい要素が特定の追加操作で取得される値を指定します。また、説明の列にはオプションの 2D 整数配列の説明を追加することもできます。



2D 整数配列の初期値を指定することはできませんが、その配列を異なる値を指定するフォームオブジェクトのソースとして使用すると、それらの値が上書きされることがあります。


上へ移動 ()、下へ移動 ()、削除 () のツールバーボタンを使用して、リストから選択肢を整理したり削除します。

コードアクセス

メソッドのコードでは、通常の Java int [][] 変数のように、intmatrix1 といったような名前の 2D 整数配列にアクセスします。

```
int[ ][ ] im = intmatrix1 ;  
intmatrix1 = new int[ ][ ] { { value11, value12 }, { value21, value22 } } ;
```

配列2D倍精度ノード

配列 2D 倍精度ノード () は、フォームオブジェクトとメソッドからアクセス可能な一つ以上の名前付き 2D 倍精度浮動小数点値配列 (行列) を宣言します。2D 倍精度配列の要素数には制限がありませんが、列数を指定できます。設定ウィンドウには、次のセクションがあります。




変数リスト

このセクションには一つのテーブルがあり、そこでは行ごとに一つの格納する 2D 倍精度配列を指定します。

名前の列に名前を、列数の列のリストから 2D 倍精度配列の列数を (行列に含まれる列数がわからない場合には **未定義** を使用)、初期値の列に初期値を、新規要素値の列に新しい要素値 (デフォルト値は 0.0) を指定します。新規要素値には、2D 倍精度配列の新しい要素が特定の追加操作で取得される値を指定します。また、説明の列にはオプションの 2D 倍精度配列の説明を追加することもできます。



2D 倍精度配列の初期値を指定することはできますが、その配列を異なる値を指定するフォームオブジェクトのソースとして使用すると、それらの値が上書きされることがあります。


上へ移動 ()、**下へ移動** ()、**削除** () のツールバーボタンを使用して、リストから選択肢を整理したり削除します。

コードアクセス







メソッドのコードでは、通常の Java double [][] 変数のように、dblmatrix1 といったような名前の 2D 倍精度配列にアクセスします。

```
double[ ][ ] dm = dblmatrix1 ;
dblmatrix1 = new double[ ][ ] { { value11, value12 }, { value21, value22 } } ;
```

選択リストノード

選択リストノード () には、コンボボックス、リスト、およびラジオボタンの選択肢のリストが含まれます。このノードをこれらの許容値の一部として参照することができます。表示するラベルを指定できる **ラベルフィールド** と、この選択リストオブジェクトの名前を指定できる **名前フィールド** があります。設定ウィンドウには、次のセクションがあります。

コンテンツをリスト

このセクションには、**値**の列と**表示名**の列を持つテーブルがあります。最初の列にはユーザーが選択するプロパティ値を入力し、2番目の列にはコンボボックスリストなどに表示するための対応するテキストを入力します。**上へ移動** ()、**下へ移動** ()、**削除** () のツールバーボタンを使用して、リストから選択肢を整理したり削除します。全てのテーブルの内容を削除するには、**テーブル消去**のボタン () をクリックします。**ファイルに保存**のボタン () をクリックすると、**ファイルに保存**ダイアログボックスに**ファイル名**を入力することができます。そこで**保存**をクリックすれば、ファイルを保存できます。**ファイルからロード**のボタン () と**ファイルからロード**のダイアログボックスを使用して、例えばスプレッドシートプログラムなどで生成されたテキストファイルのデータを読み込むことができます。




選択リストのデータソースがグローバルパラメータまたは変数である場合は、値として単純な数値を使用することをお勧めします。それ以外の場合は、その値が有効な式として評価できることを確認する必要があります。グローバルパラメータに対して無効な式は、エラー発生の原因となります。

内部的には、選択リストは N 行 2 列の倍精度文字列配列です。ここで、N はテーブルの行数です。アプリケーションを実行しているときに、**名前**フィールドで指定した名前 で新しい倍精度文字列配列をプロパティに設定して、このリストの内容を変更することができます。



選択リストノードを右クリックして、アクティベーション条件のサブノードを追加することができます。

アクティベーション条件

アクティベーション条件ノード()は、親ノードがアクティブか非アクティブかを決定する true または false 評価する条件です。例えば、親が選択リストノードである場合、その選択リストのアクティベーション条件が false のときには、それを使用しているコンボボックスからそのリストが除外されます。ラベルフィールドには表示するラベルを、名前フィールドにはアクティベーション条件オブジェクトの名前を指定します。

アクティベーション条件の設定ウィンドウには、以下のセクションがあります。



ソース

このセクションには、アプリケーションビルダーウィンドウとモデルビルダーウィンドウのツリーがフィルタリング表示されたツリーがあります。ノードは何らかの種類のデータを表すか、または子ノードを持つかのいずれかです。アクティベーション条件の場合、宣言の下の変数宣言はソースとして使用できます。データを表すノードを選択すると、ツリーの下にあるソースとして使用するツールバーボタンが有効になります。ノードをクリックするか、Enter キーを押すか、ダブルクリックするか、ノードを右クリックしてソースとして使用を選択して、選択ソースとして追加します。また、ソースセクションのヘッダーにある作成ボタン()をクリックして、フォームオブジェクトの新規変数宣言を作成してソースとして使用することもできます。後者の場合、変数を作成および使用のダイアログボックスが開き、ソースのデータタイプ(該当する場合)とその名前を選択することができます。名前は、既存の変数宣言と競合することはできません。ツリーの下ノード編集のツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムはモデルビルダーに切り替えます。


ソースとしてノードを選択すると、選択ソースラベルに選択されたソースとしてそのノードが表示されます。




複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースのデフォルト値は期待通りとは限りません。あるフォームオブジェクトのデフォルト値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。

通常、パラメータノードの下の使用可能なパラメータ、変数ノードの下の変数、および宣言ブランチの下に定義されているデータノードのみが表示されます。使用可能なデータノードのリストを拡張するには、ソースセクションのヘッダーにあるモデルビルダーヘイスイッチデータアクセスをアクティベートのボタン()をクリックし、自動的にモデルビルダーに移動します。次に、アクセスしたいデータがあるモデルビルダーブランチのノードを選択します。そのボタンをアクティブにすると、選択したノードの設定ウィンドウには、含めることができる設定の横に緑色のチェックボックスが表示されます(例えば、固体力学モデルの境界荷重ノード)。そのノードからのデータは、ツリーのソースの下アイコン()を持つノードとして表示されます。データソースに対する制限は、アプリケーションのユーザーインターフェースに渡されます。例えば、構造力学の境界荷重ノードの荷重タイプリストでは、三つの値(単位面積あたりの力、総力、圧力)しか使用できません。このデータをソースとして使用するフォームオブジェクトは、それらの値のサブセットのみをサポートできます。

条件

条件のテーブルには、データフィールドの値と照合する値を入力します。一致する値を指定すると、通常は一致条件が true になりますが、入力値で条件を反転のチェックボックスを選択すると条件を反転させることができます。削除()をクリックすると、アクティベート値がテーブルから削除されます。

ファイルノード

ファイルノード()は、アプリケーションを開発するときにファイルを参照することを可能にします。実行時にアプリケーションにインポートしたり、クライアントからアップロードしたりできる外部ファイルを宣言できます。ファイル宣言は、実行時にファイルをアプリケーションにアップロードします。例えば、ファイル宣言の名前が file1 の場合(その名前は、設定ウィンドウの

名前フィールドに入力します)、ファイルスキーム `embedded:/// file1` を使用してそれをアクセスすることができます。ファイルスキームの詳細については、[ファイルスキームとファイル処理](#)を参照してください。ユーザーがファイルをブラウズして選択できるようにするには、[ファイルインポートオブジェクト](#)を使用し、ファイルノードをソースとして選択します。ラベルフィールドに表示するラベルを入力することもできます。


ファイル宣言の**設定**ウィンドウには、次のセクションがあります。

ファイル位置

ターゲットディレクトリのリストから、ファイルのタイプと用途に応じて、次のいずれかのディレクトリを選択します。

- ユーザーを選択すると、アプリケーションのユーザーのディレクトリに格納されます。
- 一時的(デフォルト)を選択すると、アプリケーションがアクティブな間だけテンポラリファイルに格納されます。




単位セットノード

単位セットノード()では、適用可能な物理量に関する単位のリストを持つ単位のセットを定義する方法が提供されます。単位セットを定義することで、アプリケーションのユーザは、入力のために使用する単位を選択することができます。入力フィールドとスライダーフォームオブジェクトでは、単位セットの特定の単位リストを使用して、フォームオブジェクトの表示単位を指定することができます。通常、コンボボックスオブジェクトを使用する場合は、(ソースにすることによって)使用する単位セットを指定します。ラジオボタン、リストボックス、およびコンボボックスのフォームオブジェクトで、ソースとして単位セットを使用することができます。

設定ウィンドウには、次のセクションがあります。

単位グループ





このテーブルでは、一つの単位セット内で管理する複数の物理量として、または複数の単位セットノードを使用する場合の単一の物理量として、アプリケーションで使用する単位を各物理量ごとに一つずつ追加します。値の列には単位に使用する値を入力し、表示名の列には単位の名前を入力します。この単位の名前は、例えばコンボボックスオブジェクトで使用されるような場合に、そこに表示されるテキストのことです。値と表示名は、同じにすることができます(例えば、SI または inch)。このセクションで追加する各単位の行に応じて、単位リストのテーブルにはそれに対応する列が追加されます。

上へ移動()、下へ移動()、削除()のツールバーボタンを使用して、リストから選択肢を整理したり削除します。


初期値のリストから、アプリケーションを起動するときに使用される単位セットを選択します。

単位リスト

このセクションにはテーブルがあり、行にはフィジカルプロパティを追加し、列には定義されている単位グループの全てに対応する単位が追加されます。名前の列には名前を指定し(例えば、質量または長さ)、それ以降に続く列では対応する単位を指定します: 例えば、SI 単位の SI 列では kg と m、Imperial 単位では Imperial 列の lb と ft です。あるいは、例えば、ある長さに別の単位セットノードを使用する場合は、その列を cm、m、および in(それぞれセンチメートル、メートル、インチ)のようにします。

追加ボタン()をクリックして、別のフィジカルプロパティを新しい行としてリストに追加することができます。上へ移動()、下へ移動()、削除()のツールバーボタンを使用して、リストから行を整理したり削除します。

グラフィックスデータノード

グラフィックスデータノード()には、グラフィックスウィンドウでのデータピッキングに使用される全てのプロパティのオプションが含まれています。

設定ウィンドウには、以下のセクションがあります。

初期値

データの初期座標を、**座標**フィールドにカンマで区切られた値として入力します。

プロットの結果評価の初期値を、**結果評価**フィールドに入力します。

3D ジオメトリソースの初期値

このセクションでは、3D プローブ用の 3D ジオメトリソースを定義します。




ジオメトリックエンティティレベルのリストから、**ドメイン**(デフォルト)または**境界**を選択します。

ドメイン設定の下にある**ラインエントリ法**では、**ポイントおよび面法線**、**ポイントおよび方向**、**2 点**、または**なし**を選択します。**ラインに沿った深さ**では、0 と 1 の間の値を入力します。**修正されたポイント**では、**第 1 ポイント**または**第 2 ポイント**を選択します。

結果評価、ジオメトリックエンティティレベル、ラインに沿った深さ、および修正されたポイントについてここで定義された値は、例えば、入力フィールドオブジェクトのソースとして使用することができます。また、例えば、ボタンの**実行コマンド選択**セクションの**値設定**コマンドでも使用することができます。

ショートカットの追加

モデルビルダーでは、フォームオブジェクト、ツールバー、メニュー項目、リボン、およびほとんどのモデルノードにショートカットを作成することができます。ショートカットは、他の宣言と同様に Java の変数としてアプリケーションメソッドコードで使用できますが、ショートカットの変数は読み取り専用の変数です。





このようなオブジェクトとモデルノードの**設定**ウィンドウには、**名前**フィールド(**モデルビルダー**では**ラベル**フィールド)の右側に**ショートカット作成**のボタン()があります。そのボタンをクリックすると**ショートカット作成**ウィンドウが開き、そこで**ショートカットの名前**を編集することができます(デフォルトでは、ショートカットを作成するオブジェクトの名前と同じです)。また、Ctrl + K を使用してショートカットを作成することもできます。作成されたショートカットは、**宣言**の下に**ショートカットノード**()として表示されます。各オブジェクトに対して一つのショートカットしか作成できません。既に存在するとき別のショートカットを作成しようとする場合には、**ショートカットをリネーム**()ボタンをクリックします。すると、**ショートカットをリネーム**ウィンドウが開き、ショートカットの名前を編集することができます。

ショートカットは、オブジェクトの名前を変更、移動、コピー、または複製すると、自動的に更新されます。


ショートカットノードの**設定**ウィンドウには、以下のセクションがあります。

ショートカットリスト

名前の列では、必要に応じて、ショートカットの名前を編集することができます。**ターゲット**の列には、ターゲットオブジェクトのパスを含むフルネームが表示されます。**説明**の列では、ショートカットの説明を編集することができます。

上へ移動()、**下へ移動**()、**削除**()のツールバーボタンを使用して、リストからショートカットを整理したり削除します。選択したショートカットが参照するオブジェクトに移動するには、**ソース**へのボタン()をクリックします。

宣言とコマンドシーケンスでの初期値と引数の編集

宣言とコマンドシーケンスのテーブルの下で、**初期値編集**、または**引数編集**ボタン()を使用できます。**初期値編集**ボタンは、スカラー、配列 1D、および配列 2D の宣言ノードのテーブルの下にあります。同じボタンが、コマンドシーケンスをサポートするノード(**ボタン**、**イベント**、および**項目のノード**)の**設定**ウィンドウにあります。これは**引数編集**として使用されます。引数を使用するコマンドに対して有効です。メソッドの入力を定義する場合にも、**引数編集**を使用できます。

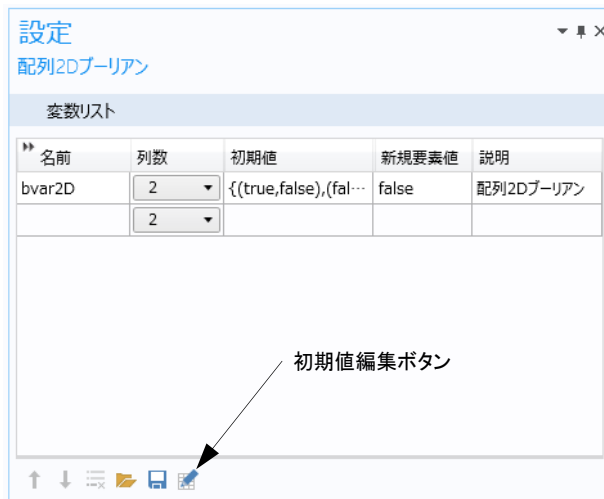


Figure 2-2: 配列 2D ブーリアンノードの設定ウィンドウの初期値編集ボタン

初期値編集ボタンをクリックすると、開いているウィンドウの初期値の列で選択した行を編集できます。例えば、次の図は、ブーリアン 2D 配列の行と列を編集する方法を示しています。

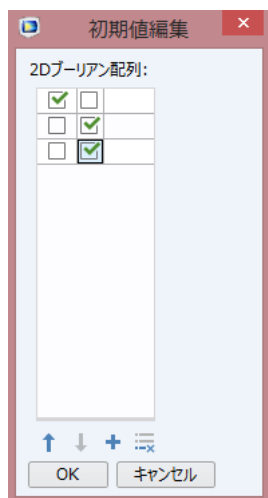


Figure 2-3: 配列 2D ブーリアンノードの初期値編集ウィンドウ

完了したら OK をクリックすると、初期値が適切な配列シンタックスに従って初期値の列に貼り付けられます。例えば、上図の編集では次の配列が生成されます。

```
{ { true, false }, { false, true }, { false, false } }
```

また、ロード(📁)と保存(💾)のボタンを使用して、テキストファイルから初期値をロードしたり、テキストファイルに保存したりすることもできます。

コマンドシーケンスでのツールバーボタンも同様に動作します。違いは、そのテーブルでは引数の列に入力するデータタイプを多くサポートしているという点です。データソースの値設定コマンドの編集では、宣言の下にあるノードの編集の場合と同じウィンドウが表示されます。プロットグループのプロットや結果テーブルへのデータの評価などの他のコマンドは、それら独自の引数編集ウィンドウが使用されています。

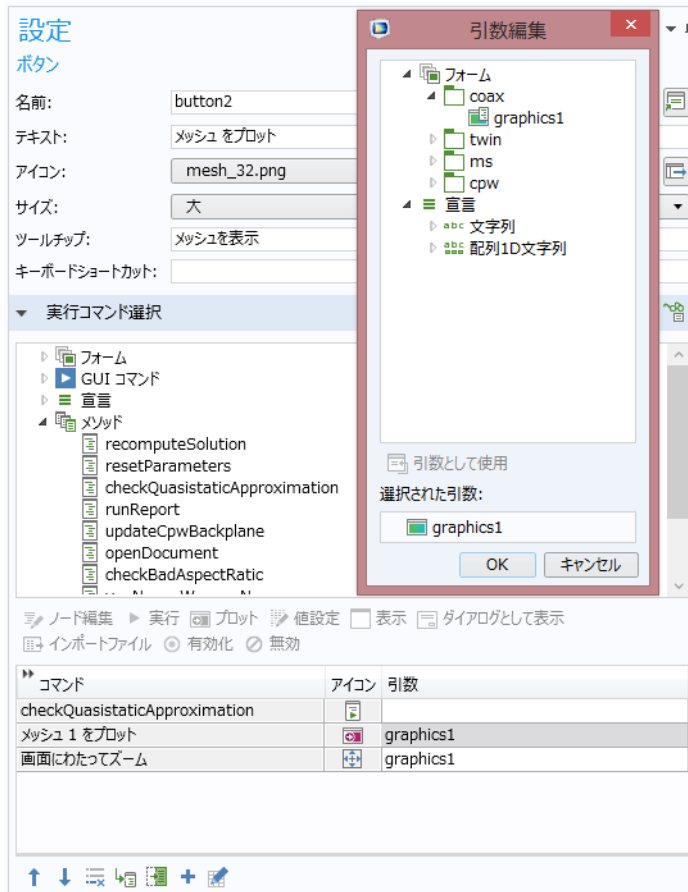




Figure 2-4: プロットコマンドとその引数編集ウィンドウを持つコマンドシーケンス

いくつかのコマンドシーケンスのコマンドでは、引数に宣言を使用することができます。このような宣言は、グラフィックスのプロット、組み込みコマンドのビュー、数値フィーチャの実行、宣言の設定、メソッドの実行などのコマンドに使用することができます。グラフィックをプロットするコマンド、組み込みコマンドをビュー表示するコマンド、および数値フィーチャを実行するコマンドについては、この機能を引数編集のダイアログボックスで使用することができます。引数をマニュアル入力することもできます。スカラー、配列としての配列、または行列としての行列に(個々の配列または行列要素にアクセスするのではなく)アクセスするには、宣言自体の名前を使用します(例えば output)。配列をスカラーとして、または行列を配列として使うには、インデックスを追加します(例えば output(1))。行列要素は、二つのインデックスを足して一つのスカラーとして取り出すことができます(例えば output(1,3))。インデックスを指定するときには、他の変数宣言を使用することができます(例えば output(n)、ここで n はスカラー整数宣言)。



プロット(Plot)のグラフィックスコマンドとビュー(View)の組み込みコマンドでは、グラフィックスオブジェクトへのパスを含んだ、文字列タイプの宣言(必要に応じて適切なインデックス付き)のみを使用することができます。それ以外の場合は、目的とする宣言(またはメソッド引数)の次元に対してソースとして使用する宣言の次元が合理的と解釈される限り、インデックスの有無にかかわらず、全てのタイプの宣言を使用することができます。例えば、プロットコマンドで graphics1 という名前のグラフィックオブジェクトと graphics1 という名前の文字列宣言がある場合、'graphics1'のようにシングルクォーテーションが使用されていない限り、グラフィックスオブジェクトが使用されるときにその宣言の内容が使用されます。数値フィーチャを実行するコマンドに引数として宣言(整数、倍精度、または文字列)を追加することができます。その場合、数値フィーチャを評価することによって生成される初期パラメータの列を取り除いたテーブルデータは、コマンドの実行時にその宣言に入れられます。このコマンドは、次元が一致していることを確認します。コマンドシーケンスをメソッドに変換するとき、それらのチェックはコードに生成されます。





デフォルト値を編集するために使用されるウィンドウ、または列の数が定義されていない 2D 配列の値を設定するために使用されるウィンドウ(**引数編集ウィンドウ** など)では、必要に応じて **行を追加** () および **列追加** () ボタンをクリックして値を段階的に入力し、いくつかの行と列の組み合わせからなる 2D 配列を作成することができます。

メソッドブランチ




メソッドブランチ

メソッドブランチ()には、アプリケーションに追加したユーザ定義のメソッドへの参照が含まれています。**メソッドノード**を右クリックし、**新規メソッド**を選択して新しいメソッドを追加します。コマンドにตอบสนองして実行されるメソッドをサポートするフォームオブジェクトから、開く**新規メソッド**ウィンドウで定義されている**ローカルメソッド**を追加し、**名前**フィールドにメソッド名を入力してから **OK** をクリックします。その名前の新しい**メソッドノード**()が追加され、メソッドのエディターウィンドウが開き、新しいメソッドのコードを定義できます ([メソッドの操作](#)を参照)。

モデルメソッドブランチ





モデルメソッドブランチ()には、**モデルビルダー**ウィンドウで定義されたモデルに関連してコードを実行するために追加した、ユーザ定義のモデルメソッドへの参照が含まれています。**メソッドノード**を右クリックし、**新しいモデルメソッド**を選択して新しいモデルメソッドを追加します。また、モデルビルダーの**デベロッパーツールバー**から、**メソッドツールバー**の**新規メソッド**をクリックし、**メソッドタイプ**リストから**モデルメソッド**を選択して、モデルメソッドを追加することもできます。その名前の新しい**モデルメソッドノード**()が追加され、モデルメソッドのエディターウィンドウが開きます。ここで、新しいメソッドのコードを定義できます ([メソッドの操作](#)を参照)。

メソッドノード

メソッドノード()はメソッドを表します。— **イベントノード**、**項目ノード**、**ボタンノード**、その他の**メソッドノード**など、他のノードから呼び出されたときに実行される小さなプログラムです。**メソッドノード**の名前はメソッドの名前で、**名前**フィールドに入力できます。右クリックして**編集**()を選択すると、モデルメソッドを編集できるエディターウィンドウが開きます。ローカルメソッドの場合は、**ソースへのボタン**()をクリックしてローカルメソッドが作成されたフォームオブジェクトに移動します。メソッドの作成については、[メソッドの作成](#)、および[メソッドノードとメソッドエディターウィンドウ](#)を参照してください。


メソッドノードの**設定**ウィンドウには、次のセクションがあります。


入力と出力

入力のテーブルの中で、メソッドに一つ以上の入力を追加できます。**追加ボタン**()をクリックして、引数をリストに追加します。**名前**列では、引数の名前を編集できます (デフォルト: arg1 など)。**タイプ**列で、データタイプ (文字列、ブーリアン、整数、倍精度、または 1D や 2D 配列タイプ) を選択します。**デフォルト**列にデフォルト値を割り当て、**説明**列に入力の説明を入力できます。この説明は、メソッドパラメーターに注釈を付けるために使用できます。リストからコマンドを整理して削除するには、**上へ移動**()、**下へ移動**()、および**削除**()のツールバーボタンを使用します。

出力リストから、出力データタイプを選択します。**なし** (デフォルトは出力なし)；**文字列**；**ブーリアン**；**整数**；**倍精度**；または 1D または 2D 配列のタイプです。全ての出力データタイプに対して、メソッドの出力の名前を**名前**フィールドに入力できます (デフォルト: out)。出力変数の値はメソッドが返す値です。割り当てられていない場合、このメソッドはそのタイプのデフォルト値 (0、null、または false) を返します。

モデルメソッドノード

モデルメソッドノード()はメソッドを表します。— モデルビルダーを使用して開発されたモデルの一部としてユーザが呼び出したときに実行される小さなプログラムです。**モデルメソッドノード**の名前はメソッドの名前で、**名前**フィールドに入力でき


ます。右クリックして**編集**()を選択すると、モデルメソッドを編集できるエディターウィンドウが開きます。[メソッドの作成](#)、および[メソッドノードとメソッドエディターウィンドウ](#)を参照してください。また、モデルメソッドのサンプルについては、*Introduction to Application Builder* も参照してください。

モデルビルダーで表示のチェックボックスがデフォルトで選択されています。モデルビルダーでモデルメソッドを**メソッド呼出し追加メニュー**と**モデルメソッドを実行メニュー**(またはメソッド呼び出しを追加するためのコンテキストメニュー)に表示させたくない場合(例えば、それが単独で実行すべきでない補足的なモデルメソッドであるような場合)には、このチェックボックスをオフにします。




モデルメソッドノードの設定ウィンドウには、次のセクションがあります。

入力と出力

他のモデルメソッドから呼び出されるモデルメソッドは、入力引数と出力引数を持つことができます。**デベロッパーリボンのメソッド呼出しを実行**のボタンを使用して、モデルビルダーの全てのモデルメソッドをモデルメソッドの呼び出しとして使用できます。但し、モデルビルダーから直接実行するモデルメソッドでは、入力引数または出力引数を持つことはできません。従って、入力または出力のないモデルメソッドは、**デベロッパーリボンのモデルメソッドを実行**にだけリスト表示される唯一のモデルメソッドです。


入力の表の中で、メソッドに一つ以上の入力を追加できます。**追加**ボタン()をクリックして、引数をリストに追加します。

名前列では、引数の名前を編集できます(デフォルト: arg1 など)。**タイプ**列で、データタイプ(**文字列**、**ブーリアン**、**整数**、**倍精度**、または 1D や 2D 配列タイプ)を選択します。**デフォルト**列では、必要に応じて入力のデフォルト値を指定できます。**説明**列には、入力の説明(オプション)を追加できます。説明を追加すると、モデルビルダーの対応するメソッド呼出しノードの**設定**ウィンドウに入力用に表示されます。この説明は、メソッドのエディターでコード補完とツールチップヘルプに表示されるときの入力引数にも表示されます。

リストからコマンドを整理して削除するには、**上へ移動**()、**下へ移動**()、および**削除**()のツールバーボタンを使用します。


出力リストから、出力データタイプを選択します。**なし**(デフォルトは出力なし); **文字列**; **ブーリアン**; **整数**; **倍精度**; または 1D または 2D 配列のタイプです。全ての出力データタイプに対して、メソッドの出力の名前を**名前**フィールドに入力できます(デフォルト: out)。出力変数の値はモデルメソッドが返す値です。割り当てられていない場合、このメソッドはそのタイプのデフォルト値(0、null、または false)を返します。

ライブラリブランチ


ライブラリブランチ()には、アプリケーションに追加した画像、サウンド、およびファイルのライブラリが含まれています。いくつかのライブラリには、あらかじめ定義されたいくつかの例が用意されています。ライブラリノードを右クリックして、次のライブラリコンポーネントのいずれかを選択することもできます。





- ユーティリティクラス。 [ユーティリティクラスノード](#)参照。
- 外部 Java ライブラリ。 [外部 Java ライブラリノード](#)参照。
- 例えば C コードに基づく外部ネイティブコードライブラリ。 [外部 C ライブラリノード](#)参照。

画像

画像ノード()には、アプリケーションで使用できる画像のライブラリ(例えば、.bmp、.gif、または.jpg ファイルなど)が含まれています。設定ウィンドウには、次のセクションがあります。

画像のリスト

このリストには、アプリケーションで使用可能な全ての画像が**名前**列に表示されます。説明列にオプションの説明を追加できます。いくつかのサンプル画像が常に含まれています。画像ファイルを追加するには、**ファイルをライブラリに追加**  ボタンをクリックします。ファイルを選択してリストに追加できるファイルブラウザが開きます。いずれかの列ヘッダーをクリックすると、その列に基づいてリストがソートされます。アルファベット順、アルファベット順、および元のリスト順の三つのソートモードがあります。


上へ移動 ()、**下へ移動** ()、および**削除** ()のボタンを使用して、画像のリストを編集および整理します。**選択画像ファイルをエクスポート** ボタン () をクリックして、画像ライブラリで選択した画像ファイルをファイルシステムに保存します。

例えば、画像をボタン上のアイコンとして利用することができます。次に、**ボタン** または **トグルボタン** オブジェクトの**設定** ウィンドウの**アイコン** リストから、plot.png などの画像を選択します。

プレビュー






この領域には、上の画像リストで選択した画像のプレビューが表示されます。

サウンド

サウンドノード()には、アプリケーションで使用できるサウンドファイル(.wav ファイル)のライブラリが含まれています。設定ウィンドウには、次のセクションがあります。


サウンドのリスト

このリストには、アプリケーションで使用可能な全てのサウンドが**名前**列に表示されます。説明列にオプションの説明を追加できます。いくつかのサンプルサウンドが常に含まれています。サウンドファイルを追加するには、**+** ボタンをクリックします。このボタンをクリックすると、サウンドを選択してリストに追加できるファイルブラウザが開きます。いずれかの列ヘッダーをクリックすると、その列に基づいてリストがソートされます。アルファベット順、アルファベットの逆順、および元のリスト順の三つのソートモードがあります。


サウンドのリストを編集して配置するには、**上へ移動** ()、**下へ移動** ()、および**削除** ()のボタンを使用します。**プレビュー** ボタン () をクリックして、そのサウンドを聞くことができます。**選択音声ファイルをエクスポート** ボタン () をクリックして、サウンドライブラリで選択したサウンドファイルをファイルシステムに保存します。





例えば、ボタンに設定されたアクションを、サウンドによって示すことができます。アプリケーションユーザーがボタンをクリックしたときにサウンドを再生するには、ボタンをコード行を追加するメソッドに接続します。このメソッドでは、サウンドファイルの名前を呼び出す必要があります。例えば、success.wav という名前のファイルからサウンドを再生するには、次の行をメソッドに追加します: `playSound("success");`;

ファイル

ファイルノード()には、アプリケーションで使用できるデータファイルのライブラリが含まれています。このライブラリに追加できるファイルには、ファイルタイプの制限はありません。設定ウィンドウには、次のセクションがあります。

ファイルのリスト

このリストには、アプリケーションで使用可能な全てのファイルが**名前**列に表示されます。**コピー**元列には、アプリケーションがコピーするファイルのフルパスと名前が含まれます。**説明**列にオプションの説明を追加できます。ファイルを追加するには、**ファイルをライブラリに追加**  ボタンをクリックします。このボタンをクリックするとファイルブラウザが開かれ、そこでファイルを選択してリストに追加できます。いずれかの列のヘッダーをクリックすると、その列に基づいてリストがソートされます。アルファベット順、アルファベットの逆順、および元のリスト順の三つのソートモードがあります。

ファイルのリストを編集して配置するには、**上へ移動**()、**下へ移動**()、および**削除**()のボタンを使用します。**選択ファイルをエクスポート**ボタン()をクリックして、ライブラリで選択したファイルをファイルシステムに保存します。

アプリケーションでは、(例えば data.txt ファイルを参照する場合、) `embedded:/// data.txt` を使用して追加したファイルを参照することができます。埋め込みファイルスキームの詳細については、[ファイルスキームとファイル処理](#)を参照してください。

アプリケーションの計画と準備

以下のセクションでは、アプリケーションを計画し準備するときの考え方について説明しています。通常は、既にお持ちの COMSOL マルチフィジックスの一つまたは複数のモデルが、アプリケーション開発の出発点となります。

アプリケーションの準備

アプリケーションビルダーを最大限に活用したい場合は、アプリケーションを作成する際にこれらの手順とガイドラインを検討してください。

アプリケーションに含めるべきものを検討してください。関心のあるパラメータは何ですか？ ユーザーが見たいと思っている出力は何ですか？ プロットは便利ですが、通常、最大応力、変位、温度などの特定の数値的結果が重要です。

ユーザーインターフェイスとそのコントロールとオブジェクトのスケッチを作成し、アプリケーションに含める必要がある入力の種類、メニュー、ボタン、プロットなどを纏めてみます。アプリケーションでフォームのレイアウトを作成します。

アプリケーションの入力として使用するパラメータ、出力および結果として使用する導出値とプロットを含んでいるアプリケーションの COMSOL マルチフィジックスモデルを作成します。導出された値とプロットを区別するために、記述説明的な名前を持つようにノードの名前を変更します。

アプリケーションに含める全てのパラメータに対して、説明欄に記述をするようにします。新規フォームウィザードでは、入力としてそれらを選択することができます。このウィザードでは、その説明が対応するテキストフィールドのラベルとして使用されます。

COMSOL マルチフィジックスモデルを解いて、ユーザーが関心のある出力を生成するためにはどのようなスタディーをアプリケーションに含めるのかを検討します。

COMSOL マルチフィジックスモデルを、アプリケーションの開始点として使用可能な MPH ファイルとして保存します。

アプリケーションビルダーには、ほとんどのニーズに適したアプリケーションを仕立てるための組み込みツールが含まれています。コーディングが必要なアプリケーションの特殊な側面があるかどうかを検討し、必要な場合は、使用する必要があるオブジェクトとプロパティに関する情報を準備します。次のステップは、アプリケーションビルダーを起動することです。

複数のプロットを使用してソリューションのさまざまな側面を示す場合、全てのプロットボタンがそのソリューションを再計算する必要があるかどうかを検討します。あるいは、ソリューションの計算は別のボタンを使用して行い、プロットボタンでは現在のソリューションデータを使用してプロットを更新するようにすることもできます。

モデルからのアプリケーション作成

以下のセクションでは、モデルからコードを抽出してアプリケーションをテストするのに役立つツールについて説明しています。

コードをクリップボードにコピー

埋め込みモデルのノードの内容を対応するコードとしてコピーすることによって、アプリケーションのカスタムメソッドを作成することができます。埋め込みモデルでノードを選択して右クリックすると、コンテキストメニューに**クリップボードにコードをコピー**というサブメニューが表示されます。

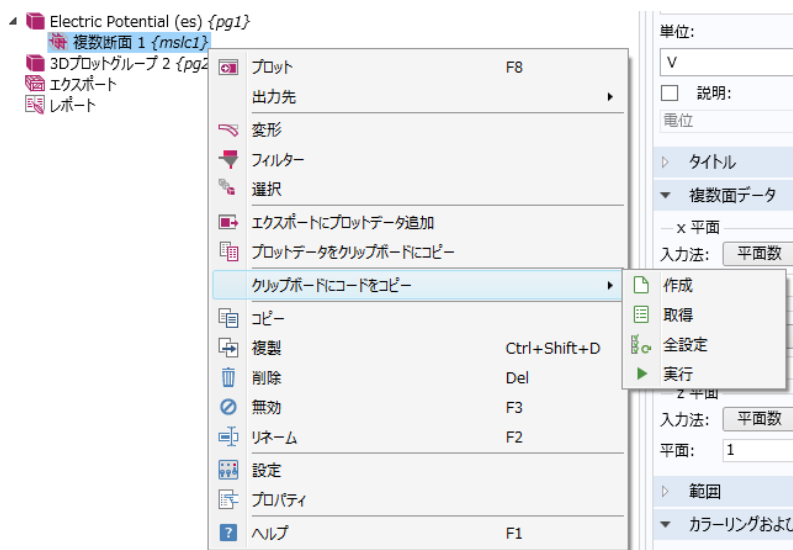


Figure 2-5: クリップボードにコードをコピーのサブメニューのオプション

サブメニューには、最大四つの選択肢が含まれています。全ての選択肢は、ソースコードをシステムクリップボードにコピーします。すなわち：

- **作成** (適用可能な場合)： 対応するオブジェクトを作成するためのコードスニペットを生成します。
- **取得**： 対応するオブジェクトにアクセスするためのコードスニペットが生成されます。
- **全設定**： コードスニペットを生成して、対応するオブジェクトのデフォルト以外のプロパティを設定します。
- **実行** (該当する場合)： 対応するシーケンスを実行するコードスニペットを生成します。


作成と**実行**のアクションは、それぞれ、作成できるノードと実行できるノードにのみ適用できます。

クリップボードにコードをコピーのアクションは、コードを簡単に記述できるようにするためのものです。特定のモデルエンティティにアクセスするためにどのコードを書き出すのかを把握するのは難しいかもしれませんが、**クリップボードにコードをコピー** > **取得**を利用してその結果をメソッドエディターウィンドウに貼り付けるだけで、対応するコードを知る必要はありません。同様に、モデルエンティティの特定のプロパティを設定する方法を理解するには、埋め込みモデルのプロパティを変更し、**クリップボードにコードをコピー** > **全設定**を利用して、その結果をメソッドエディターウィンドウに貼り付け、挿入されたコードを編集することもできます。**作成**と**実行**のアクションは同じ機能を提供しますが、それぞれはフィーチャの作成と実行のタスクに対応しています。





クリップボードにコードをコピー > **全設定**では、フィーチャの設定の中で直接変更されたプロパティのみが考慮されます。特に、ジオメトリシーケンスを挿入することによって設定されるプロパティは、考慮されません。

アプリケーションのテスト

ホームリボンタブで、テストセクションの**アプリケーションをテスト**()をクリックします。2番目のデスクトップウィンドウが開き、アプリケーションのテスト実行が開始されます。**アプリケーションをテスト**ボタンを再度クリックすると、実行中のアプリケーションが新しいアプリケーションのコピーに置き換えられ、ビルダーの最近の変更が考慮されます。




実行中のアプリケーションで作業をしてからビルダーで**アプリケーションをテスト**を再度クリックすると、アプリケーションのテスト実行までに行った保存されていない作業が失われます。

アプリケーションをテストして、ウェブブラウザでの動作を確認することもできます。**ウェブブラウザでテスト**ボタン()をクリックして、COMSOL サーバーを使用してウェブブラウザでアプリケーションを起動します。アプリケーションは、デフォルトブラウザの新しいブラウザウィンドウで開きます。**ウェブブラウザでテスト**ボタン()の右下にある下向きの矢印をクリックして、Google Chrome™、Firefox®、または Internet Explorer® (それらのブラウザをインストールしている場合)のどのブラウザでアプリケーションをテストするかを選択できるメニューを開きます。

実行中のアプリケーションへの変更の適用

ビルダーモデルに加えた変更は、実行中のアプリケーションに再適用せずに適用することができます。これは、テストが必要なメソッドを開発している場合や、レイアウトの変更を試したい場合に便利です。ホットコードスワップメカニズムは、実行中のアプリケーションをビルダーアプリケーションからの新しいコードおよび設定で更新するために使用されます。実行中のアプリケーションとビルダーモデルとの間に矛盾がある場合、ホットコードスワップは失敗し、代わりにアプリケーションを再起動する必要があります。

変更を適用するには、**アプリケーション**リボンタブに移動し、**変更を適用**()をクリックします。メソッド、イベント、宣言、フォーム、およびウィジェットを追加、編集、および削除できます。これには、レイアウト変更、編集コマンドシーケンス、およびラベルが含まれます。このボタンを使用して、プロット設定などのモデルを変更することもできます。モデルを変更すると、実行中のアプリケーションでモデルと矛盾する危険性があり、ホットコードスワップを防止する必要があります。そこで、実行中のアプリケーションにコードがスワップされる前にこの矛盾が検出された場合は、変更を適用せずにアプリケーションを実行し続けるか、アプリケーションを再起動するかを選択できるようになっています。

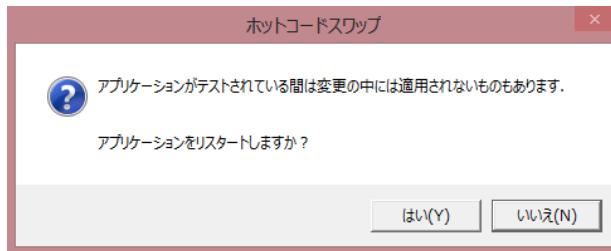
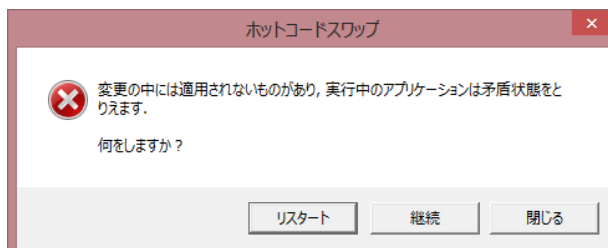


Figure 2-6: ホットコードスワップが失敗したときのエラーメッセージ

いくつかのコードがすでに実行中のアプリケーションにスワップされるまで、矛盾は検出されなくなります。アプリケーションの実行は継続できますが、その状態は一般的には定義されていません。下図のように、アプリケーションの実行、再起動、または終了を選択することができる場合もあります。



キーボードショートカット

次の表には、アプリケーションビルダーで一般的に使用できるキーボードショートカットを示しています。フォームウィンドウでフォームを操作する場合や、メソッドエディターウィンドウでメソッドを操作する場合に使用します。

TABLE 2-1: キーボードショートカット

ショートカット	アクション	一般	フォーム ウィンドウ	メソッド エディター
Ctrl+A	全て選択	✓	✓	✓
Ctrl+D	全て選択解除	✓		
Ctrl+C	コピー	✓	✓	✓
Ctrl+V	ペースト貼り付け		✓	✓
Ctrl+X	カット		✓	✓
Del	削除	✓	✓	✓
Ctrl+N	新規アプリケーションの作成	✓	✓	✓
Ctrl+S	アプリケーションの保存	✓	✓	✓
Ctrl+F8	アプリケーションのテスト	✓	✓	✓
Ctrl+Shift+F8	変更を適用	✓	✓	✓
Ctrl+R	コードを記録			✓
F1	ヘルプの表示	✓	✓	✓
F2	アプリケーションノードの名前の変更	✓		
F3	アプリケーションノードを無効化	✓		
F4	アプリケーションノードを有効化	✓		
F11	ノードへ			✓
Ctrl+1	ローカル変数宣言の作成			✓
Ctrl+7	コメントのオン/オフの切り替え			✓
Ctrl+up arrow	アプリケーションノードを上へ移動	✓		
Ctrl+down arrow	アプリケーションノードを下へ移動	✓		
Ctrl+Z	取り消し	✓	✓	✓
Ctrl+Y (Ctrl+Shift+Z on Mac)	やり直し	✓	✓	✓
F5	継続 (デバッグ時)			✓
F6	ステップ (デバッグ時)			✓
F7	ステップへ (デバッグ時)			✓
F8	シンタックスチェック			✓
Ctrl+F	アプリケーションで検索用語を検索、またはメソッドのテキストを検索と置換	✓		✓
Ctrl+K	オブジェクトやメニューへのショートカットの作成、使用、名前変更	✓	✓	
Ctrl+space, Ctrl+/ Ctrl+M	メソッドのコードの自動コンプリート			✓
Ctrl+M	一致する括弧間でカーソルを移動			✓
Ctrl+Shift+M	一致する括弧間のテキストを選択			✓
Ctrl+U	選択されたコードを小文字に変換			✓
Ctrl+Shift+U	選択されたコードを大文字に変換			✓
Ctrl+B	選択行のブレイクポイントを切り替え			✓

TABLE 2-1: キーボードショートカット

ショートカット	アクション	一般	フォーム ウィンドウ	メソッド エディター
Ctrl+scroll wheel up	メソッドウィンドウのコードをズームイン			√
Ctrl+scroll wheel down	メソッドウィンドウのコードをズームアウト			√
Ctrl+all arrow keys	スケッチモードでフォームオブジェクトの位置を少し微調整		√	
All arrow keys	スケッチモードでフォームオブジェクトの位置をさらに微調整、グリッドモードではセルをトラバースして選択		√	
Shift+all arrow keys	グリッドモードで複数のセルを選択		√	
Ctrl+Alt+A	アプリケーションビルダーウィンドウへ戻る	√		
Ctrl+Alt+M	モデルビルダーウィンドウへ戻る	√		
Ctrl+Alt+double-click	別のメソッドから呼び出されたメソッドのエディターウィンドウを開く			√
Alt+F4	ウィンドウを閉じる	√	√	√
Ctrl+F4	フォームまたはメソッドウィンドウを閉じる		√	√
Ctrl+Shift+F4	全てのフォームまたはメソッドウィンドウを閉じる		√	√
Ctrl+Pause (Command+. on Mac)	実行中のアプリケーションをテストするときは、メソッドの実行を停止	√		

フォームの操作

この章では、アプリケーションのカスタムユーザーインターフェースであるフォームを作成するためのツールについて説明します。フォームには、入力フィールド、グラフィックス、ボタンなど、さまざまなフォームオブジェクトを含めることができます。アプリケーションビルダーでは、アプリケーションのフォームを作成するために、フォームオブジェクトのスケッチや配列のためのインタラクティブなツールが備わっています。



以下に、この章の構成を示します。

- [はじめに](#)
- [フォームオブジェクト](#)

はじめに


フォームを作成するためのフォームとツールの概要


アプリケーションビルダーのフォームツールを使用すると、グラフィックスやリスト、テキスト入力フィールド、チェックボックス、ボタンなどのさまざまなフォームオブジェクトを含むフォーム（ペインまたはウィンドウとも呼ばれる）をインタラクティブに作成および編集できます。アプリケーションビルダーのデスクトップ環境には、以下の部分が含まれています。

- 新規フォームを作成し、フォームオブジェクトを挿入し、グリッドとレイアウトを制御するためのボタンが備わった**フォーム**というコンテキストualなリボンツールバー。
- **モデル**のブランチは含まれていないアプリケーションツリーが備わった**アプリケーションビルダー**ウィンドウ。
- グラフィックスやその他のフォームオブジェクトを挿入し、インタラクティブにフォームをデザインすることができる一つまたは複数の**フォームウィンドウ**（エディターウィンドウ）。アプリケーションの各フォームごとにフォームウィンドウを持ちます。**環境設定**ダイアログボックスの**アプリケーションビルダー**のページにある**終了前のエディターの最大数**の設定を使用して、同時に開くことができるエディターウィンドウの最大数を制御できます。エディターの数のチェックを無効にするには、関連するチェックボックスをオフにします。
- 選択したフォームまたはフォームオブジェクトの設定を行う**設定ウィンドウ**も備わっています。これらの設定は、フォームおよびフォームオブジェクトのプロパティを編集する場合に使用します。このウィンドウを開いたり閉じたりするには、**ホーム**リボンツールバーの**設定**ボタン()をクリックします。複数のフォームオブジェクトを選択すると、複数の**オブジェクトの設定**ウィンドウには、選択した全てのフォームオブジェクトに共通する設定が含まれます。通常は、**位置とサイズ**と**外観**セクションにある設定です。
- **フォーム 1** などとは別のウィンドウ（名前はプレビューしているフォームの名前を反映）に、そのフォームのプレビューを表示し、レイアウトとデザインを確認することができます。それを開いたり閉じたりするには、**フォーム**リボンツールバーの**プレビュー**ボタン()をクリックします。

フォームの操作と新規フォームウィザードの使用

フォームの操作を開始し、次の方法で**フォーム**リボンツールバーにアクセスできます。

- 既存のフォームウィンドウをクリックして選択します。
- アプリケーションツリーの**フォーム**ノードを右クリックし、**編集** ()を選択します。

新しいフォームを作成するには、**新規フォーム**ボタン()をクリックします。そこで**新規フォーム**というウィザードウィンドウが開かれ、そのウィザードを使用して次の手順を実行します。

- 1 **新規フォーム**ウィンドウでは、基本となるフォームオブジェクト（コンポーネント）による新しいフォームを作成できます。追加された全てのフォームオブジェクトについて、右側の**プレビュー**セクションのフォーム外観に表示されます。**フォームのタイトル**フィールド（最初のフォームの場合、デフォルトは**フォーム 1**）でフォームのラベルを、**フォーム名**フィールド（最初のフォームの場合、デフォルトは `form1`）でその名前を変更することができます。**トップにラベル追加**のチェックボックスを選択すると、入力フィールドやデータ表示オブジェクトのラベルを入力や表示の左に配置して表示する代わりに、上に配置変更することができます。後で追加の調整や修正を行うことができます。
- 2 **入力/出カタブ**をクリックして、ユーザーが入力として使用する**パラメータ**値を入力できるように入力フィールドを追加し、数値結果を表示できるように数値データ用の表示を追加します。**選択元**の下では、**宣言**の下にあるアプリケーションに定義されている適用可能な変数、**グローバル定義**の下にあるモデル内のいくつかのパラメータ、**コンポーネント** > **定義** の下にあるモデル内のいくつかの変数を選択して、入力フィールドを定義することができます。**データアクセス**を使用して他のモデルプロパティを追加することもできます。プロパティに応じて、その入力にはラベルとテキストフィールド、チェックボックス、あるいはコンボボックスの形体として置かれます。デフォルトのラベルには、**パラメータ**と**変数**に対して定義されている説明が設定されます。可能な場合、ウィザードでは**値に単位を追加**の設定の入力フィールドが作成されます。


また、**結果** > **派生値**の下にあるモデルの評価値を選択して、数値データとして表示することもできます。Enter キーを押すか、選択した入力または出力をダブルクリックするか、または**選択対象追加** ボタンをクリックして、入力フィールドまたはデータ表示をフォームおよび**選択**の下のリストに追加することができます。**選択**のリストから項目を削除するには、Enter キーを押すか、ダブルクリックするか、**選択対象を削除** ボタンをクリックします。

3 **グラフィックスタブ**をクリックして、次のグラフィックスウィンドウ(プロット)をフォームに追加します。





- **コンポーネン** > **定義**の下で、使用可能な任意の明示的な選択ノードをグラフィックスオブジェクトの初期グラフィックスコンテンツのソースとして追加します。
- **コンポーネン**の下に、ジオメトリをプロットするための**ジオメトリ**を追加します。
- **コンポーネン**の下で、メッシュをプロットするための**メッシュ**を追加します。
- **結果**の下で、使用可能なプロットグループと**エクスポート**の下のノードを追加します。

ダブルクリックするか、**選択対象追加** ボタンをクリックして、フォームにグラフィックスを追加し、**選択**のリストに追加します。**選択**のリストから項目を削除するには、ダブルクリックするか、**選択対象を削除** ボタンをクリックします。




- 4 フォームにボタンを追加するには、**ボタンタブ**をクリックします。デフォルトでは、追加されるボタンは大きなボタンです。ユーザーが**モデル**ブランチのアプリケーションで実行できる以下のアクションのボタンを追加することができます。
- **コンポーネント**の下に、**ジオメトリ (Geometry) をプロット**を追加してジオメトリのプロットを更新するボタンを追加するか、**メッシュ (Mesh) をプロット**を追加してメッシュのプロットを更新するボタンを追加します。
 - **コンポーネント**の下に、**メッシュ (Mesh) をプロット**を追加してメッシュのプロットを更新するボタンを追加します。
 - **計算 Study** を追加して、スタディを計算するためのボタンを追加します (そのスタディで定義されたシミュレーションを実行し、ソリューションのデフォルトプロットを提示します)。
 - **結果**の下に、モデルで定義された各プロットグループをプロットするためのプロットボタンを追加します (例えば、固体力学モデルで応力をプロットする応力プロットグループの場合の**応力プロット**)。このようなボタンは、プロットをアプリケーションのグラフィックスウィンドウに送信します。**エクスポート**の下のノードでデータをエクスポートするためのボタンを追加することもできます。

フォームの下に、アプリケーションの既存のフォームを表示するボタンを作成するために、例えば **Show form1** ボタン () を追加します。








GUI コマンド > ファイルコマンドの下では、ファイル関連の操作に以下のボタンを追加することができます。

- **アプリケーションを保存**ボタン () : 現在の名前でアプリケーションを保存します。
- **名前を付けてアプリケーションを保存**ボタン () : アプリケーション (拡張子.mph) を新しい名前で保存します。
- **サーバー上でアプリケーションを保存**ボタン () : 現在の名前でサーバーにアプリケーションを保存します。
- **サーバー上でアプリケーションを別名で保存**ボタン () : アプリケーション (拡張子.mph) を新しい名前でサーバーに保存します。

アプリケーションが COMSOL サーバー上で実行されている場合、**サーバー上でアプリケーションを保存**と**サーバー上でアプリケーションを別名で保存**のコマンドは、現在のアプリケーションを COMSOL サーバーのアプリケーションライブラリに新規アプリケーションとして保存します。



- **ファイルを開く**ボタン () : アプリケーション内のファイルを開きます。
- **名前を付けてファイルを保存**ボタン () : ファイルを選択してクライアントアプリケーションにダウンロードできます。
- **アプリケーションを終了**ボタン () : アプリケーションを終了します。


GUI コマンド > グラフィックスコマンドの下で、グラフィックス関連のプロパティと操作のために、以下のボタンを追加することができます。


- **画面にわたってズーム**ボタン () : プロットされているオブジェクトまたはジオメトリの範囲をズームインできます。
- **現在のビューをリセット**ボタン () : グラフィックスウィンドウの現在のビューを、アプリケーションの起動時に表示されるビューにリセットすることができます。
- **シーンライト**ボタン () : グラフィックスウィンドウにシーンライトを追加します。
- **透明度**ボタン () : 3D グラフィックスオブジェクトを透明にします。
- **印刷**ボタン () : グラフィックスオブジェクトのコンテンツを印刷します。
- **全選択**ボタン () : グラフィックスウィンドウ内の全てのオブジェクトを選択します。
- **選択対象消去**ボタン () : グラフィックスウィンドウ内の全てのオブジェクトの選択をクリアします。

全てのグラフィックコマンドでは、コマンドを適用するグラフィックスオブジェクトの名前を引数として追加します。

GUI コマンド > モデルコマンドの下で、以下のボタンをモデル関連の操作に追加することができます。

- **全解除**ボタン () : アプリケーション内の全てのソリューションをクリアします。
- **全メッシュ消去**ボタン () : アプリケーションの全てのメッシュをクリアします。

ダブルクリックするか、**選択対象追加**  ボタンをクリックして、フォームにグラフィックスを追加し、**選択リスト**に追加します。




選択のリストから項目を削除するには、ダブルクリックするか、**選択対象を削除**  ボタンをクリックします。

5 OK をクリックしてウィザードを終了すると、定義したフォームのコンポーネントが生成されます。

データアクセス

アプリケーションビルダーでデータアクセス機能を使用することによって、モデルに依存するデータとプロパティ、および実行中のアプリケーションから変更可能なアプリケーション固有のプロパティを追加することができます。これは、**新規フォーム**ウィザードでアクセスする定義済みのモデルのデータとモデルのプロパティに加えて追加されます。データアクセスがアクティブな場合は、**コンポーネント**ノードの下にある多くのノードの**設定**ウィンドウ内の適用可能なデータとプロパティ、およびアプリケーション内のいくつかのフォーム、フォームオブジェクト、メニューとツールバー項目の**設定**ウィンドウ内で、該当する設定箇所の左側に緑色のチェックボックスが表示されます。このチェックボックスを選択すると、データソースとして選択することができ、**新規フォーム**ウィザードの**入力/出力**ページの**選択元**の下にあるツリー、および適用可能なフォームオブジェクトの**ソース**セクションのツリーで、対応するプロパティを追加することができます。対応するプロパティをツリーの**選択元**に追加します。データのタイプに応じて、テキストラベル、入力フィールド、チェックボックス、あるいはコンボボックスとして追加されます。

データアクセスをアクティブにするには、






- **ホーム**ツールバーで**モデルビルダー** () をクリックし、必要に応じて**デベロッパ**タブをクリックして**データアクセス** () をクリックします。
- 該当するフォームオブジェクトの**ソース**セクションで、**モデルビルダー**へスイッチし**データアクセスをアクティブ**のボタン () をクリックして、データアクセスが有効になっている**モデルビルダー**ウィンドウに直接移動します。

フォームツールバー

フォームのコンテキストリボンツールバーは、フォームとフォームオブジェクトの作成と編集のための機能へのアクセスを提供してくれています。このツールバーは、フォームウィンドウがアクティブなときに使用できます。このセクションでは、**フォーム**ツールバーのボタンについて簡単に説明しています。



メインセクション

このセクションには、さまざまなウィンドウに移動して新しいフォームとメソッドを作成するための以下のボタンが含まれています。

- **モデルビルダー**ボタン () : アプリケーションビルダーからモデルビルダーウィンドウの標準の COMSOL デスクトップに切り替えます。
- **新規フォーム**ボタン () : **新規フォーム**ウィザードを使用して新しいフォームを作成します。[フォームの操作と新規フォームウィザードの使用](#)を参照してください。
- **新規メソッド**ボタン () : 新しい**メソッド**ノードまたは**モデルメソッド**ノードを作成し、そのコードを新しいエディタータブで開きます。[メソッドノードとメソッドエディターウィンドウ](#)を参照してください。
- **設定**ボタン () : **設定**ウィンドウを開いたり閉じたりするためのボタンです。
- **プレビュー**ボタン () : アプリケーションのフォームとメソッドをライブプレビューしてくれる**プレビュー**ウィンドウを表示したり非表示にしたりするボタンです。**プレビュー**ウィンドウでは、スクロールしてアプリケーションの全てのフォームとメソッドのプレビューを確認することができます。これは、例えば、フォームとやりとりするメソッドを操作している場合などに便利です。**プレビュー**ウィンドウでフォームのプレビューを表示するには、**アプリケーションビルダー**ウィンドウでフォームまたはメソッドノードを選択します。

フォームセクション

このセクションには、新規フォームを作成し、現在のフォームにフォームオブジェクトを挿入するための以下のボタンが含まれています。

- **新規フォームボタン**(): **新規フォームウィザード**を使用して新しいフォームを作成します。[フォームの操作と新規フォームウィザードの使用](#)を参照してください。
- **オブジェクトの挿入ボタン**(): 全てのタイプのフォームオブジェクトから選択して現在のフォームに追加するためのメニューを開きます。[フォームオブジェクト](#)を参照してください。



レイアウトセクション

このセクションでは、スケッチレイアウトとグリッドレイアウトを切り替えることができます。

- グリッドモードに切り替えるには、**グリッドボタン**()をクリックします。[グリッドモード](#)を参照してください。
- スケッチモードに切り替えるには、**スケッチボタン**()をクリックします。[スケッチモード](#)を参照してください。










スケッチセクション

このセクションには、スケッチモードで使用できるツールのボタンが含まれています。グリッドモードでは、これらのボタンは無効になっています。

- フォームウィンドウでオーバーレイグリッドを追加または削除するには、**グリッドライン表示ボタン**()をクリックします。
- 選択したフォームオブジェクトを整列させるツールを含むメニューを開くには、**アレンジボタン**()をクリックします。




グリッドセクション


このセクションには、グリッドモードで使用できるツールのボタンが含まれています。スケッチモードでは、これらのボタンは無効になっています。また、対応するアクションが可能となるグリッド内のセルが選択されていない限り、ほとんどのボタンは無効になっています。以下に、含まれているボタンを掲載しています。

- **行設定ボタン**()： 選択した行の行サイズの設定メニューから選択します。[行と列の設定](#)を参照してください。
- **列設定ボタン**()： 選択した列の列サイズの設定メニューから選択します。[行と列の設定](#)を参照してください。
- **挿入ボタン**()： 選択した行または列の隣に行または列を挿入します。開かれるメニューから、行を上または下に挿入するか、列を左または右に挿入するかを選択します。[行と列の挿入と削除](#)を参照してください。
- **削除ボタン**()： 現在の行または列を削除します。[行と列の挿入と削除](#)を参照してください。
- **アラインボタン**()： 開かれるメニューに、行と列の中のコンテンツをアライン(整列)したり全幅にするためのツールが含まれています。[フォームオブジェクトの整列\(アライン\)](#)を参照してください。
- **セルをマージボタン**()： 選択した複数のセルをマージします。[セルのマージと分割](#)を参照してください。
- **セル分割ボタン**()： 選択したマージされているセルを分割します。[セルのマージと分割](#)を参照してください。
- **サブフォームを抽出ボタン**()： 現在のフォームから選択したセルを抽出し、新規フォームに挿入します。[サブフォームの抽出](#)を参照してください。
- **行および列ボタン**()： グリッド内の行数と列数を指定できるダイアログボックスを開きます。[グリッドサイズの変更](#)を参照してください。

テストセクション




このセクションには、アプリケーションをテストするための以下のボタンが含まれています。

- **アプリケーションをテストボタン**()： 別のウィンドウでアプリケーションを起動してテストすることができます。[アプリケーションのテスト](#)を参照してください。
- **変更を適用ボタン**()： 実行中のアプリケーションにコード変更をコンパイルして適用します(ホットコードスワップと呼ばれます)。[実行中のアプリケーションへの変更の適用](#)を参照してください。
- **プレビューフォームボタン**()： フォームのプレビューを含むウィンドウを開きます。[フォームのプレビューとテスト](#)を参照してください。

- **ウェブブラウザでテストボタン**(): ウェブブラウザでアプリケーションをテスト実行します。[アプリケーションのテスト](#)を参照してください。

ビューセクション

ビューセクションには、アプリケーションビルダーのデスクトップウィンドウでビューを再配置するための次のボタンがあります。



- **タイトル表示**()と**移動先**()ボタン: アプリケーションビルダーのウィンドウを再配置します。
- **デスクトップリセット**ボタン(): デスクトップのレイアウトをデフォルトの状態にリセットします。

フォームウィンドウのレイアウトモード

フォームウィンドウ(エディターウィンドウ)で作業するときを使用することができるレイアウトモードには、以下の二つの種類があります。

- **スケッチモード**: フォームオブジェクトを自由に配置し、ドラッグして簡単に整理することができます。スケッチモードは、所望するおおまかなレイアウトをすばやく作成するための出発点として役立ちます。但し、スケッチモードで表示される内容は、アプリケーションの実行時に表示される内容と完全には一致していません。
- **グリッドモード**: フォームの最終的なレイアウトがどのように表示されるかを詳細に制御することができます。

スケッチモードでは、含まれる各フォームオブジェクトの絶対座標、高さ、幅から行と列のレイアウトが計算されます。グリッドモードでは、含まれているフォームオブジェクトの行と列のサイズを指定できます。

モードを選択するには、**フォームリボンツールバー**の**レイアウトグループ**で**スケッチ**ボタン()または**グリッド**ボタン()をクリックします。各モードの詳細については、以降のセクションを参照してください。

環境設定のダイアログボックスの**フォーム**ページで、デフォルトとして使用するレイアウトモードを指定することができます。

フォームウィンドウから他のフォームウィンドウを開く

フォームオブジェクトの中には、グリッドパネルのようにそれ自体がフォームであるものや、カードスタックのようにフォームを含むものもあります。このようなフォームの新しい**フォーム**ウィンドウを開くには、それらのオブジェクトを Alt キーを押しながらクリックします。

フォームオブジェクトのドラッグとコピー

どちらのレイアウトモードでも、フォームオブジェクトをドラッグ & ドロップして移動することができます。代わりにフォームオブジェクトのコピーを作成して新しい場所にドラッグ & ドロップする場合には、元のフォームオブジェクトをドラッグし、Ctrl キーを押しながら目的の場所でコピーを貼り付けます。Esc を押すか、フォームエディターの外にフォームオブジェクトをドロップすると、ドラッグ操作を中止できます。


スケッチモード

スケッチモードでは、フォームオブジェクトを自由に配置することができるキャンバスが使用されます。但し、COMSOL マルチフィジックスは、実行中のアプリケーションのユーザーインターフェイスを生成するときにオブジェクトをグリッドに配置します。このグリッドでは、完全に自由に配置することはできません。従って、スケッチモードで作成するレイアウトは、アプリケーションの実行時のレイアウトとおおよそ同じではありません。スケッチモードを開始点として使用して所望するおおまかなレイアウトをすばやく作成してから、グリッドモードに切り替えて微調整します。これにより、スケッチモードによるフォームオブジェクトの配置に基づいてグリッドが作成されます。



フォームオブジェクトの選択と追加

フォームオブジェクトを追加するには、**フォームリボンツールバー**にある**オブジェクトを挿入**のボタンをクリックし、そこで開かれるパレットからオブジェクトを選択します(使用可能なフォームオブジェクトの詳細については、[フォームオブジェクト](#)を参照)。すると、選択したタイプのフォームオブジェクトがキャンバスに追加され、それを必要な場所にドラッグすることができます。


フォームオブジェクトを追加した場合には、そのフォームオブジェクトが選択されます。別のフォームオブジェクトを選択するには、そのフォームオブジェクトをクリックします。Ctrl キーを押しながらクリックするか、クリックとドラッグによるボックスでいくつかのフォームオブジェクトを囲むことで、複数のフォームオブジェクトを選択することができます。Shift キーを押しながらドラッグすると、囲まれたフォームオブジェクトが現在の選択範囲に追加されます。







フォームオブジェクトを削除するには、フォームオブジェクトを選択し、Delete キーを押すか、クイックアクセスツールバーの削除ボタン()をクリックします。

グリッド線の表示とグリッドへのスナップ

フォームを分割し、フォームオブジェクトをスナップ可能なグリッドのグリッド線を表示するには、**フォーム**リボンツールバーの**スケッチ**セクションにある**グリッドライン表示**ボタン()をクリックします。ハンドルを使用してフォームオブジェクトのサイズをインタラクティブに変更すると、フォームオブジェクトがグリッドにスナップされます。再度**グリッドライン表示**ボタン()をクリックすると、グリッドラインの表示とグリッドスナップの表示がオフになります。**フォーム**ノードの**設定**ウィンドウの**スケッチグリッド**セクションで、グリッドとスナップの設定を制御します。

フォームオブジェクトのアレンジ

フォームオブジェクトをドラッグすると、オブジェクトを互いに対して整列させるのに役立つ線が表示されます。また、**フォーム**リボンツールバーの**スケッチ**セクションにも**アレンジ**メニュー()があり、そこにフォームオブジェクトを整列するための以下のツールがあります。

- **左にアライン**()： 選択したフォームオブジェクトを左端のオブジェクトの位置に左揃えします。
- **アライン中心**()： 選択したフォームオブジェクトを水平方向に中央揃えします。
- **右にアライン**()： 選択したフォームオブジェクトを右端のオブジェクトの位置に右揃えします。
- **上にアライン**()： 選択したフォームオブジェクトを一番上方向のオブジェクトの位置に上揃えします。
- **中央にアライン**()： 選択したフォームオブジェクトを垂直方向に中央揃えします。
- **アラインボタン**()： 選択したフォームオブジェクトを一番下方向のオブジェクトの位置に下揃えします。

グリッドモード

COMSOL マルチフィジックスでは、実行中のアプリケーションのユーザーインターフェースを作成するときにグリッドを使用します。グリッドモードでは、グリッドを編集し、フォームオブジェクトをグリッド内の任意の位置に配置して、目的のレイアウトを取得することができます。

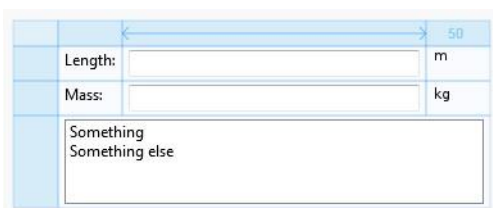


Figure 3-1: フォームオブジェクトを含むグリッドの例。グリッド内のセルは、複数の行と列にまたがるすることができます。例えば、リストボックスが置かれた一番下の行のセルは、三つの列全体にまたがっています。

フォームオブジェクトの追加と選択

新しいフォームオブジェクトを追加するには、まずグリッド内の空のセルを選択します。次に、**フォーム**リボンツールバーにある**オブジェクトを挿入**のボタンをクリックし、そこで開かれるパレットからオブジェクトを選択します(使用可能なフォームオブジェクトの詳細については、[フォームオブジェクト](#)を参照)。新しいオブジェクトがセルに挿入されるか、またはセルが選択されていない場合には、最初の空のセルに挿入されます。空のセルがない場合には、ソフトウェアは新しい行を追加し、その行の最初のセルにフォームオブジェクトを配置します。

フォームオブジェクトを追加すると、そのフォームオブジェクトが選択されます。別のフォームオブジェクトまたは空のセルを選択するには、そのセルをクリックします。Ctrl キーを押しながらクリックすると、複数のフォームオブジェクトとセルを選択できます。また、ボックスをドラッグして複数のフォームオブジェクトとセルを選択することもできます。セルを選択するためにボックスをドラッグする場合は、空のセルまたはグリッドの外側でドラッグを開始するときにその機能が有効になります。

コピー、複製、カット、および削除

追加した全てのフォームオブジェクトについて、フォームオブジェクトのセル、セルのグループ、または列や行全体を右クリックすると、以下の操作を含むコンテキストメニューが開かれます。

- **カット** (✂): フォームオブジェクトをフォームから切り取り (削除し)、それを貼り付けるために格納することができます。Ctrl + X のキー操作を使用することもできます。
- **コピー** (📄): フォームオブジェクトのコピーを作成します。Ctrl + C のキー操作を使用することもできます。
- **複製** (📄➕): フォームに直接追加されるフォームオブジェクトの複製を作成します。
- **削除** (🗑): フォームオブジェクトを削除します。また、Delete キーを押してフォームオブジェクトを削除できます。
- **設定** (⚙): フォームオブジェクトの**設定**ウィンドウを開きます。複数のオブジェクトを選択している場合には、共通プロパティの**設定**ウィンドウを開きます。
- **ヘルプ** (❓): **ヘルプ**ウィンドウを開き、該当する場合はフォームまたはフォームオブジェクトに関する情報を表示します。

例えば、データ変更で実行されるメソッドをサポートするフォームオブジェクトを考えた場合、そのコンテキストメニューには、ローカルメソッドを作成するための**ローカルメソッド作成**、または既存のローカルメソッドやその他のメソッドをエディターウィンドウで開くための**ローカルメソッド編集**や**メソッド編集**が含まれています。

フォームウィンドウのキャンバスを右クリックすると、以下の追加アクションを含むコンテキストメニューが表示されます。

- **プレビューフォーム** (🎥): フォームのコンテンツをプレビューするアプリケーションウィンドウを起動します。
- **ボタンをペースト** (📄): 例えば、コピーまたは切り取ったフォームオブジェクト (この場合はボタン) を貼り付けるために使用します。Ctrl + V のキー操作を使用することもできます。

矢印キーを使用して、グリッド内のセル選択を移動していくことができます。この走査させられるセル選択は、一度に一つだけです。Shift キーを押しながら矢印キーを使用して複数のセルを選択することもできます。

フォームオブジェクトの移動

フォームオブジェクトをドラッグ & ドロップして別のセルに移動することができます。セルをクリックして、セル内のフォームオブジェクトを選択します。次に、オブジェクトをドラッグして配置するセルにドロップします。フォームオブジェクトをドロップしたセルが別のオブジェクトによって占有されている場合、そのオブジェクトはドラッグしたオブジェクトによって空になったセルに移動されます。


これにより、二つのフォームオブジェクト間の場所を簡単に変更できます。




行と列の設定


行と列のサイズを制御するのは、三つの方法があります。行や列の制御方法を選択するには、まず、グリッドの左または上のヘッダーをクリックして行または列を選択します。下図は列を選択した場合を示しています。






Figure 3-2: グリッドレイアウトモードを使用している場合の列の選択

列の幅を制御する方法を変更するには、**フォーム**リボンツールバーの**列設定**のメニュー()を使用します。同じメニューは、フォームウィンドウで列を選択したときにも使用できます。また、一つまたは複数のセルを選択して右クリックした場合にも、そこで表示される**列**というサブメニューとして使用できます。以下のオプションを使用できます。


- **幅合わせ**()： 列内の最も幅の広いオブジェクトと同じ幅にします。
- **列を自動拡大縮小**()： グリッドのコンテナの幅が変更されると、列の幅が変更されます。コンテナとは、フォームが使用されるウィンドウ、または親フォーム内の列が該当します。一つまたは複数の列を自動拡大縮小するということは、利用可能なスペースに合わせてサイズを調整するユーザーインターフェースを作成するのに適しています。
- **幅固定**()： 列の幅を正確な数値で設定します。幅は、列のヘッダー内の列の区切り線部分をドラッグすることで調整することができます。このオプションを使用する場合、その列の幅による使用可能なスペースに列内の全てが収まることを確認してください。また、アプリケーションが実行される可能性のある全てのクライアントで、全てのオブジェクトのサイズが同じにはならないことを考慮してください。





同様に、行の高さを制御する方法を変更するには、**フォーム**リボンツールバーの**行設定**のメニュー()を使用します。同じメニューは、フォームウィンドウで行を選択したときにも使用できます。また、一つまたは複数のセルを選択して右クリックした場合にも、そこで表示される**行**というサブメニューとして使用できます。以下のオプションを使用できます。


- **高さ合わせ**()： 行内の最も高いオブジェクトと同じ高さにします。
- **行を自動拡大縮小**()： グリッドのコンテナの高さが変更されると、行の高さが変更されます。コンテナとは、フォームが使用されるウィンドウ、または親フォーム内の行が該当します。
- **固定行**()： 行の高さを正確な数値で設定します。行は、行のヘッダー内の行の区切り線部分をドラッグすることで調整することができます。このオプションを使用する場合、その行の高さによる使用可能なスペースに行内の全てが収まることを確認してください。また、アプリケーションが実行される可能性のある全てのクライアントで、全てのオブジェクトのサイズが同じにはならないことを考慮してください。



ヘッダーには行と列の設定が表示されます。行の自動拡大縮小を使用すると、ヘッダーには、[Figure 3-2](#)の中央の列のように矢印記号が表示されます。固定行を使用すると、サイズがヘッダーに表示されます([Figure 3-2](#)の右側の列に示されています)。

行と列の挿入と削除

行または列を選択している状態のときに、**フォーム**リボンツールバーの**グリッド**セクション、または選択した行または列を右クリックした際のコンテキストメニューにある**挿入**メニュー()を使用することによって、行または列を挿入することができます。以下のオプションを選択することができます。


- **上に挿入**()： 選択した行の上側に新しい行を挿入します。
- **下に挿入**()： 選択した行の下側に新しい行を挿入します。
- **左に挿入**()： 選択した列の左側に新しい列を挿入します。
- **右に挿入**()： 選択した列の右側に新しい列を挿入します。

削除メニュー()を使用して、行や列を削除することもできます。









- **行を削除**()： 選択した行をフォームから削除します。
- **列を削除**()： 選択した列をフォームから削除します。

該当する削除コマンドは、選択した行または列を右クリックした際に表示されるコンテキストメニューでも使用することができます。

フォームオブジェクトの整列(アライン)

フォームリボンツールバーの**グリッド**セクションにある**アライン**メニュー()のツールを使用して、セル内のフォームオブジェクトのアラインメントを調整することができます。また、一つまたは複数のセルを選択して右クリックした場合にも、そこで表示さ

れる**アライン**というサブメニューとして使用できます。アラインメントを変更するには、まず、アラインメントを行うフォームオブジェクトを含むセルを選択し、以下のいずれかのアラインのオプションを選択します。

- **水平全幅** (): フォームオブジェクトのサイズを、その列内の水平方向の全ての空き領域を占めるようにします。全幅にする際は、フォームオブジェクトに割り当てられている幅のサイズは考慮されず、代わりに列のサイズが与えられます。
- **左にアライン** (): フォームオブジェクトをセル内の左端に揃えます。
- **アライン中心** (): フォームオブジェクトをセル内の水平方向の中心に揃えます。
- **右にアライン** (): フォームオブジェクトをセル内の右端に揃えます。
- **垂直全幅** (): フォームオブジェクトのサイズを、その行内の垂直方向の全ての空き領域を占めるようにします。全幅にする際は、フォームオブジェクトに割り当てられている高さのサイズは考慮されず、代わりに行のサイズが与えられます。
- **上にアライン** (): フォームオブジェクトをセル内の上端に揃えます。
- **中央にアライン** (): フォームオブジェクトをセル内の垂直方向の中央に揃えます。
- **下にアライン** (): フォームオブジェクトをセル内の下端に揃えます。

フォームオブジェクトのサイズ調整

いくつかのフォームオブジェクトはドラッグによってサイズを変更することができますが、全てのオブジェクトのサイズを変更することはできません。一つの例はラベルであり、サイズはラベルのテキストによって決められます。

フォームオブジェクトのサイズを変更するには、まずそれを含むセルをクリックしてそのフォームオブジェクトを選択します。下図は、選択したリストボックスを示しています。

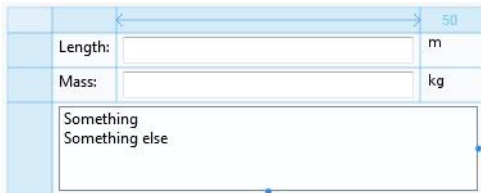




Figure 3-3: 選択したリストボックスをドラッグすると幅と高さを変更できます。


選択すると、リストボックスの右と下の辺にハンドルが表示され、これをドラッグすることによってサイズを調整することができます。オブジェクトの水平方向のアラインメントが全幅に設定されている場合、水平方向にドラッグすると、アラインメントは左に変更されます。同様に、垂直方向にドラッグすると、アラインメントは上に変更されます。青のガイドラインの表示によって、そのオブジェクトを同じフォーム内の他のオブジェクトと整列させるのが容易です。

セルのマージと分割

フォームオブジェクトが複数の行または列にまたがるようにするためには、セルをマージします。セルをマージするには、セルを選択し、**フォーム**リボンツールバーの**グリッド**セクションにある**セルをマージ**のボタン () をクリックします。マージする場合、マージされるセルにフォームオブジェクトがあったとしても一つでなければなりません。これは、新しいセルには一つのフォームオブジェクトしか含むことができないためです。

セルのマージの逆は、複数の行または列にまたがって結合されたセルを分割することです。セルを分割するには、セルを選択し、**フォーム**リボンツールバーの**グリッド**セクションで**セル分割** () をクリックします。元のマージされたセルにフォームオブジェクトがあった場合は、新しく分割された複数のセルの左上のセルに配置されます。

サブフォームの抽出

フォームから新しいフォームに矩形領域 (サブグリッド) を抽出する場合は、抽出したいグリッドの領域を選択し、右クリックして**サブフォームを抽出** () を選択します。また、**フォーム**リボンツールバーの**グリッド**セクションにある**サブフォームを抽出**のボタンをクリックすることもできます。抽出されたサブグリッドは、元のフォームのその領域にあるフォームオブジェクトとともに新しいフォームに表示され、それに対応するフォームウィンドウが開かれます。元のフォームのセルは、新しいフォームを参照する**フォーム参照**オブジェクトに変更されます。

グリッドとフォームのサイズ変更

拡大可能な行または列が少なくとも1つある場合は、フォーム内のグリッド全体のサイズを変更できます。拡大する行または列がコンテナによって指定された利用可能なスペースにサイズ調整される場合、使用可能なスペースが変更されたときのフォームの外観を確認するためにサイズ変更を使用することができます。

グリッドの左上隅をクリックすると、グリッド全体が選択されてグリッドのサイズを変更できることを示すハンドルが表示されます。下の画像は、枠の右側にドラッグハンドルがあるように選択されたグリッドを示しています。ボーダーラインに沿ってどのボーダーでも掴んでサイズを変更することができます。

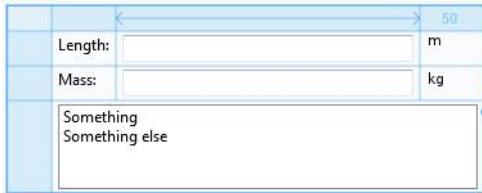




Figure 3-4: 右側のグリッドハンドルは、グリッドのサイズを変更できることを示しています。

最初にグリッドモードを使用するサイズ変更可能なフォームを選択せずに、フォームのサイズ変更をすることができ、行数または列数を増やすことができます。右または下の枠線をドラッグして、フォームのサイズを変更します(サイズ変更が利用可能ときにサイズ変更カーソルが表示されます)。フォームのサイズ設定が自動に設定されている場合は、フォームエディタ内でサイズ変更したフォームサイズが初期サイズとして使用されます。

グリッドサイズの変更

グリッド内の行と列の数を変更するには、フォームリボンツールバーのグリッドセクションの行および列ボタン()をクリックします。開かれる行および列のダイアログボックスで、行と列フィールドに行と列の数を指定してOKをクリックします。新しいサイズが現在のグリッドサイズよりも小さい場合は、グリッドの右下隅から行と列が削除されます。また、グリッドのサイズを変更すると、新しいグリッドに収まらないフォームオブジェクトも削除されます。

フォームのプレビューとテスト

フォームの外観を確認し、入力フィールドやその他のフォームオブジェクトが指定通りに動作するかどうかをテストするには、フォームを別のウィンドウで起動します。これを行うには、フォームリボンツールバーのプレビューフォーム() ボタンをクリックします。

フォームオブジェクトでのメソッドの実行

ボタンなどの一部のフォームオブジェクトでは、ユーザーがボタンをクリックしたときに実行するコマンドを追加できます。これらのコマンドには、事前に定義されたコマンド(プロットなど)を使用できます。入力フィールドなどの他のフォームオブジェクトでは、フォームオブジェクトがフォーカスを取得または喪失したときに発生するイベントのメソッドを追加できます。フォームオブジェクトで実行するメソッドを追加すると、フォームウィンドウ内のフォームオブジェクトの外観にメソッド接続が表示されます。

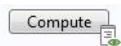


Figure 3-5: ユーザーがクリックしたときにメソッドが実行されることを示す表示付きの Compute ボタン


メソッドにコンパイルエラーがある場合、メソッド接続を示す表示部分に赤い十字が表示され、コンパイルエラーが修正されるまでボタンをテストできないことを示します。



Figure 3-6: メソッドにコンパイルエラーがあることを示す Compute ボタン

フォームオブジェクト

フォームオブジェクトの概要

フォームリボンツールバーの**オブジェクトを挿入**ボタン()をクリックすると、以下のタイプのフォームオブジェクトを含むパレット(ギャラリー)が表示されます。


- **入力フォームオブジェクトのセット:**
 - **入力フィールド**オブジェクト: ユーザー入力用
 - **ボタン**オブジェクト
 - **トグルボタン**オブジェクト
 - **チェックボックス**オブジェクト
 - **コンボボックス**オブジェクト
- **ラベルのセット:**
 - **テキストラベル**オブジェクト
 - **単位**オブジェクト
 - **方程式**オブジェクト
 - **ライン**オブジェクト
- **表示オブジェクトのセット:**
 - **データ表示**オブジェクト
 - **グラフィックス**オブジェクト
 - **ウェブページ**オブジェクト
 - **画像**オブジェクト
 - **ビデオ**オブジェクト
 - **進捗バー**オブジェクト
 - **ログ**オブジェクト
 - **メッセージログ**オブジェクト
 - **結果テーブル**オブジェクト
- **サブフォームのセット:**
 - **フォーム**オブジェクト
 - **フォームコレクション**オブジェクト
 - **カードスタック**オブジェクト
 - **カード**オブジェクト: カードスタックオブジェクトから追加や編集が可能
- **コンボジットフォームオブジェクトのセット:**
 - **ファイルインポート**オブジェクト
 - **情報カードスタック**オブジェクト: アプリケーション情報カードの表示用
 - **配列入力**オブジェクト
 - **ラジオボタン**オブジェクト
 - **選択入力**オブジェクト

• **その他のフォームオブジェクトのセット:**

- テキストオブジェクト
- リストボックスオブジェクト
- テーブルオブジェクト
- スライダーオブジェクト
- ハイパーリンクオブジェクト
- ツールバーオブジェクト
- スペーサーオブジェクト

フォームオブジェクトの各タイプの詳細については、以降のセクションを参照してください。挿入するフォームオブジェクトを選択します。そのタイプのフォームオブジェクトが、選択されたフォームのエディターウィンドウに挿入されます。選択したフォームオブジェクトの**設定**ウィンドウにそのプロパティが表示されます。このプロパティは、オブジェクトの外観と動作を制御するために編集することができます。複数のフォームオブジェクトを選択した場合、**設定**ウィンドウには共通のプロパティのみが表示されます。

入力フィールド

入力フィールド()は、一行のテキストを入力するためのフォームオブジェクトです。


名前フィールドに入力フィールドオブジェクトの名前を入力します。

デフォルトでは、**編集可能**のチェックボックスが選択されているため、アプリケーションのユーザーは入力フィールドの値を変更できます。**編集可能**チェックボックスをオフにすると、初期値が読み取り専用の値として表示されます。





ツールチップフィールドに、ユーザーが入力フィールドの上にポインタを置いたときにツールチップとして表示されるテキストを入力します。また、入力フィールドオブジェクトには、入力フィールドに収まらない場合に入力されたテキスト、またはエラーや警告を表示するビルトインツールチップが組み込まれています。

また、**設定**ウィンドウには以下のセクションがあります。

ソース




このセクションでは、入力フィールドのデータソースを定義します。このセクションには、**アプリケーションビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類データを表すか、または子ノードを持ちます。入力フィールドでは、COMSOL マルチフィジックスモデルのパラメーターと変数、および**宣言**の下で定義したスカラー変数をデータソースとして使用できます。使用可能なデータノードのリストを拡張するには、**ソース**セクションのヘッダーで**モデルビルダー**へ**スイッチ**し**データアクセスをアクティブ**のボタン()をクリックしてから、アクセスしたいデータのある**モデルビルダー**内のノードを選択します。このボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることが可能な設定箇所の横に**データソース**選択チェックボックス(緑色の四角形)が表示されます。そのチェックボックスをクリックして選択すると、入力フィールドに使用可能なソースノードとしてそのデータが挿入されます。

フィールドが読み取り専用になるように**編集可能**チェックボックスがオフになっている場合には、次のいずれかの情報ノードをソースとして使用することができます。それらはメインの**モデル**ノードと各**スタディ**ノードの下にあります。

- **モデル**>**情報**()の下にある**予想計算時間**ノード()： 予想計算時間は、**root** ノードの**設定**ウィンドウの**予想**フィールドに入力可能な値です。
- **モデル**>**情報**の下にある**前回の計算時間**()： このノードには、最後に計算されたスタディの最後に測定された計算時間が表示されます。
- 各**モデル**>**スタディ**>**情報**の下の**前回の計算時間**ノード()： このノードには、スタディの最後に測定された計算時間が表示されます。

最初にアプリケーションを起動すると、最後に測定された時間がリセットされるため、前回の計算時間は表示されません。

(注) 原文では「**Not available yet** が表示される」となっていますが、弊社では確認できていません。

データを表すノードを選択すると、ツリーの下にあるソースとして使用ツールバーボタン()が有効になります。ノードをクリックするか、Enter キーを押すか、ダブルクリックするか、ノードを右クリックしてソースとして使用を選択して、選択ソースとして追加します。ソースセクションのヘッダーの作成ボタン()をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。変数を作成および使用のダイアログボックスが開き、ソースのデータタイプ(該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下ノード編集ツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。

ノードをソースとして選択すると、そのノードは**選択ソース**の下に選択されたソースとして表示されます。





複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースの値が期待通りにならない可能性があります。また、あるフォームオブジェクトの初期値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。


初期値のリストから**データソース参照**を選択すると、この入力フィールドをソースセクションで定義されたデータソースにリンクし、そのソースで指定された値が使用されます。**データソース参照**は、新規の入力フィールドのデフォルト設定です。値フィールドに入力フィールドの初期値を入力するには、**カスタム値**を選択します。この値には、その文字列に指定した値が上書きされます。

データ検証

このセクションの設定を使用して、ユーザー入力の単位と値に関する検証をさせます。

単位次元チェックのリストから、以下のいずれかのオプションを選択します。

- なし(デフォルト): 単位または値のチェックは行われません。
- **物理量と互換**: **単位次元チェック**リストの下に表示される**物理量**のリストを使用して指定した物理量と入力単位とが互換性があることを確認するチェックが行われます。**量を選択**ボタン()をクリックして**物理量**ダイアログボックスを開き、使用する物理量を検索します。また、ダイアログボックスの上部にあるテキストフィールドに検索文字列を入力し、**フィルター**ボタン()をクリックして物理量のリストをフィルタリングすることもできます。例えば、電位と入力して**フィルター**ボタンをクリックすると、ある種の可能性のある物理量のみがリスト表示されます。
- ユーザーが入力した値または式が互換性がない場合は、値または式がオレンジ色で警告として強調表示され、単位の不一致を説明するツールチップが表示されます。単位の不一致がある場合、アプリケーションは入力された値または式の数値を使用し、それに指定されたデフォルトの単位を追加します(例えば、予想される物理量が長さの場合は 9 [kg] が 9 [m] に変換されます)。
- **単位式と互換**: **単位次元チェック**リストの下に表示される**単位式**フィールドで指定した単位式と入力単位とが互換性があることを確認するチェックが行われます。単位の不一致がある場合は、前のケースと同様の警告表示と処理が行われます。
- **値に単位を追加**: アプリケーションユーザーは、角括弧で囲まれた単位構文を使用して単位を追加することなく、値を入力することができます。アプリケーションは、**単位次元チェック**のリストの下に表示される**単位式**フィールドに単位式を指定します。
- **単位セットから単位を追加**: 入力フィールドは、**宣言**の下に追加された**単位セット**ノードから単位を追加します(**単位セットノード**を参照)。 **単位セット**のリストから使用する単位セットを指定し、**単位リスト**からは使用する単位を指定します。単位リストには、選択された単位セットの全ての定義済みプロパティとその単位がリストされます。**単位なし**のオプションの選択も可能です。

該当する場合は、**単位ラベル追加**のボタン()をクリックして入力フィールドにリンクされた単位ラベルを追加し、入力フィールドの右側に配置します。

なし、**値に単位を追加**、**単位セットから単位を追加**のオプションについては、**フィルター**を使用して数値入力を検証することもできます。**数値検証**の下で、**フィルター**のリストから以下のいずれかのフィルターを選択します。

- **なし**: **単位次元チェック**がなしオプションの場合にのみ表示されます。その場合にはデフォルト設定です。
- **倍精度**(**値に単位を追加**オプションの場合のデフォルト): 入力値が浮動小数点数(倍精度)であることを確認します。対応するテキストフィールドに下限と上限をそれぞれ指定するには、**最小値**と**最大値**のチェックボックスをオンにします。そのテキストフィールドには数値と式を入力でき、**グローバルパラメーターノード**の**設定**ウィンドウで定義されたパラメーターの入力も可能です。パラメータ値が変更されると、制限も動的に変更されます。最小値または最大値の変更後、入力フィールドの値が一時的に無効になる可能性があります。最小値または最大値のパラメータまたは式が**単位式**フィールドに入力された単位と同じ次元を持たない場合、パラメータまたは式は黄色で表示され、ツールチップに単位の不一致が記述されます。パラメータまたは式の次元が同じではあるが、**単位式**フィールドに入力された単位以外の単位である場合には、その単位への単位変換が行われます。
- **整数**: 入力された値が整数であることを確認します。対応するテキストフィールドに下限と上限(整数値として)をそれぞれ指定するには、**最小値**と**最大値**のチェックボックスをオンにします。
- **正規表現**(**なし**オプションの場合にのみ): 正規表現を使用して入力文字列との一致をチェックし、カスタムエラーを発生します。**正規表現**フィールドに照合する式、および**エラーメッセージ**フィールドにエラーメッセージ(デフォルト: 無効な入力)を指定します。

正規表現は、ユーザー入力と一致するパターンで構成されます。パターンには、通常の文字と特殊文字を含めることができます。特殊文字は特殊な意味を持ち、それ自体だけでは何も表しません。特殊文字 `\` は、それを組み合わせた文字が独自の意味をなすように引用する特殊文字として使われます; 例えば、`\\` はバックslashを表します。通常の文字に `\` を使用すると、それは特殊な意味を持ちます。例えば、`\t` はタブ文字を意味し、`\n` は改行文字を意味します。特殊文字 `.` (ドット)は任意の文字を表します。`[abc]` (角括弧内の文字)は、a、b、または c のいずれかの文字を表します。`[^ abc]` は、a、b、または c 以外の文字を表します。`(abc)` (丸括弧内の文字)は、文字列 abc を表します。`*` (アスタリスク)はゼロ回を含む繰り返しを意味します。`+` (プラス)は少なくとも 1 回の繰り返しを意味します。正規表現の詳細については、Java の `java.util.regex` クラスのドキュメントを参照してください。

例:

- `a.c` は文字 a の後に任意の文字が続き、その後に文字 c が続くことを意味します。
- `[^ ab]c` は a と b を除く任意の文字を意味し、その後に c を続けます。
- `a * b +` は、任意の数の文字 a (出現しない場合も含む)の後に少なくとも一つの b が続くことを意味します。

アプリケーションをテストを使用してアプリケーションをテストしてデータ検証のエラーメッセージが表示された場合、エラーメッセージにはフォームオブジェクトの種類とエラーメッセージの出所を特定するためのパスに関する追加情報が含まれています。**アプリケーションを実行**または COMSOL サーバー経由でアプリケーションを実行した際には、追加情報が表示されません。

位置およびサイズ

このセクションには、親フォームのグリッドにある入力フィールドの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、入力フィールドの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチ**セクションの中の**アレイジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

入力フィールドの幅を指定することもできます。幅フィールドに幅(ポイント単位で)を入力します。水平アラインメントのリストから全幅を選択した場合は、最小幅を指定することができます。最小幅を自動的に計算するには自動を選択します(通常は最小サイズ0を意味します)。または、マニュアルを選択して、その下のテキストフィールドに最小幅を指定します。高さフィールドは、入力フィールドの高さがソフトウェアによって決められるため設定変更できません。

さらに、スケッチモードでは、位置 x および位置 y フィールドを使用して、入力フィールドの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、行、列、行スパン、および列スパンの値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。セルマージンのリストから、以下を選択します。

- なし: フォームオブジェクトの周りに余白を使用しません。
- 親フォーム参照(デフォルト): 親フォームに設定されている余白を使用します。
- カスタム: フォームオブジェクトの余白を水平および垂直のテキストフィールドに指定します。

外観

このセクションでは、入力フィールドの背景とテキストの外観を制御できます。

テキスト色のリストから、継承(デフォルト)を選択してフォームノードの設定からテキスト色を継承するか、あるいはリストにある事前定義された色(黒など)を選択します。カスタムを選択して、カラーパレットからカスタムテキストカラーを選択することもできます。

背景色のリストから、入力フィールド内の背景として使用する色として、白(デフォルト)、透明、定義済みの基本色、またはカスタムを選択します(カスタムでは、カラーパレットからカスタム色を選択できます)。

テキストアラインメントのリストでは、入力フィールドのテキストの配置として左、中心、または右から選択します。

テキストのフォントとフォントサイズには、フォームノードのフォント設定がデフォルトとして使用されます。フォントリストから、デフォルトフォントまたはリストにある使用可能なフォントを選択してください。必要に応じて、フォントサイズのコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントのデフォルトサイズが使用されます。

太字のフォントを使用するにはボールドのチェックボックスを選択し、斜体(イタリック体)を使用するにはイタリックのチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときに入力フィールドの初期状態を制御できます。デフォルトでは、入力フィールドは可視表示されて有効になっています。入力フィールドが非表示または使用不可になるように初期状態を設定する場合は、可視または有効化のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

イベント

特定のタイプのフォームオブジェクトでは、データ入力などのイベントが発生したときに実行するメソッドを指定できます。データ変更時のリストには、なし(デフォルト)と使用可能なメソッドが含まれています。イベントのローカルメソッドを追加するには、データ変更時のリストの右側にあるローカルメソッド作成のボタン(+)をクリックします。すると、データ変更時リストで選択したメソッドがローカルメソッドに変更されます。空の onDataChange エディターウィンドウが開かれ、そこでローカルメソッドを定義することができます。入力フィールドオブジェクト上で Ctrl + Alt を押しながらクリックするか、それを右クリックしてローカルメソッドを作成するか、(メソッド編集またはローカルメソッド編集を選択して)コマンドに関連付けられたメソッドを開きま

す。選択したメソッドを開くには、**ソース**へのボタン(📄)をクリックします。ローカルメソッドを削除するには、**ローカルメソッドを削除**のボタン(✕)をクリックします。

データ変更によってトリガされるイベントの場合、新しいデータ値がデータソースに格納された後にイベントが発生されます。



複数のフォームオブジェクトを選択し、それらが全て**データ変更時**のイベントをサポートしている場合は、例えばプロットや出力が無効であることをユーザーに通知するなどの方法を指定することができます。

ボタン

ボタン(OK) フォームオブジェクトはプッシュボタンを表します。ユーザがボタンを押すと、そのプログラムがコマンドのリストを実行します。

ボタンオブジェクトの名前は、**名前**フィールドに入力します。

テキストフィールドには、ボタンに表示されるテキストを入力します。

ボタンに表示するアイコン(画像)を追加するには、ライブラリの**画像**リストにある全ての画像を含む**アイコン**リストから使用する画像ファイルを指定するか、**ライブラリに対する画像を追加してここで使用**のボタン(+)をクリックしてファイルシステム上で使用する画像を検索します。その後、その画像はライブラリの**画像**リストに取り込まれ、ボタンで使用するアイコンとして選択されます。ボタンのサイズがデフォルトの小さい場合、アイコンはボタン上のテキストの代わりに置き換えられます。大きいボタンの場合には、テキストとアイコンは空でない限り表示されます。テキストのみを表示するには、**アイコン**リストから**なし**を選択します。画像をファイルシステムにエクスポートするには(例えば、別のアプリケーションで使用するような場合)、**エクスポート**ボタン(📁)をクリックします。

サイズリストから**小**(デフォルト)を選択するか、**大**を選択して大きな正方形のボタンにします。

ツールチップのフィールドに入力したテキストは、ボタンのツールチップになります。

キーボードショートカットのフィールドの入力によって、ボタンアクションのショートカットを定義することもできます。キーボードショートカットを追加するには、**キーボードショートカット**フィールドをアクティブにしてキーボードショートカットを入力します。

キーボードショートカットでは、普通の文字だけでなく、例えば CTRL + SHIFT + D などの修飾キーを使用する必要があります。ショートカットには、Ctrl キー(CTRL)、Alt キー(ALT)、Shift キー(SHIFT)を使用できます。Ctrl キーは OS X のコマンドとして解釈されることに注意してください。ショートカットでは次のキーを使用しないでください。

- Backspace: ショートカットを消去するために使用できるため
- Delete: ショートカットを消去するために使用できるため
- Escape
- Alt 単独使用: ファイルメニューのショートカットとの競合を避けるため



他のキーボードショートカットをオーバーライドすることが可能ですので、使用するショートカットキーの組み合わせを選択する際には注意してください。

また、**設定**ウィンドウには、次のセクションがあります。

実行コマンド選択

このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーのフィルター処理されたツリーが含まれています。ノードは、コマンドをサポートしているか、または子ノードを持っています。一つまたは複数のコマンドをサポートするノードを選択すると、ツリーの下ツールバーでそれに対応するボタンが有効になります。また、ノードを右クリックして、そのノードの使用可能なコマンドのリストを取得することもできます。選択したノードのコマンドをクリックする(または Enter キーを押すか、ダブルクリックして、**実行**、**プロット**、**値設定**などのデフォルトコマンドでコマンドを追加する)ことによ

て、ツリーの下テーブルの最後の行にコマンドとノードが表示されます。このテーブルには、実行する全てのノードが含まれます。テーブルの下にあるツールバーを使用して、コマンドを削除したり移動したりすることができます。

モデルブランチでは、**パラメーターノード**の下のパラメーターなど、何らかの種類のデータ値を表すノードの全てが、**値設定**のコマンドをサポートしています。**値設定**コマンドをテーブルに追加すると、3番目の列の**引数**が有効になります。この列に、設定する値を入力します。配列を表すデータの場合は、中括弧とコンマを使用して配列要素を入力します。例えば、{1, 2, 3}を入力して、値が1,2,3の3要素配列を設定します。配列と行列を入力する方法の詳細については、[配列 1D 文字列ノード](#)を参照してください。**補間関数ノード**の下にある**ファイル名ノード**のように、ファイルのインポートを表すノードの場合は、**インポートファイル**のコマンドを使用できます。また、全ての**ビューノード**に対して**プロット**コマンドを追加することもでき、引数としてグラフィックスオブジェクトの名前を指定します。**エクスポートノード**の下にある**ノード**の場合は、**実行**コマンドを追加できます。**エクスポート**の下にある**アニメーションノード**に於いてプレーヤーをターゲットとして実行される場合、引数としてグラフィックスオブジェクトの名前を指定します。

ツリーには、**モデルブランチ**に加えて、**アプリケーションツリー**からのいくつかのブランチが含まれています。

- **フォームブランチ:** **フォームノード**は、フォームをアプリケーションのメインフォーム(つまり、アプリケーションウィンドウのコンテンツはこのフォームになります)として設定する**表示**コマンドと、フォームを別のダイアログウィンドウとして表示する**ダイアログ**として**表示**をサポートしています。

- **GUI コマンドブランチ:** このブランチの下にあるコマンドは、三つのサブカテゴリにグループ化されています。

- **ファイルコマンド:** これには、**アプリケーションを保存**(現在の名前前でアプリケーションを保存する); **名前を付けてアプリケーションを保存**(ユーザーがアプリケーションを適切な場所に保存できるようにするファイルブラウザダイアログを開く); **サーバー上でアプリケーションを保存**; **サーバー上でアプリケーションを別名で保存**; **ファイルを開く**(引数列で有効な URI パスを使用して指定されたアプリケーションファイルリソースを開く); **名前を付けてファイルを保存**(同様に、ユーザーが引数列で指定した名前前でファイルを保存できるようにする); および**アプリケーションを終了**(実行中のアプリケーションを閉じる)が含まれています。アプリケーションが COMSOL サーバー上で実行されている場合、**サーバー上でアプリケーションを保存**と**サーバー上でアプリケーションを別名で保存**のコマンドは、COMSOL サーバーアプリケーションライブラリに新しいアプリケーションとして現在の状態を保存します。

- **グラフィックスコマンド:** **画面にわたってズーム**、**現在のビューをリセット**、**シーンライト**、**透明度**、**印刷**、**全選択**、および**選択対象消去**のコマンドがあります。全てのグラフィックコマンドで、コマンドに引数として適用するグラフィックスオブジェクトの名前を追加します。

- **モデルコマンド:** ここでは、**全解除**と**全メッシュ消去**のコマンドが利用できます。

上記のいずれかのノードをダブルクリックまたは右クリックして、**実行**コマンドを追加します。


- **宣言ブランチ:** このブランチには、**アプリケーションビルダー**ウィンドウの**宣言**ブランチの下に追加した変数宣言がタイプ別にグループ化されています。パラメーターと同様に、**値設定**コマンドがサポートされています。

- **メソッドブランチ:** **メソッドノード**は、**実行**コマンドをサポートしています。

- **ライブラリブランチ:** **サウンド**の下で、コマンドシーケンスで再生するサウンドファイルを選択できます。

ツリーの下にあるボタンの一つをクリックすると、現在選択されているコマンドが下のテーブルの**コマンド**列に表示されます。また、**アイコン**列と**引数**列もあり、後者ではコマンドで使用される任意の引数を入力できます。

コマンドのリストの下にある**ローカルメソッド作成**ボタン(+)をクリックすると、テーブル内のコマンドのリスト全体が同等のコードを含む新しいローカルメソッドに変換され、そのローカルの `onClick` メソッドのエディターウィンドウが開かれます。ボタンオブジェクトを `Ctrl + Alt` を押しながらクリックするか、それを右クリックしてローカルメソッドを作成するか、(**メソッド編集**または**ローカルメソッド編集**を選択するか)によって、コマンドに関連付けられたメソッドが開かれます。ローカルメソッドを作成していないときは、**新しいメソッドに変換**ボタン(📄)をクリックして、テーブル内のコマンドのリスト全体を同等のコードを含む新しい**メソッドノード**に変換します。この操作の後、コマンドのリストには、作成された**メソッドノード**の1回の**実行**操作だけが設定された形になります。**コマンド**の下で一つのメソッドを選択するか、リスト内に一つのメソッドしかない場合、**メソッド**へのボタン(📄)をクリックしてそのメソッドのエディターウィンドウに移動することができます。**引数編集**のボタン(🔍)の詳細については、[宣言とコマンドシーケンスでの初期値と引数の編集](#)を参照してください。**上へ移動**(↑)、**下へ移動**(↓)、**削除**

() のツールバーボタンを使用して、リストからコマンドを整理したり削除することができます。(また、ローカルメソッドを削除することもできます。)

ダイアログアクション

ダイアログウィンドウでボタンを使用すると、そのボタンによってダイアログを制御できます。ユーザーがボタンをクリックしたときにダイアログウィンドウを閉じるには、**ダイアログを閉じる**のチェックボックスをオンにします。ダイアログウィンドウ内の全てのフォームとフォームオブジェクトは、それらの値を一時的に保存します。これは、ユーザーがダイアログウィンドウをキャンセルして、フォームとフォームオブジェクトをバックアップしている実際のデータにデータを格納せずに閉じるかもしれないからです。ユーザーがボタンをクリックしたときにデータを保存するには、**変更を保存**のチェックボックスをオンにします。

位置およびサイズ

このセクションには、親フォームのグリッドにあるボタンの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、入力フィールドの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

ボタンの幅を指定する場合は、まず**幅**リストから**マニュアル**を選択します。その後、関連するフィールドに幅(ポイント単位)を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、代わりに**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ 0 を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。

ボタンの高さを指定する場合は、まず**高さ**リストから**マニュアル**を選択します。その後、関連するフィールドに高さ(ポイント単位)を入力します。**垂直アライメント**のリストから**全幅**を選択した場合は、代わりに**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します(通常は最小サイズ 0 を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および **位置 y** フィールドを使用して、入力フィールドの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、ボタン上のテキストの外観を制御できます。


テキスト色のリストから、テキストに使用する色を選択します。**継承**(デフォルト; フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

ボタンラベルのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォントリスト**から、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体(イタリック体)を使用するには**イタリック**のチェックボックスを選択します。



状態では、ユーザーがアプリケーションを実行するときにボタンの初期状態を制御できます。デフォルトでは、ボタンは可視表示されて有効になっています。ボタンが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

トグルボタン

トグルボタン() フォームオブジェクトは、トグルボタンを表します。トグルボタンは、ユーザーがボタンを選択したときに実行される一つのアクション(コマンドシーケンス)と、ユーザーがボタンを選択解除したときに実行される別のアクションの二つのボタンとして機能します。

トグルボタンオブジェクトの名前は、**名前**フィールドに入力します。

テキストフィールドには、ボタンに表示されるテキストを入力します。

ボタンに表示するアイコン(画像)を追加するには、ライブラリの**画像**リストにある全ての画像を含む**アイコン**リストから使用する画像ファイルを指定するか、**ライブラリに対する画像を追加してここで使用**のボタン()をクリックしてファイルシステム上で使用する画像を検索します。その後、その画像はライブラリの**画像**リストに取り込まれ、ボタンで使用するアイコンとして選択されます。ボタンのサイズがデフォルトの小さい場合、アイコンはボタン上のテキストの代わりに置き換えられます。大きいボタンの場合には、テキストとアイコンは空でない限り表示されます。テキストのみを表示するには、**アイコン**リストから**なし**を選択します。画像をファイルシステムにエクスポートするには(例えば、別のアプリケーションで使用するような場合)、**エクスポート**ボタン()をクリックします。

サイズリストから**小**(デフォルト)を選択するか、**大**を選択して大きな正方形のボタンにします。

ツールチップのフィールドに入力したテキストは、ボタンのツールチップになります。

キーボードショートカットのフィールドの入力によって、ボタンアクションのショートカットを定義することもできます。キーボードショートカットを追加するには、**キーボードショートカット**フィールドをアクティブにしてキーボードショートカットを入力します。

キーボードショートカットでは、普通の文字だけでなく、例えば CTRL + SHIFT + D などの修飾キーを使用する必要があります。ショートカットには、Ctrl キー(CTRL)、Alt キー(ALT)、Shift キー(SHIFT)を使用できます。Ctrl キーは OS X のコマンドとして解釈されることに注意してください。ショートカットでは次のキーを使用しないでください。


- Backspace: ショートカットを消去するために使用できるため
- Delete: ショートカットを消去するために使用できるため
- Escape
- Alt 単独使用: ファイルメニューのショートカットとの競合を避けるため






他のキーボードショートカットをオーバーライドすることが可能ですので、使用するショートカットキーの組み合わせを選択する際には注意してください。

また、設定ウィンドウには、次のセクションがあります。

ソース

このセクションでは、トグルボタンのデータソースを定義します。このセクションには、**アプリケーションビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類のデータを表すか、または子ノードを持ちます。トグルボタンでは、COMSOL マルチフィジックスモデルの変数、および宣言の下で定義したスカラー文字列またはブーリアン変数をデータソースとして使用できます。使用可能なデータノードのリストを拡張するには、**ソースセクションのヘッダーでモデルビルダーへスイッチしデータアクセスをアクティブ**のボタン()をクリックして、モデルビルダーに切り替わってから、アクセスしたいデータのある**モデルビルダー**ブランチ内のノードを選択します。このボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることが可能な設定箇所の横に**データソース選択**チェックボックス(緑色の四角形)が表示されます。そのチェックボックスをクリックして選択すると、トグルボタンに使用可能なソースノードとしてそのデータが挿入されます。

データを表すノードを選択すると、ツリーの下にある**ソースとして使用**ツールバーボタン()が有効になります。ノードをクリックするか、Enter キーを押すか、ダブルクリックするか、ノードを右クリックして**ソースとして使用**を選択して、選択ソースとして追加します。また、**ソースセクションのヘッダーの作成**ボタン()をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。**変数を作成および使用**のダイアログボックスが開き、ソースのデータタイプ(該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下**ノード編集**ツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。

ノードをソースとして選択すると、そのノードは**選択ソース**の下に選択されたソースとして表示されます。



複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースのデフォルト値は期待通りとは限りません。あるフォームオブジェクトのデフォルト値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。

初期値のリストから**データソース参照**を選択すると、このトグルボタンを**ソースセクション**で定義された変数にリンクし、そのデータソースで指定された値が使用されます。**カスタム値**を選択すると、**初期状態**リストから**選択**(デフォルト)または**クリア**のいずれかの初期状態を指定することができます。選択されたトグルボタンの値は on で、クリアされたトグルボタンの値は off です。



グローバルパラメータがトグルボタンのデータソースである場合は、二つのパラメータ on と off をそれぞれ値 1 と 0 として追加する必要があります。そうでない場合、COMSOL マルチフィジックスのソフトウェアがパラメータ on と off を検出するとエラーが発生します。

実行コマンド選択

このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーのフィルター処理されたツリーが含まれています。ノードは、コマンドをサポートしているか、または子ノードを持っています。一つまたは複数のコマンドをサポートするノードを選択すると、ツリーの下**ツールバー**でそれに対応するボタンが有効になります。また、ノードを右クリックして、そのノードの使用可能なコマンドのリストを取得することもできます。選択したノードのコマンドをクリックする(または Enter キーを押すか、ダブルクリックして、**実行**、**プロット**、**値設定**などのデフォルトコマンドでコマンドを追加する)ことによって、ツリーの下**テーブル**の最後の行にコマンドとノードが表示されます。このテーブルには、実行する全てのノードが含まれます。テーブルの下にある**ツールバー**を使用して、コマンドを削除したり移動したりすることができます。

モデルブランチでは、**パラメーター**ノードの下**パラメーター**など、何らかの種類のデータ値を表すノードの全てが、**値設定**のコマンドをサポートしています。**値設定**コマンドをテーブルに追加すると、3 番目の列の**引数**が有効になります。この列に、設定する値を入力します。配列を表すデータの場合は、中括弧とコンマを使用して配列要素を入力します。例えば、{1, 2, 3} を入力して、値が 1,2,3 の 3 要素配列を設定します。配列と行列を入力する方法の詳細については、**配列 1D 文字列ノード**を参照してください。**補間関数**ノードの下にある**ファイル名**ノードなどのように、ファイルのインポートを表すノードの場合は、イ

ンポートファイルのコマンドを使用できます。また、全てのビューノードに対して**プロット**コマンドを追加することもでき、引数としてグラフィックスオブジェクトの名前を指定します。**エクスポート**ノードの下にあるノードの場合は、**実行**コマンドを追加できます。**エクスポート**の下にある**アニメーション**ノードに於いてプレーヤーをターゲットとして実行される場合、引数としてグラフィックスオブジェクトの名前を指定します。

ツリーには、**モデル**ブランチに加えて、アプリケーションツリーからのいくつかのブランチが含まれています。

- **フォーム**ブランチ: **フォーム**ノードは、フォームをアプリケーションのメインフォーム(つまり、アプリケーションウィンドウのコンテンツはこのフォームになります)として設定する**表示**コマンドと、フォームを別のダイアログウィンドウとして表示する**ダイアログ**として**表示**をサポートしています。
- **GUI コマンド**ブランチ: このブランチの下にあるコマンドは、三つのサブカテゴリにグループ化されています。

- **ファイル**コマンド: これには、**アプリケーションを保存**(現在の名前前でアプリケーションを保存する); **名前を付けてアプリケーションを保存**(ユーザーがアプリケーションを適切な場所に保存できるようにするファイルブラウザーダイアログを開く); **サーバー上でアプリケーションを保存**; **サーバー上でアプリケーションを別名で保存**; **ファイルを開く**(引数列で有効な URI パスを使用して指定されたアプリケーションファイルリソースを開く); **名前を付けてファイルを保存**(同様に、ユーザーが引数列で指定した名前前でファイルを保存できるようにする); および**アプリケーションを終了**(実行中のアプリケーションを閉じる)が含まれています。アプリケーションが COMSOL サーバー上で実行されている場合、**サーバー上でアプリケーションを保存**と**サーバー上でアプリケーションを別名で保存**のコマンドは、COMSOL サーバーアプリケーションライブラリに新しいアプリケーションとして現在の状態を保存します。
- **グラフィックス**コマンド: **画面にわたってズーム**、**現在のビューをリセット**、**シーンライト**、**透明度**、**印刷**、**全選択**、および**選択対象消去**のコマンドがあります。全てのグラフィックコマンドで、コマンドに引数として適用するグラフィックスオブジェクトの名前を追加します。
- **モデル**コマンド: ここでは、**全解除**と**全メッシュ消去**のコマンドが利用できます。

上記のいずれかのノードをダブルクリックまたは右クリックして、**実行**コマンドを追加します。

- **宣言**ブランチ: このブランチには、**アプリケーションビルダー**ウィンドウの**宣言**ブランチの下に追加した変数宣言がタイプ別にグループ化されています。パラメーターと同様に、**値設定**コマンドがサポートされています。
- **メソッド**ブランチ: **メソッド**ノードは、**実行**コマンドをサポートしています。
- **ライブラリ**ブランチ: **サウンド**の下で、コマンドシーケンスで再生するサウンドファイルを選択できます。

ツリーの下にあるボタンの一つをクリックすると、現在選択されているコマンドが下のテーブルの**コマンド**列に表示されます。また、**アイコン**列と**引数**列もあり、後者ではコマンドで使用される任意の引数を入力できます。

コマンドのリストの下にある**ローカルメソッド作成**ボタン(+)をクリックすると、テーブル内のコマンドのリスト全体が同等のコードを含む新しいローカルメソッドに変換され、そのローカルの `onClick` メソッドのエディターウィンドウが開かれます。ボタンオブジェクトを `Ctrl + Alt` を押しながらクリックするか、それを右クリックしてローカルメソッドを作成することもできます。ローカルメソッドを作成していないときは、**新しいメソッドに変換**ボタン(📄)をクリックして、テーブル内のコマンドのリスト全体を同等のコードを含む新しい**メソッド**ノードに変換します。この操作の後、コマンドのリストには、作成された**メソッド**ノードの1回の**実行**操作だけが設定された形になります。**コマンド**の下で一つのメソッドを選択するか、リスト内に一つのメソッドしかない場合、**メソッド**へのボタン(📄)をクリックしてそのメソッドのエディターウィンドウに移動することができます。**引数編集**のボタン(🔍)の詳細については、**宣言とコマンドシーケンスでの初期値と引数の編集**を参照してください。**上へ移動**(↑)、**下へ移動**(↓)、**削除**(✖)のツールバーボタンを使用して、リストからコマンドを整理したり削除することができます。(また、ローカルメソッドを削除することもできます。)

位置およびサイズ

このセクションには、親フォームのグリッドにあるトグルボタンの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、トグルボタンの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

- **垂直アラインメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アラインメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチセクション**の中の**アレンジメニュー**でアラインメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アラインメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

トグルボタンの幅を指定したい場合には、**幅**のリストから**マニュアル**を選択し、関連するフィールドに幅(ポイント単位で)を入力します。**水平アラインメント**のリストから**全幅**を選択した場合は、代わりに**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ 0 を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。

トグルボタンの高さを指定する場合は、まず**高さ**リストから**マニュアル**を選択します。その後、関連するフィールドに高さ(ポイント単位)を入力します。**垂直アラインメント**のリストから**全幅**を選択した場合は、代わりに**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します(通常は最小サイズ 0 を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、トグルボタンの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、トグルボタン上のテキストの外観を制御できます。

テキスト色のリストから、トグルボタン上のテキストに使用する色を選択します。**継承**(デフォルト; フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

トグルボタンラベルのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォント**リストから、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体(イタリック体)を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにトグルボタンの初期状態を制御できます。デフォルトでは、トグルボタンは可視表示されて有効になっています。トグルボタンが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォ

ームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

チェックボックス


チェックボックス () フォームオブジェクトはチェックボックスを表します。チェックボックスでは、選択またはクリアができません。選択されたチェックボックスはデータソース値を *on* に設定し、チェックボックスをオフにすると *off* に設定されます。




チェックボックスオブジェクトの名前は、**名前**フィールドに入力します。

テキストフィールドには、チェックボックスの横に表示されるラベルを入力します。

また、**設定**ウィンドウには、次のセクションがあります。

ソース

このセクションでは、チェックボックスのデータソースを定義します。このセクションには、**アプリケーションビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類のデータを表すか、または子ノードを持ちます。チェックボックスでは、COMSOL マルチフィジックスモデルの変数、および**宣言**の下で定義したスカラー文字列またはブーリアン変数をデータソースとして使用できます。使用可能なデータノードのリストを拡張するには、**ソース**セクションのヘッダーで**モデルビルダーへスイッチしデータアクセスをアクティブ**のボタン () をクリックして、モデルビルダーに切り替わってから、アクセスしたいデータのある**モデルビルダー**ブランチ内のノードを選択します。このボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることが可能な設定箇所の横に**データソース**選択チェックボックス (緑色の四角形) が表示されます。そのチェックボックスをクリックして選択すると、チェックボックスに使用可能なソースノードとしてそのデータが挿入されます。

データを表すノードを選択すると、ツリーの下にある**ソースとして使用**ツールバーボタン () が有効になります。それをクリックするか、Enter キーを押すか、ダブルクリックするか、ノードを右クリックして**ソースとして使用**を選択して、選択ソースとして追加します。また、**ソース**セクションのヘッダーの**作成**ボタン () をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。**変数を作成および使用**のダイアログボックスが開き、ソースのデータタイプ (該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下**ノード編集**ツールバーボタン () をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。

ノードをソースとして選択すると、そのノードは**選択**ソースの下に選択されたソースとして表示されます。



複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースのデフォルト値は期待通りとは限りません。あるフォームオブジェクトのデフォルト値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。

初期値のリストから**データソース参照**を選択すると、このチェックボックスを**ソース**セクションで定義された変数にリンクし、そのデータソースで指定された値が使用されます。**カスタム**値を選択すると、**初期状態**リストから**選択** (デフォルト) または**クリア**のいずれかの初期状態を指定することができます。選択されたチェックボックスの値は *on* で、クリアされたチェックボックスの値は *off* です。



グローバルパラメータがチェックボックスのデータソースである場合は、二つのパラメータ *on* と *off* をそれぞれ値 1 と 0 として追加する必要があります。そうでない場合、COMSOL マルチフィジックスのソフトウェアがパラメータ *on* と *off* を検出するとエラーが発生します。

位置およびサイズ

このセクションには、親フォームのグリッドにあるチェックボックスの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、チェックボックスの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向（該当する場合）の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向（該当する場合）の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチセクション**の中の**アレンジメニュー**でアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。**幅フィールド**と**高さフィールド**は、チェックボックスの寸法がソフトウェアによって決められるため設定変更できません。

さらに、スケッチモードでは、**位置 x** および **位置 y** フィールドを使用して、チェックボックスの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン（グリッドモードの場合のみ）では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**：フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**（デフォルト）：親フォームに設定されている余白を使用します。
- **カスタム**：フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、チェックボックスラベルのテキストと背景の外観を制御できます。

テキスト色のリストから、チェックボックスラベルのテキストに使用する色を選択します。**継承**（デフォルト；フォームオブジェクトが配置されているフォームの設定を使用）、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

背景色のリストから、チェックボックス内の背景として使用する色を選択します。**透明**（デフォルト）、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

チェックボックスラベルのテキストのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォントリスト**から、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ（ポイント単位）を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。


太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体（イタリック体）を使用するには**イタリック**のチェックボックスを選択します。あるいは、そのテキストに下線を付けるには**アンダーライン**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにチェックボックスの初期状態を制御できます。デフォルトでは、チェックボックスは可視表示されて有効になっています。チェックボックスが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。


イベント



ユーザーがチェックボックスの値を変更したときにアプリケーションが実行するコードメソッドを追加することができます。新しいデータ値がデータソースに格納されると、その後にイベントが起動されます。**データ変更時**のリストには、なし(デフォルト)と使用可能なメソッドが含まれています。イベントのローカルメソッドを追加するには、**データ変更時**のリストの右側にある**ローカルメソッド作成**のボタン(+)をクリックします。すると、**データ変更時**リストで選択したメソッドが**ローカルメソッド**に変更されます。空の **onDataChange** エディターウィンドウが開かれ、そこでローカルメソッドを定義することができます。チェックボックス上で Ctrl + Alt を押しながらクリックするか、それを右クリックしてローカルメソッドを作成するか、(**メソッド編集**または**ローカルメソッド編集**を選択して)コマンドに関連付けられたメソッドを開きます。選択したメソッドを開くには、**ソース**へのボタン(🔍)をクリックします。ローカルメソッドを削除するには、**ローカルメソッドを削除**のボタン(✖)をクリックします。

コンボボックス

コンボボックス() フォームオブジェクトは、ドロップダウンリストボックスと編集可能なテキストフィールドの組み合わせ、または編集不可能なドロップダウンリストボックスとして機能するコンボボックスを表します。**設定**ウィンドウには、次のセクションがあります。

ソース

このセクションでは、コンボボックスのデータソースを定義します。このセクションには、**アプリケーションビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類 of データを表すか、または子ノードを持ちます。コンボボックスでは、モデルの変数、および**宣言**の下で定義したものをデータソースとして使用でき、そこには**単位セット**ノードも含まれます。使用可能なデータノードのリストを拡張するには、**ソース**セクションのヘッダーで**モデルビルダーヘスイッチ** **データアクセスをアクティブ**のボタン()をクリックして、モデルビルダーに切り替わってから、アクセスしたいデータの**あるモデルビルダー**ブランチ内のノードを選択します。このボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることが可能な設定箇所の横に**データソース**選択チェックボックス(緑色の四角形)が表示されます。そのチェックボックスをクリックして選択すると、コンボボックスに使用可能なソースノードとしてそのデータが挿入されます。

データを表すノードを選択すると、ツリーの下にある**ソースとして使用**ツールバーボタン()が有効になります。それをクリックするか、代わりに Enter キーを押すか、ダブルクリックするか、ノードを右クリックして**ソースとして使用**を選択して、選択ソースとして追加します。また、**ソース**セクションのヘッダーの**作成**ボタン(+)をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。**変数を作成および使用**のダイアログボックスが開き、ソースのデータタイプ(該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下**のノード編集**ツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。





ノードをソースとして選択すると、そのノードは**選択ソース**の下に選択されたソースとして表示されます。



複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースのデフォルト値は期待通りとは限りません。あるフォームオブジェクトのデフォルト値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。

初期値のリストから、コンボボックスの初期値を定義する方法を選択します。オプションには、**最初の許可される値**、**データソース参照**(デフォルト。選択したデータソースで指定された値を使用。)、**カスタム値**があります。**カスタム値**のオプションの場合、**値**のリストには現在フォームオブジェクトに表示されている許可される値が表示され、これは選択可能な選択リストとその値に依存します。データソースが、許可される値のリストを持つ埋め込みモデルの設定である場合は、これらの値も**値**リストに含まれます。例えば、選択された初期値が選択リストから削除されたために無効になった場合、それは、**無効なデフォルト値**の後にその値が続けて記載されるテキストによって示される初期値として保持されます。

選択リスト

選択リストのセクションでは、コンボボックスに許可された値を与える**選択リストノード**を追加します。選択したデータソースが許可された値のセットを持つリストである場合、それらの値のサブセットのみがコンボボックスの許可された値の中に表示されます。選択された選択リスト内の他の値は全て無視されます。利用可能な**選択リストノード**は、**選択元**というリスト領域に表示されます。**選択対象追加**  のボタンをクリックすると、選択した**選択リストノード**が**選択**というリスト領域に追加挿入されます。**選択対象を削除**  のボタンをクリックすると、選択済みの**選択リストノード**が**選択**というリスト領域から削除されます。また、領域内の**選択リストノード**をダブルクリックして、それを**選択元**の領域から**選択**の領域に、あるいはその逆に移動することができます。**新規選択リスト追加**のボタン()をクリックして**選択リスト**ウィンドウを開き、新しい選択リストを定義することができます。そのウィンドウのテーブルで、**値**の列に許可されている値、**表示名**の列に対応する名前を入力します。**OK** をクリックすると、**アプリケーションビルダー**ツリーの**宣言ノード**の下の**選択リストノード**()に新しい選択リストが追加され、同時に**選択**の領域にも追加されます。

ソースセクションで、許可された値のリストをデータソースとして持つプロパティを選択すると、そのプロパティは最初に**選択**のリスト領域に置かれた**ノード**になります。これを**選択元**のリスト領域に移動すると、許可された値のリストがクリアされます。それを再度元に戻すか、プロパティの値リストに属する値を持ったカスタム選択リストを追加することも可能です。プロパティリストと**選択リストノード**が両方とも**選択**のリスト領域にある場合、それらはマージされます。その際に同一の値があった場合には、**選択**のリスト領域内で最初にくる項目の記述が選ばれます。この方法を利用して、プロパティリスト内の項目の一つを名前変更することができます。許可された値のリストもある埋め込みモデル内の別のプロパティにソースを切り替えた場合には、以前のプロパティリストノードが**選択元**の領域リストと**選択**の領域リストの両方から削除され、新規ノードが**選択**の領域リストに追加されます。

その他の値を許可のチェックボックスを選択すると、任意の値を入力することができるコンボボックスにすることができます。そのようなコンボボックスは任意の値を受け入れることができ、選択リストによって定義された値に限定されません。

位置およびサイズ

このセクションには、親フォームのグリッドにあるコンボボックスの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、コンボボックスの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォーム**ツールバーの**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

コンボボックスの幅を指定することもできます。**幅**フィールドに幅(ポイント単位で)を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ 0 を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。**高さ**フィールドは、コンボボックスの高さがソフトウェアによって決められるため設定変更できません。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、コンボボックスの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、コンボボックスの外観を制御できます。

テキスト色のリストから、コンボボックス上のテキストに使用する色を選択します。**継承**(デフォルト; フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

コンボボックス上のテキストのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォントリスト**から、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。


太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体(イタリック体)を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにコンボボックスの初期状態を制御できます。デフォルトでは、コンボボックスは**可視表示**されて有効になっています。コンボボックスが**非表示**または**使用不可**になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

イベント

コンボボックスのデータが変更されたときにアプリケーションが実行するコードメソッドを追加することができます。新しいデータ値がデータソースに格納されると、その後イベントが起動されます。**データ変更時**のリストには、**なし**(デフォルト)と使用可能なメソッドが含まれています。イベントのローカルメソッドを追加するには、**データ変更時**のリストの右側にある**ローカルメソッド作成**のボタン(+)をクリックします。すると、**データ変更時**リストで選択したメソッドが**ローカルメソッド**に変更されます。空の **onDataChange** エディターウィンドウが開かれ、そこでローカルメソッドを定義することができます。コンボボックス上で **Ctrl + Alt** を押しながらクリックするか、それを右クリックしてローカルメソッドを作成するか、(**メソッド編集**または**ローカルメソッド編集**を選択して)コマンドに関連付けられたメソッドを開きます。選択したメソッドを開くには、**ソース**へのボタン(🔍)をクリックします。ローカルメソッドを削除するには、**ローカルメソッドを削除**のボタン(✖)をクリックします。

テキストラベル

テキストラベル() フォームオブジェクトは、フォームウィンドウに情報を表示するために使用できる静的なテキストを表します。

テキストラベルの名前は、**名前**フィールドに入力します。

複数行のテキストを表示したい場合には、**複数行テキスト**のチェックボックスをオンにします。デフォルトでは、ラベルは1行のテキストを表示します。

テキストフィールドには、表示するテキストを入力します。

また、**設定**ウィンドウには、次のセクションがあります。

位置およびサイズ

このセクションには、親フォームのグリッドにあるテキストラベルの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、テキストラベルの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向（該当する場合）の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向（該当する場合）の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

テキストの寸法はソフトウェアによって決められるため、**幅フィールド**と**高さフィールド**に設定することはできません。

さらに、スケッチモードでは、**位置 x** および **位置 y** フィールドを使用して、テキストラベルの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン（グリッドモードの場合のみ）では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**：フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**（デフォルト）：親フォームに設定されている余白を使用します。
- **カスタム**：フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、テキストと背景の外観を制御できます。

テキスト色のリストから、テキストに使用する色を選択します。**継承**（デフォルト；フォームオブジェクトが配置されているフォームの設定を使用）、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

背景色のリストから、テキストラベルの背景として使用する色を選択します。**透明**（デフォルト）、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。


テキストのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォント**リストから、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ（ポイント単位）を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体（イタリック体）を使用するには**イタリック**のチェックボックスを選択します。あるいは、そのテキストに下線を付けるには**アンダーライン**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにテキストラベルの初期状態を制御できます。デフォルトでは、テキストラベルは可視表示されて有効になっています。テキストラベルが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。

フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

単位

単位 () フォームオブジェクトは、単位を表示するためのあらかじめ定義されたラベルです。**入力フィールド**からこのノードにリンクするとき、そのフィールドの単位チェックとしても機能します([入力フィールド](#)を参照)。

単位オブジェクトの名前は、**名前**フィールドに入力します。

ラベルリストから、単位をどのように表示するかのオプションを選択します。**レファレンス参照**では、入力フィールドフォームオブジェクトなどの参照元に定義されている単位を表示します。**ラベルのソース**セクションのツリーからソースを選択します。または、**固定**を選択し、表示される**単位式**のフィールドに単位式を入力します。

デフォルトでの単位表示は Unicode レンダリングを使用します。単位の表示が期待通りにならない場合は、**LaTeX 記法**のチェックボックスを選択して、LaTeX レンダリングを使用することを検討してください。その場合、単位表示は選択されたフォントに依存しません。また、環境設定の問題があるかもしれません。このチェックボックスが選択されているかどうかによって、単位表示の見た目と配置がわずかに変わります。

また、**設定**ウィンドウには、次のセクションがあります。

位置およびサイズ

このセクションには、親フォームのグリッドにある単位の全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、単位の水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するよう場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォーム**ツールバーの**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。単位ラベルの寸法はソフトウェアによって決められるため、**幅**フィールドと**高さ**フィールドに設定することはできません。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、単位ラベルの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン (グリッドモードの場合のみ) では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照** (デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、単位ラベルのテキストと背景の外観を制御できます。

テキスト色のリストから、単位ラベルのテキストに使用する色を選択します。**継承**(デフォルト; フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

背景色のリストから、単位表示内の背景として使用する色を選択します。**透明**(デフォルト)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

単位ラベルのテキストのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォント**リストから、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体(イタリック体)を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときに単位の初期状態を制御できます。デフォルトでは、単位は可視表示されて有効になっています。単位が非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。


方程式

方程式(Δu) フォームオブジェクトは、方程式などの LaTeX レンダリングによるシンボルを表示できます。



方程式オブジェクトの名前は、**名前**フィールドに入力します。

設定ウィンドウには、次のセクションがあります。

方程式

LaTeX シンタックスで**方程式**を入力するフィールドに、表示する方程式または数式を入力します。プログラムは、テキストフィールドを離れた後、レンダリングされた LaTeX シンタックスのプレビューを表示します。それ以外の場合は、**方程式プレビュー**を**リフレッシュ**のボタン()をクリックして手動でプレビューします。デフォルトの方程式は $-\nabla \cdot (k \nabla u)$ で、次のように表示されます。

$$-\nabla \cdot (k \nabla u)$$

方程式セクションのヘッダーでは、**数式を追加**()および**数式置換**()ボタンを使用して、ベクトル演算、ギリシャ文字、ビッグオペレーター、セット、ロジック、テキストフォーマットといった事前定義された LaTeX 式を選択します。これらは、方程式の中に挿入したり、新しい方程式の最初の段階で利用することができます。

位置およびサイズ

このセクションには、親フォームのグリッドにある方程式の全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、方程式の水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アラインメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバーのスケッチセクション**の中の**アレンジメニュー**でアラインメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アラインメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。方程式の寸法はソフトウェアによって決められるため、**幅フィールド**と**高さフィールド**に設定することはできません。

さらに、スケッチモードでは、**位置 x** および **位置 y** フィールドを使用して、方程式の絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および **列スパン** の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン (グリッドモードの場合のみ) では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン** のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照** (デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、方程式の外観を制御できます。

テキスト色 のリストからフォームオブジェクトが配置されているフォームの設定を使用するには、**継承** (デフォルト) を選択します。また、定義済みの基本色のどれかを選択することができ、**カスタム** を選択してカラーパレットからカスタムカラーを選択することもできます。

方程式のテキストのデフォルトのフォントサイズは、**フォームノード** のフォント設定が使用される **デフォルトサイズ** です。必要に応じて、**フォントサイズ** のコンボボックスでフォントサイズ (ポイント単位) を選択するか入力します。

状態 では、ユーザーがアプリケーションを実行するときに方程式の初期状態を制御できます。デフォルトでは、方程式は可視表示されて有効になっています。方程式が非表示または使用不可になるように初期状態を設定する場合は、**可視** または **有効化** のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

ライン

ライン (—) フォームオブジェクトは、水平または垂直の線をフォームに配置して、フォームオブジェクトのグループを区切りません。

ラインオブジェクトの名前は、**名前フィールド** に入力します。

方向 リストから **水平** (デフォルト) または **垂直** を選択します。**水平** デバイダとして、水平ライン内に表示されるテキストを追加することもできます。そのようなテキストを追加するには、**デバイダテキストを含める** のチェックボックスを選択し、関連する **テキストフィールド** にテキストを入力します。

また、**設定** ウィンドウには、次のセクションがあります。

位置およびサイズ

このセクションには、親フォームのグリッドにあるラインの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、ラインの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

水平アライメントのリストで**全幅**を選択していない場合には、水平ラインの幅を指定することもできます。**幅**フィールドに幅(ポイント単位で)を入力します。水平ラインの高さはソフトウェアによって決められるため、**高さ**フィールドは設定できません。垂直ラインの場合には、代わりに高さを指定することができます。但し、**垂直アライメント**のリストで**全幅**を選択していない場合に限りです。**高さ**フィールドに高さ(ポイント単位で)を入力します。垂直ラインの幅はソフトウェアによって決められるため、**幅**フィールドは設定できません。

水平アライメントのリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ 0 を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。**垂直アライメント**のリストから**全幅**を選択した場合は、**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します(通常は最小サイズ 0 を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および **位置 y** フィールドを使用して、ラインの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、ラインのデバイダテキストの外観を制御できます。


テキスト色のリストから、デバイダテキストに使用する色を選択します。**継承**(デフォルト: フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

デバイダテキストのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォント**リストから、**デフォルト**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体(イタリック体)を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにラインの初期状態を制御できます。デフォルトでは、ラインは可視表示されて有効になっています。ラインが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

データ表示

データ表示 () は、数値データとしてのいくつかのプロパティを表示するフォームオブジェクトです。


データ表示オブジェクトの名前は、**名前**フィールドに入力します。

デフォルトでのデータ表示は Unicode レンダリングを使用します。データ表示が期待通りにならない場合、または表示する式に LaTeX シンタックスが含まれている場合は、**LaTeX 記法**のチェックボックスを選択して、LaTeX レンダリングを使用することを検討してください。その場合、データ表示は選択されたフォントに依存しません。また、環境設定の問題があるかもしれません。このチェックボックスが選択されているかどうかによって、データ表示の見た目と配置がわずかに変わります。

ツールチップのフィールドには、ユーザーがデータ表示上にポインタを置いたときにツールチップとして表示されるテキストを入力します。







また、**設定**ウィンドウには、次のセクションがあります。

ソース

このセクションでは、表示するデータのソースを定義します。このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類 of データを表すか、または子ノードを持ちます。データ表示では、COMSOL マルチフィジックスモデルの変数、スカラー文字列、材料プロパティなどのモデルデータの配列と行列設定、**宣言**の下で定義する数値 (倍精度および整数) 変数と配列または行列をデータソースとすることができます。データ表示は、LaTeX シンタックスで配列または行列を表示することができます。使用可能なデータノードのリストを拡張するには、**ソース**セクションのヘッダーで**モデルビルダーへスイッチしデータアクセスをアクティブ**のボタン () をクリックし、モデルビルダーに切り替わってから、アクセスしたいデータのある**モデルビルダー**内のノードを選択します。このボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることが可能な設定箇所の横に**データソース** **選択**チェックボックス (緑色の四角形) が表示されます。



そのデータをデータ表示オブジェクトの使用可能なソースノードとして含めるには、チェックボックスをクリックして選択します。


メインの**モデル**ノードと各**スタディ**ノードの下にある以下の**情報**ノードのいずれかを、ソースとして使用するために選択することもできます。

- **モデル**>**情報** () の下の**予測計算時間**ノード () : 予測計算時間は、アプリケーション開発者が **root** ノードの**設定**ウィンドウの**予想**フィールドに入力できる値です。
- **モデル**>**情報** () の下の**前回の計算時間**ノード () : このノードには、最後に計算されたスタディの最後に測定された計算時間が表示されます。
- 各**モデル**>**スタディ**>**情報** () の下の**前回の計算時間**ノード () : このノードには、スタディの最後に測定された計算時間が表示されます。

最初にアプリケーションを起動すると、最後に測定された時間がリセットされるため、前回の計算時間は表示されません。

(注) 原文では「**Not available yet** が表示される」となっていますが、弊社では確認できていません。

データを表すノードを選択すると、ツリーの下にある**ソースとして使用**ツールバーボタン () が有効になります。それをクリックするか、代わりに Enter キーを押すか、ダブルクリックするか、ノードを右クリックして**ソースとして使用**を選択して、選択ソースとして追加します。また、**ソース**セクションのヘッダーの**作成**ボタン () をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。**変数を作成および使用**のダイアログボックスが開き、ソースのデータタイプ (該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下**のノード**編

集ツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。

ノードをソースとして選択すると、そのノードは**選択ソース**の下に選択されたソースとして表示されます。

数字フォーマット

桁数フィールドには、出力に表示する有効桁数を入力します(デフォルト: 4)。表記法のリストからは、データに使用する表記法を選択します。このオプションには、**自動**(デフォルト)、**サイエンティフィック**、および**小数**があります。**自動**を使用すると、出力が約 3 桁以上または 1 より小さい場合、プログラムによってサイエンティフィック表記に切り替えられます。**自動**および**サイエンティフィック**のオプションの場合、**指数**リストからサイエンティフィック表記法のための指数形式を選択することもできます。この指数の選択肢は、**10 のべき乗**(デフォルト)と**e 形式**です。**全桁表示**のチェックボックス(サイエンティフィックと小数のオプションで使用可能となる)を選択すると、指定された精度に相当するように末尾がゼロでも常に表示されます。表示されたデータを表す数字の後に単位を追加したくない場合は、**値に単位を追加**のチェックボックスをクリアします。

位置およびサイズ

このセクションには、親フォームのグリッドにあるデータ表示の全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、データ表示の水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

幅フィールドと**高さ**フィールドは、表示される数値に依存するデータ表示の寸法がソフトウェアによって決められるため設定変更できません。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、データ表示の絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、データ表示内のテキストと背景の外観を制御できます。

テキスト色のリストから、データ表示内のテキストに使用する色を選択します。**継承**(デフォルト; フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。


背景色のリストから、データ表示内の背景として使用する色を選択します。**透明**(デフォルト)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

データ表示内のテキストのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォント**リストから、**デフォルト**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体(イタリック体)を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにデータ表示の初期状態を制御できます。デフォルトでは、データ表示は可視表示されて有効になっています。データ表示が非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

グラフィックス

グラフィックス()フォームオブジェクトでは、COMSOL デスクトップの**グラフィックス**ウィンドウでプロットできるものと同じものをプロットすることができます。通常、プロットするためには、モデルの**結果**ブランチの下にあるプロットグループを選択します。また、ジオメトリやメッシュをプロットしたり、プロットタイプを動的に変更したりすることもできます。更に、実行可能なアプリケーションで求解中にプロットする機能(プローブプロットを含む)を含めることもできます。これを実行するには、スタディステップの設定でそれを有効にしてから、プロットしているプロットグループがグラフィックスフォームオブジェクトのソースとして設定されていることを確認します。

グラフィックスオブジェクトには、ズーム、透明度、ライティング、印刷などのボタンを含むプロットツールバーが備わっています。

標準のプロットツールバーをカスタムボタンで拡張または置き換えることができます。

グラフィックスオブジェクトの名前は、**名前**フィールドに入力します。


デフォルトでは、**最初のプロット領域にズーム**のチェックボックスが選択されています。この設定によって、グラフィックスキャンパスの初期状態が空か最初に何かがプロットされるような場合に、アプリケーションの起動時にグラフィックスキャンパスに最初に表示されるプロットが拡大されます。領域へのズーム操作を無効にするには、このチェックボックスをオフにします。




グラフィックスウィンドウでデータピッキングを有効にするには、**データ選択**のチェックボックスをオンにします。データピッキングは、グラフィックスフォームオブジェクトとユーザーとの対話処理をする機能を提供しています。

設定ウィンドウには、次のセクションがあります。

初期グラフィックスコンテンツのソース



このセクションでは、グラフィックウィンドウの初期コンテンツになるプロットを表すノードを指定します。

このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類のデータを表すか、または子ノードを持ちます。フィルタリングされたビューには、明示的選択ノードも含まれます。使用可能なデータノードのリストを拡張するには、**初期グラフィックスコンテンツのソース**セクションのヘッダーで**モデルビルダーへスイッチしデータアクセスをアクティブ**のボタン()をクリックして、モデルビルダーに切り替わってから、アクセスしたいデータのある**モデルビルダー**ブランチ内のノードを選択します。このボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることが可能な設定箇所横に**データソース選択**チェックボックス(緑色の四角形)が表示されます。そのチェックボックスをクリックして選択すると、グラフィックオブジェクトに使用可能なソースノードとしてそのデータが挿入されます。



データを表すノードを選択すると、ツリーの下にある**ソースとして使用**ツールバーボタン()が有効になります。それをクリックするか、代わりに Enter キーを押すか、ダブルクリックするか、ノードを右クリックして**ソースとして使用**を選択して、選択ソースとして追加します。ノードをソースとして選択すると、そのノードは**選択ソース**の下に選択されたソースとして表示されます。**結果**ブランチと全てのジオメトリおよびメッシュノードの下にある全てのプロットグループとプレーヤーアニメーションから選択することができます。また、明示的選択ノードを選択することもできます。これにより、ユーザーはジオメトリックエンティティを選択してその選択をグラフィックス内で直接更新することができます。更に、アプリケーション内の選択入力オブジェクトを使用して、ジオメトリックエンティティの選択をアクティブにすることができます。ツリーの下**ノード編集**ツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。**ソースを消去**のツールバーボタン()をクリックすると、選択したソースが削除されます。

宣言ブランチの下**の文字列**を選択することもできます。この文字列の値が、表示するプロットを制御します。この値は、選択できるノードの一つへのパスを表します。例えば、その値が pg1 の場合、プロットはこのタグを持つプロットグループを示します。値が / GeomList / geom1 の場合は geom1 のタグを持つジオメトリがプロットされ、/ MeshList / mesh1 の場合は mesh1 のタグを持つメッシュがプロットされます。プロットグループに / Results / ResultFeatureList / pg1 という値を使用することもできますが、この構文はむしろ面倒です。

データピッキングのターゲット

設定ウィンドウの上の方にある**データ選択**のチェックボックスを選択している場合には、このセクションのリストから有効なデータピッキングターゲットを選択します。ターゲットは、宣言された倍精度浮動小数点変数であり、グラフィックスウィンドウ内でクリックされた位置のある量を表示するためにフォームで使用されます。このリストには、モデルで定義されたプローブと、**宣言**の下に追加された**グラフィックスデータノード**も含まれています([グラフィックスデータノード](#)を参照)。プローブの場合、例えば、スライダーを使用して、アプリのユーザーが 3D ジオメトリのラインに沿った深さを判断できるようにすることができます。**グラフィックスデータノード**を使用すると、クリックした座標とその場所の結果評価の両方を出力することができます。選択したノードをデータピッキングのターゲットにするには、**ターゲットとして使用**のボタン()をクリックします。**選択ターゲット**の下にそのノードが表示されます。**ノード編集**ボタン()をクリックすると、ツリー内の選択したノードの設定ウィンドウに移動します。

外観

右上のコーナー(例えばロゴタイプなど)にアイコンを追加するには、ライブラリの**画像**リストにある全ての画像を含む**アイコン**リストから使用する画像ファイルを指定するか、**ライブラリに対する画像を追加してここで使用**のボタン()をクリックしてファイルシステム上で使用する画像を検索します。その後、その画像はライブラリの**画像**リストに取り込まれ、このグラフィックスオブジェクトで使用するアイコンとして選択されます。アイコンを使用しない場合は、**アイコン**リストから**なし**を選択します。画像をファイルシステムにエクスポートするには(例えば、別のアプリケーションで使用するような場合)、**エクスポート**ボタン()をクリックします。

このセクションには、背景色の設定も含まれています。**2D プロットの背景**の下で**カラー**リストを使用して、2D プロットと 1D グラフの背景色を選択します。3D 背景では、上から下へのグラデーションがサポートされています。**3D プロットの背景**の下で、**トップカラー**と**ボトムカラー**のリストで上と下の色を選択します。

全てのプロットの背景色について、デフォルトの設定は**デフォルト使用**となっています。この設定は、2D プロットと 1D グラフが白い背景、3D プロットがライトブルーのグラデーションです。定義済みの色のセットに加えて、**透明**と**RGBA**を選択することもできます。RGBA を選択した場合は、**RGBA** のフィールドのカンマで区切られた値として表示されるコンポーネントに、赤; 緑; 青; アルファに対する 0~255 の値(不透明度、0 は完全な透明度、255 は完全に不透明)を入力します。



フォームウィンドウで、あるいはフォームの**プレビュー**を実行したときに、グラフィックスキャンパスはプレーズホルダを表示します。このため、これらの設定の変更は反映されません。**アプリケーションをテスト**を実行して、グラフィック設定に対する変更の効果を確認する必要があります。

状態では、ユーザーがアプリケーションを実行するときにグラフィックスオブジェクトの初期状態を制御できます。デフォルトでは、グラフィックスオブジェクトは可視表示されて有効になっています。グラフィックスオブジェクトが非表示または使用不可になるように初期状態を設定する場合は、**可視または有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

ツールバー

このセクションでは、プロットツールバーに項目を追加し、次のテーブルオプションを有効にすることができます。




位置のリストでは、ツールバーを相対的に配置する位置を、**下**、**上** (デフォルト)、**左**、または**右**から選択します。





ズーム、ビューの変更、レジェンドの表示、シーンのライトと透明度の追加、画像スナップショットの作成とプロットの印刷といった標準のプロットツールバーボタンを表示しない場合は、**標準ツールバーボタンを含める**のチェックボックスをクリアします。デフォルトでは、追加するツールバー項目は、標準ツールバーの後^(注)に配置されます。標準ツールバー項目の前^(注)にカスタムアイテムを配置するには、**カスタムアイテムの前に標準ツールバーを配置**のチェックボックスをクリアにします。

(注) 原文を修正しております。

二つのアイコンサイズから選択することができます。**アイコンサイズ**リストから、**小** (デフォルト) または **大** を選択します。

下のテーブルでは、一つ以上のボタンを追加してプロットツールバーを作成することができます。

- **項目追加ボタン** (): **カスタムツールバーアイテム編集**のダイアログボックスを開き(**カスタムツールバーアイテム編集ダイアログボックス**を参照)、カスタムツールバーボタンを作成してプロットツールバーに追加します。
- **トグルアイテム追加ボタン** (): **カスタムツールバーアイテム編集**のダイアログボックスを開き(**カスタムツールバーアイテム編集ダイアログボックス**を参照)、カスタムツールバートグルボタンを作成してプロットツールバーに追加します。
- **セパレーター追加ボタン** (): ツールバーのボタンのグループ間にセパレーターを追加します。

カスタムツールバーアイテム編集ダイアログボックスでカスタムツールバーボタンの外観や動作を変更するには、テーブル内のボタンを選択して**編集ボタン** () をクリックします。**上へ移動ボタン**および**下へ移動ボタン** ( および ) をクリックすると、ツールバーのボタンの順序を移動して並べ替えることができます。**削除ボタン** () をクリックすると、選択したボタンが削除されます。

テーブルには、追加された各項目の行が含まれており、**名前**、**アイコン**、**テキスト**、および**ツールチップ**の各列にそれぞれ名前、アイコン、テキスト、およびツールチップが表示されます。

位置およびサイズ

このセクションには、親フォームのグリッドにあるグラフィックスオブジェクトの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、グラフィックスオブジェクトの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようになります。

グラフィックスオブジェクトの幅と高さを指定することもできます。**幅**フィールドに幅(ポイント単位で)を、**高さ**フィールドに高さ(ポイント単位で)を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。

最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ0を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。**垂直アライメント**のリストから**全幅**を選択した場合は、**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します(通常は最小サイズ0を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および **位置 y** フィールドを使用して、そのフォームオブジェクト(グラフィックスオブジェクト)の絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および **列スパン**の値として表示できます。




これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

ウェブページ

ウェブページ()フォームオブジェクトは、ユーザーインターフェースの一部としてウェブページのコンテンツを表示することができます。例えば、アプリケーションに関する情報を提供するために使用することができます。ウェブページオブジェクトの名前は、**名前**フィールドに入力します。

設定ウィンドウには、以下のセクションがあります。

ソース

ソースのリストからページのソースを三つの異なる方法で指定することができます。

- デフォルトの**ページオプション**を使用して、リストの下のテキスト領域の<html>のスタートタグと</html>のストップタグで囲まれた範囲に、HTMLコードを入力します。
- **URL** オプションを使用して、オンラインウェブページを**ページ URL** のフィールドに入力して指定します。
- **ファイル**オプションを使用して、HTMLコードを含むローカルファイルリソースを指定します。**ファイル**フィールドにファイル名を入力するか、**参照**をクリックしてローカルファイルシステム上でファイルを検索します。



参照ファイルとは、アプリケーションの実行中には表示されないかもしれないローカルシステム上のファイルリソースを指しています。保存すると、全ての参照ファイルが MPH ファイルに埋め込まれるため、アプリケーションとともに添付ファイルとして配布する必要はありません。実行中のアプリケーションは、いつも最初に埋め込みリソースを探して確認します。

ブラウザープレビューでは、ウェブページの表示がどのように見えるかをプレビューで確認することができます。

外観

外観では、ユーザーがアプリケーションを実行するときにウェブページオブジェクトの初期状態を制御できます。デフォルトでは、ウェブページオブジェクトは可視表示されて有効になっています。ウェブページオブジェクトが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

位置およびサイズ

このセクションには、親フォームのグリッドにあるウェブページの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、ウェブページの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**を選択します。
- **垂直アライメント**のリストから、**中間**、**上**、**下**を選択します。

グリッドモードでは、フォームオブジェクトが自動的に拡大されてセルの水平または垂直方向(該当する場合)の全幅までサイズ変更される、**全幅**を選択することもできます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

ウェブページオブジェクトの幅と高さを指定することもできます。**幅**フィールドに幅(ポイント単位で)を、**高さ**フィールドに高さ(ポイント単位で)を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ0を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。**垂直アライメント**のリストから**全幅**を選択した場合は、**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します(通常は最小サイズ0を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および **位置 y** フィールドを使用して、ウェブページの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。




これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。


セルマージン


セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

画像

画像()フォームオブジェクトを使用すると、フォーム内に画像を追加できます。Ctrl + V を使用して、(例えば、イメージプログラムや別のフォームなどから)クリップボードにコピーした画像をフォームに貼り付けることで、画像フォームオブジェクトを追加することもできます。

名前フィールドに、画像の名前を指定します。名前は、フォーム内のフォームオブジェクト間で一意でなければなりません。**画像ライブラリ**の全ての画像を含む**画像**リストから使用する画像ファイルを指定するか、あるいは、**ライブラリに対する画像を追加してここで使用**のボタン()をクリックしてファイルシステムで使用する画像を検索します。画像は**画像ライブラリ**の一部になり、この**画像**オブジェクトの画像として選択されます。画像を表示したくない場合には、**画像**リストから**なし**を選択します。

エクスポートボタン()をクリックすると、画像をファイルシステムにエクスポートすることができます(例えば、別のアプリケーションで使用する場合)。



参照ファイルとは、アプリケーションの実行中には表示されないかもしれないローカルシステム上のファイルリソースを指しています。保存すると、全ての参照ファイルが MPH ファイルに埋め込まれるため、アプリケーションとともに添付ファイルとして配布する必要はありません。実行中のアプリケーションは、いつも最初に埋め込みリソースを探して確認します。

また、**設定**ウィンドウには、以下のセクションがあります。

位置およびサイズ

このセクションには、親フォームのグリッドにある画像の全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、画像の水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォーム**ツールバーの**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。**幅**フィールドと**高さ**フィールドは、画像の寸法がソフトウェアによって決められるため設定変更できません。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、画像の絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン


セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

外観では、ユーザーがアプリケーションを実行するときに画像の初期状態を制御できます。デフォルトでは、画像は可視表示されて有効になっています。画像が非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

ビデオ

ビデオ()フォームオブジェクトを使用すると、フォーム内にビデオを追加することができます。ビデオは HTML5 で作成されたウェブページです。サポートされているビデオファイル形式は、.mp4、.ogv、および.webm ファイルです。

名前フィールドに、ビデオの名前を指定します。名前は、フォーム内のフォームオブジェクト間で一意でなければなりません。

ビデオリストから使用するビデオファイルを指定するか、あるいは、**ファイルをライブラリを追加してここで使用**のボタン(+)をクリックしてファイルシステムで使用するビデオファイルを検索します。ビデオはファイルライブラリの一部になり、このビデオオブジェクトのビデオとして選択されます。ビデオを表示しない場合には、**ビデオリストからなし**を選択します。**エクスポート**ボタン(📁)をクリックすると、ビデオファイルをファイルシステムにエクスポートすることができます(例えば、別のアプリケーションで使用する場合)。



参照ファイルとは、アプリケーションの実行中には表示されないかもしれないローカルシステム上のファイルリソースを指しています。保存すると、全ての参照ファイルが MPH ファイルに埋め込まれるため、アプリケーションとともに添付ファイルとして配布する必要はありません。実行中のアプリケーションは、いつも最初に埋め込みリソースを探して確認します。

フォームエディターではプレースホルダイメージが表示され、アプリケーションの実行時にビデオが映し出されます。

ビデオコントロール表示のチェックボックスは、デフォルトで選択されています。アプリケーションのユーザーは、このコントロールを使用してビデオを制御して、ビデオの再生や一時停止、サウンドのミュートやミュート解除をすることができます。コントロールを表示せずにビデオを表示するには、このチェックボックスをクリアします。

ユーザーがアプリケーションを実行したときにビデオの再生を自動的にスタートさせる場合には、**自動スタート**のチェックボックスを選択します。

ビデオを繰り返し再生実行させるには、**繰り返しの**チェックボックスを選択します。

最初はビデオをミュート(無音)にしておきたい場合には、**初期にミュートの**チェックボックスを選択します。

また、**設定**ウィンドウには、以下のセクションがあります。

位置およびサイズ

このセクションには、親フォームのグリッドにあるビデオの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、ビデオの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。**幅**フィールドと**高さ**フィールドは、画像の寸法がソフトウェアによって決められるため設定変更できません。

さらに、スケッチモードでは、**位置 x** および **位置 y** フィールドを使用して、ビデオの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン


セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

外観では、ユーザーがアプリケーションを実行するときにビデオの初期状態を制御できます。デフォルトでは、ビデオは可視表示されて有効になっています。ビデオが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

進捗バー

進捗バー()フォームオブジェクトは、アプリケーションのフォームに進捗バーを追加します。進捗バーには、進捗バーを更新するメソッドを使用している実行中のアプリケーションで、進捗を示す値に基づいた進捗が表示されます。進捗バーの進捗状況の更新と表示は、**名前**フィールドに入力した名前を参照することによって制御されます。例えば、次の2行のコードをメソッドエディターウィンドウに追加します。

```
startProgress ( "progressbar1" );  
setProgress ( "progressbar1", 12 );
```

最初の行は、progressbar1 という名前の進捗バーオブジェクトの進行を開始します。2行目は、進捗状況を12%に更新します。進捗状況を制御するための使用可能なコマンドの完全なリストは、**メソッドリボンツールのコードセクション**でアクセスすることができる**言語要素ウィンドウのユーザーインターフェース**の**進捗**で表示されます。また、**モデル進捗を含める**のチェックボックスを選択することで、組み込みモデルの進捗状況(COMSOL マルチフィジックスのメインの進捗状況)をベースにすることができます。オプションで、二つのレベルの進捗状況を使用することができます。二つの進捗レベルを使用(二つの進捗バーを使用して表示)するには、**進捗レベル**のリストから**2つ**を選択します(デフォルト: **1**)。

キャンセルボタンのチェックボックスを選択すると、進捗バーの下に**キャンセル**のボタンが追加され、アプリケーションの進行をキャンセルすることができます。

キャンセルボタンのチェックボックスを選択すると、**キャンセル時にダイアログを閉じる**のチェックボックスを選択またはクリアにすることができます。このチェックボックスを選択すると、ユーザーが**キャンセルボタン**をクリックしたときにダイアログフォームが閉じられます。



メインウィンドウノードの設定ウィンドウで、ソルバーなどの組み込みアクションから進捗情報のための進捗バーを追加することもできます。

三つ目の選択肢は、startProgress (progressBarname)、setProgress (int workDone)、および closeProgress ()などの利用可能なメソッドを使用して、アプリケーション自身の進捗のための進捗ダイアログを追加することです。*Introduction to the Application Builder* のドキュメントで、アプリケーションビルダーのメソッドの全リストを参照してください。

設定ウィンドウには、以下のセクションがあります。

外観

外観では、ユーザーがアプリケーションを実行するときに進捗バーの初期状態を制御できます。デフォルトでは、進捗バーは可視表示されて有効になっています。進捗バーが非表示または使用不可になるように初期状態を設定する場合は、**可視**ま

たは有効化のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

位置およびサイズ

このセクションには、親フォームのグリッドにある進捗バーの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、進捗バーの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向（該当する場合）の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向（該当する場合）の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチセクション**の中の**アレンジメニュー**でアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

進捗バーの幅を指定することもできます。**幅フィールド**に幅（ポイント単位で）を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します（通常は最小サイズ 0 を意味します）。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。**高さフィールド**は、進捗バーの高さがソフトウェアによって決められるため設定変更できません。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、進捗バーの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン（グリッドモードの場合のみ）では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**：フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**（デフォルト）：親フォームに設定されている余白を使用します。
- **カスタム**：フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

ログ

ログ (📄) フォームオブジェクトは、ソルバー操作に関するような、COMSOL マルチフィジックスのソフトウェアからのメッセージを表示するログウィンドウを追加します。ログオブジェクトの名前は、**名前**フィールドに入力します。

標準ログツールバーを含めるのチェックボックスは、デフォルトで選択されています。これによって、COMSOL デスクトップで表示される**ログウィンドウ**のツールバーが含まれます。削除する場合は、このチェックボックスをオフにします。

また、**設定**ウィンドウには、以下のセクションがあります。

位置およびサイズ

このセクションには、親フォームのグリッドにあるログの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、ログの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向（該当する場合）の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向（該当する場合）の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

ログのウィンドウの幅と高さを指定することもできます。幅を調整するには、**幅**のリストから**マニュアル**を選択し、関連するフィールドに幅（ポイント単位で）を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します（通常は最小サイズ 0 を意味します）。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。高さを調整するには、**高さ**のリストから**マニュアル**を選択し、関連するフィールドに高さ（ポイント単位で）を入力します。**垂直アライメント**のリストから**全幅**を選択した場合は、**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します（通常は最小サイズ 0 を意味します）。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、ログウィンドウの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン（グリッドモードの場合のみ）では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**：フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**（デフォルト）：親フォームに設定されている余白を使用します。
- **カスタム**：フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、ログ内のテキストと背景の外観を制御できます。

テキスト色のリストから、ログ内のテキストに使用する色を選択します。**継承**（デフォルト：フォームオブジェクトが配置されているフォームの設定を使用）、**定義済みの基本色**のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。


背景色のリストから、ログ内の背景として使用する色を選択します。**透明**（デフォルト）、**定義済みの基本色**のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

ログ内のテキストのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォント**リストから、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ（ポイント単位）を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体（イタリック体）を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにログの初期状態を制御できます。デフォルトでは、ログは可視表示されて有効になっています。ログが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

メッセージログ

メッセージログ () フォームオブジェクトは、**メッセージ**ウィンドウを追加します。そこでは、アプリケーションが組み込みのメソッド `message(String message)` を使用して実行する操作に関して、ユーザーに通知するメッセージを表示させることができます。メッセージログオブジェクトの名前は、**名前**フィールドに入力します。

標準メッセージログツールバーを含めるのチェックボックスは、デフォルトで選択されています。これによって、COMSOL デスクトップで表示される**メッセージ**ウィンドウのツールバーが含まれます。削除する場合は、このチェックボックスをオフにします。**COMSOL メッセージ表示**のチェックボックスも、デフォルトで選択されています。これによって、COMSOL マルチフィジックスのソフトウェアからのメッセージが**メッセージ**ウィンドウに含められます。このチェックボックスをオフにすると、アプリケーション自体からのメッセージのみが表示されます。

また、**設定**ウィンドウには、以下のセクションがあります。

位置およびサイズ

このセクションには、親フォームのグリッドにあるメッセージログの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、メッセージログの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向 (該当する場合) の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向 (該当する場合) の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォーム**ツールバーの**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

メッセージウィンドウの幅と高さを指定することもできます。幅を調整するには、**幅**のリストから**マニュアル**を選択し、関連するフィールドに幅 (ポイント単位で) を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します (通常は最小サイズ 0 を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。高さを調整するには、**高さ**のリストから**マニュアル**を選択し、関連するフィールドに高さ (ポイント単位で) を入力します。**垂直アライメント**のリストから**全幅**を選択した場合は、**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します (通常は最小サイズ 0 を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および **位置 y** フィールドを使用して、メッセージログの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン (グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照** (デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、メッセージログ内のテキストと背景の外観を制御できます。

テキスト色のリストから、メッセージログ内のテキストに使用する色を選択します。**継承** (デフォルト; フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

背景色のリストから、メッセージログ内の背景として使用する色を選択します。**透明** (デフォルト)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

メッセージログ内のテキストのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォント**リストから、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ (ポイント単位) を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体 (イタリック体) を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにメッセージログの初期状態を制御できます。デフォルトでは、メッセージログは可視表示されて有効になっています。メッセージログが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

結果テーブル


結果テーブル (📄) フォームオブジェクトは、テーブルに数値結果を表示できる結果テーブルを追加します。通常、結果データのソースを COMSOL モデルのグローバル評価として指定します。例えば、あるボタンの設定で「評価 グローバル評価」のコマンドを追加する場合、その更新結果を表示させたい結果テーブルオブジェクトの名前をそこで指定することができます。また、ブーリアン値を指定して、結果テーブルをクリアするか (true: デフォルト)、またはクリアしないか (false) を指定することもできます。結果テーブルオブジェクトの名前は、**名前**フィールドに入力します。

標準結果テーブルツールバーを含めるのチェックボックスは、デフォルトで選択されています。これによって、COMSOL デスクトップで表示される**テーブル**ウィンドウにツールバーが含まれます。削除する場合は、このチェックボックスをオフにします。ツールバーには、テーブルデータの精度と表記法のためのボタン、テーブルとヘッダーをクリップボードにコピーするためのボタン、テーブルデータをファイルにエクスポートするためのボタンがあります。ユーザーがテーブルデータを保存することができるファイルタイプは、テキストファイル (.txt)、CSV ファイル (.csv)、データファイル (.dat)、およびライセンスに LiveLink™ for Excel® が含まれている場合には Microsoft Excel® ファイル (.xlsx) です。

また、**設定**ウィンドウには、以下のセクションがあります。

ソース

このセクションでは、結果データのソースを定義します。このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類のデータを表すか、または子ノードを持ちます。出力を表すノードを選択すると、ツリーの下にある**ソースとして使用**ツールバーボタン (📄) が有効になります。それをクリックするか、代わりに Enter キーを押すか、ダブルクリックするか、ノードを右クリックして**ソースとして使用**

選択して、選択ソースとして追加します。ノードをソースとして選択すると、そのノードは**選択ソース**の下に選択されたソースとして表示されます。**モデル>結果**の下の**計算値とテーブル**ブランチにある全てのノードは、有効な出力ノードです。ツリーの下の**ノード編集**ツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。

位置およびサイズ

このセクションには、親フォームのグリッドにある結果テーブルの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、結果テーブルの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォーム**ツールバーの**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windowsクライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

結果テーブルの幅と高さを指定することもできます。幅を調整するには、**幅**のリストから**マニュアル**を選択し、関連するフィールドに幅(ポイント単位で)を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ0を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。高さを調整するには、**高さ**のリストから**マニュアル**を選択し、関連するフィールドに高さ(ポイント単位で)を入力します。**垂直アライメント**のリストから**全幅**を選択した場合は、**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します(通常は最小サイズ0を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および **位置 y** フィールドを使用して、結果テーブルの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、結果テーブル内のテキストの外観を制御できます。

テキスト色のリストから、テキストに使用する色を選択します。**継承**(デフォルト; フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

結果テーブル内のテキストのフォントとフォントサイズには、**フォーム**ノードのフォント設定がデフォルトとして使用されます。**フォント**リストから、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**

のコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体(イタリック体)を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときに結果テーブルの初期状態を制御できます。デフォルトでは、結果テーブルは可視表示されて有効になっています。結果テーブルが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

フォーム

フォーム (📄) オブジェクトは、**フォーム**へのリンクを含むサブフォームであり、複数の場所でユーザーインターフェースを再利用することができます。フォームオブジェクトの名前は、**名前**フィールドに入力します。

フォームのリストには、このサブフォームがリンクするフォームへの参照先が掲載されています。このフォームオブジェクトが置かれているフォーム以外の既存のフォームオブジェクトをリストから選択するか、フォームにリンクしたくない場合は**なし**を選択します。

サブフォームを境界線で囲むには、**ボーダー追加**のチェックボックスを選択します。そのチェックボックスをオフにすると、サブフォームの境界線は表示されません。境界線(ボーダー)を追加すると、サブフォームがリンクするフォームのタイトルがそこに表示されます。

また、**設定**ウィンドウには、以下のセクションがあります。

位置およびサイズ

このセクションには、親フォームのグリッドにあるフォームオブジェクトの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、フォームオブジェクトの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォーム**ツールバーの**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

フォームオブジェクトの幅と高さを指定することもできます。デフォルトでは、それらは自動的に決められます。幅を調整するには、**幅**のリストから**マニュアル**を選択し、関連するフィールドに幅(ポイント単位で)を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ0を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。高さを調整するには、**高さ**のリストから**マニュアル**を選択し、関連するフィールドに高さ(ポイント単位で)を入力します。**垂直アライメント**のリストから**全幅**を選択した場合は、**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します(通常は最小サイズ0を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および **位置 y** フィールドを使用して、フォームオブジェクトの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および **列スパン** の値として表示できます。




これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン (グリッドモードの場合のみ) では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン** のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照** (デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を **水平** および **垂直** のテキストフィールドに指定します。

フォームコレクション

フォームコレクション () フォームオブジェクトは、タブ形式のペインフォームオブジェクトにペインとして表示されるフォームの集まりで、各ペインにはそのフォームのコンテンツが表示されます。各フォームでは、いろいろなレイアウトオプションのセットを使用することができます。また、ペインを選択するためのリストによって表示するか、あるいは個別のセクションとして表示することもできます。タブ形式のペインには、現在選択されているペインを保持し、メソッドからのペインの切り替えを有効にするためのデータソースが必要です。従って、コンボボックスフォームオブジェクトの場合と同様に、データソースと初期値を選択する必要があります。フォームコレクションオブジェクトの名前は、**名前** フィールドに入力します。

タイプ のリストからは、フォームコレクションのレイアウトのタイプを選択します。

- **タブレイアウト** (デフォルト) では、タブ付きペインを使用してフォームを表示します。
- **リストレイアウト** では、フォームペインの左側にリストが表示され、そこで表示するフォームを選択できます。
- **セクションレイアウト** では、各フォームが別々のセクションに表示されます。
- **タイルまたはタブレイアウト** では、同じフォームで二つの異なる外観があり、いずれのモード (タイルモードまたはタブ付きモード) が表示されるかを制御するブーリアンタイプのソースを持ちます。タブモードは、タイプが **タブ** に設定されたフォームコレクションと同じです。タイルモードでは、全てのフォームが同時にグリッドに表示されます。




onLoad および onClose イベントメソッドは、フォームを開いたり閉じたりするときや、フォームコレクション内のタブを切り替えるときにアクティブになります。onClose メソッドは、フォームを使用してあるタブに切り替えるときに実行され、onLoad メソッドはフォームを使用してあるタブから切り替えるときに実行されます。イベントは、**タブ** または **リスト** のレイアウトタイプが設定されたフォームコレクションに対してのみ実行されます。レイアウトタイプが **セクション** の場合には、イベントは実行されません。

また、**設定** ウィンドウには、以下のセクションがあります。

タイルまたはタブ




このセクションは、上記の **タイプ** リストから **タイルまたはタブ** を選択した場合にのみ使用可能です。


ツリービューから、タイル表示とタブ表示の切り替えのソースとして使用するブーリアン変数を選択します。**ソースとして使用** のボタン () をクリックするか、または右クリックして **ソースとして使用** を選択すると、選択したブーリアン変数がフォームコレクションの表示を切り替えるソースになります。

タイルモード設定 では、タイルモードのいくつかの設定を指定できます。デフォルトでは、**タイルモードで境界を追加** のチェックボックスが選択されており、フォームオブジェクトの周りに罫線が追加されます。**タイル方式** のリストでは、**列最初** (デフォルト) または **行最初** を選択して、タイリングされるフォームオブジェクトの並べる順序を制御します。タイルモードの列数または

行数は、それぞれ、**列数**フィールド(デフォルト: 2 列)または**行数**フィールド(デフォルト: 2 行)で指定することができます。二つのタイル方式の列数または行数は、同期した同じ設定になります。

アクティブペイン選択

このセクションでは、**タイプ**リストが**タブ**または**リスト**に設定されているときに、フォームコレクションの表示枠(ペイン)を制御するアクティブペイン選択を定義します。このセクションには、**アプリケーションビルダー**ウィンドウのツリーのフィルター処理されたツリーが表示されます。ノードは、何らかの種類データを表すか、または子ノードを持ちます。フォームコレクションの場合、**宣言**の下で定義した文字列変数をアクティブペイン選択として使用することができます。それを表すノードを選択すると、ツリーの下にある**ソースとして使用**ツールバーボタン()が有効になります。ノードを右クリックして、**ソースとして使用**を選択することもできます。また、**ソース**セクションのヘッダーの**作成**ボタン()をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。**変数を作成および使用**のダイアログボックスが開き、ソースのデータタイプ(該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下の**ノード編集**ツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。




ノードをソースとして選択すると、そのノードは**選択ソース**の下に選択されたソースとして表示されます。アクティブペイン選択としてのデータソースが必要ない場合、ソースツリーの下にある**ソースを消去**のツールバーボタン()をクリックします。表示するタブやリストをクリックする以外の方法で表示枠(ペイン)を制御するには、ここでソースを選択する必要があります。



データソースは、タブとリストタイプのフォームコレクションにのみ適用されます。セクションのタイプとして表示されるフォームコレクションの場合、データソースの設定は効果がなく、ソースは適用されません。

ペイン^(注)

(注) 原文は「ペイン」ですが、COMSOL デスクトップでは「パネル」となっています。

選択したフォームをペインとして使用のリストで、**選択元**のリスト領域から選択したフォームオブジェクトを  ボタンを使用して**選択**のリスト領域に追加します。各フォームは、リストに表示される順序でのペインを表します。**デフォルトペイン**のリストから、選択したフォームオブジェクトの一つを選択してデフォルトペインにします。データソースを選択していた場合、デフォルトペインが設定されていてもデータソースが初期化され、データ宣言で指定されているデフォルトが全て上書きされます。フォームコレクションに接続されているデータソースに許可される値は、form1 や form2 などのフォームの名前です。**選択**の領域リストに表示中のフォームの順序を変更するには、**上へ移動**()および**下へ移動**()ボタンを使用します。

位置およびサイズ

このセクションには、親フォームのグリッドにあるフォームコレクションの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、フォームコレクションの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォーム**ツールバーの**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

フォームコレクションの幅と高さを指定することもできます。デフォルトでは、それらは自動的に決められます。幅を調整するには、**幅**のリストから**マニュアル**を選択し、関連するフィールドに幅(ポイント単位で)を入力します。**水平アライメント**の

ストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ0を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。高さを調整するには、**高さ**のリストから**マニュアル**を選択し、関連するフィールドに高さ(ポイント単位で)を入力します。**垂直アライメント**のリストから**全幅**を選択した場合は、**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します(通常は最小サイズ0を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、フォームコレクションの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、フォームコレクション内のテキストの外観を制御できます。


テキスト色のリストから、テキストに使用する色を選択します。**継承**(デフォルト: フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

フォームコレクション内のテキストのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォント**リストから、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体(イタリック体)を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにフォームコレクションの初期状態を制御できます。デフォルトでは、フォームコレクションは可視表示されて有効になっています。フォームコレクションが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。


カードスタック




カードスタック()は、カードだけを含む特別なタイプのフォームオブジェクトです。カードスタックは、一度に1枚ずつ表示するために、カードスタック内のカード間をめぐって入れ替えることができます。例えば、アプリケーションのイベントや結果に応じて、異なる画像やテキストを表示できます。どのカードを表示するかを制御するデータソースに、カードスタックを関連付けます。各カードには、カードスタックのデータソースと比較する値を指定します。カードスタックには、値が一致する最初のカードが表示されます。どのカードもデータソースと一致しない場合には、何も表示されません。カードには、ローカルカードと、既存のフォームオブジェクトへの参照であるカードの2種類があります。

カードスタックオブジェクトの名前は、**名前**フィールドに入力します。

設定ウィンドウには、以下のセクションがあります。

アクティブ情報カード選択

このセクションでは、アクティブ情報カード選択のデータソースを定義します。このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類のデータを表すか、または子ノードを持ちます。カードスタックでは、通常、**パラメーター**の下の使用可能なパラメーター、**変数**の下の変数、および**宣言**ブランチの下で定義されたデータノードのみが表示されます。使用可能なデータノードのリストを拡張するには、**アクティブ情報カード選択**セクションのヘッダーで**モデルビルダーへスイッチしデータアクセスをアクティブ**のボタン()をクリックして、モデルビルダーに切り替わってから、アクセスしたいデータのある**モデルビルダー**ブランチのノードを選択します。このボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることが可能な設定箇所の横に**データソース**選択チェックボックス(緑色の四角形)が表示されます。そのチェックボックスをクリックして選択すると、カードスタックに使用可能なソースノードとしてそのデータが挿入されます。

アクティブ情報カード選択のソースを表すノードを選択すると、ツリーの下にある**ソースとして使用**ツールバーボタン()が有効になります。ノードを右クリックして、**ソースとして使用**を選択することもできます。また、**アクティブ情報カード選択**セクションのヘッダーの**作成**ボタン()をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。**変数を作成および使用**のダイアログボックスが開き、ソースのデータタイプ(該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下**ノード編集**ツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。

ノードをソースとして選択すると、そのノードは**選択ソース**の下に選択されたソースとして表示されます。


このフォームでは、カードスタックオブジェクトは、カードスタックのデータソースのデフォルト値と一致するアクティブ値を持つカードを表示します。




複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースのデフォルト値は期待通りとは限りません。あるフォームオブジェクトのデフォルト値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。


カード


このセクションのテーブルには、カード(**カード**の列)とそれに関連するアクティブ値(**アクティブ値**の列)が含まれています。スタックが表示するカードは、このセクションで入力するアクティブ値によって決定されます。その際、この値はソースの値と照合されます。全てのカードについて、**アクティブ値**の列にアクティブ値を入力できます。ローカルカードの場合は、**カード**の列でカードの名前を編集することもできます。

カード追加のボタン()をクリックして、カードをテーブルに追加します。**カード追加**のダイアログボックスが開き、そこで新しいカードを指定することができます。

カードタイプのリストから**ローカル**(デフォルト)を選択してカードスタックのローカルカードを作成するか、**既存のフォーム**を選択して既存のフォームをカードとして使用します。ローカルカードの場合は、**名前**フィールドにカード名を入力します。フォームの場合は、フォームリストから既存のフォームの一つを選択します。必要に応じて、**アクティブ値**のフィールドにアクティブ値として機能する一意の値を入力します。

選択したカードを削除するには、**削除**ボタン()をクリックします(または、テーブル内のそのカードの入力内容を右クリックします)。

編集ボタン()をクリックすると、個々のカードを編集することができます。また、テーブル内のそのカードの入力内容を右クリックして、**編集**をクリックするか、フォームウィンドウでカードスタックオブジェクトを右クリックして、例えば**編集 card1** を選択することもできます。

複製のボタン()をクリックすると、カードスタック内のカードを複製することができます。また、フォームウィンドウのカードスタックオブジェクトを右クリックし、例えば**複製 card2** を選択するなどして、スタックにカードを複製することもできます。

外観

外観では、ユーザーがアプリケーションを実行するときにカードスタックの初期状態を制御できます。デフォルトでは、カードスタックは可視表示されて有効になっています。カードスタックが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

位置およびサイズ

このセクションには、親フォームのグリッドにあるカードスタックの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、カードスタックの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向（該当する場合）の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向（該当する場合）の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

カードスタックの幅と高さを指定することもできます。デフォルトでは、それらは自動的に決められます。幅を調整するには、**幅**のリストから**マニュアル**を選択し、関連するフィールドに幅（ポイント単位で）を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します（通常は最小サイズ 0 を意味します）。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。高さを調整するには、**高さ**のリストから**マニュアル**を選択し、関連するフィールドに高さ（ポイント単位で）を入力します。**垂直アライメント**のリストから**全幅**を選択した場合は、**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します（通常は最小サイズ 0 を意味します）。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、カードスタックの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。




これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。


セルマージン

セルマージン（グリッドモードの場合のみ）では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**：フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**（デフォルト）：親フォームに設定されている余白を使用します。
- **カスタム**：フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

カード

カード () は、カードスタックの子である特別なフォームオブジェクトです。ローカルカードオブジェクトを編集する際、カードに表示されるフォームオブジェクト（画像やテキストなど）を追加するための新しいフォームウィンドウにそれが表示されます。

既存のフォームを使用するカードを編集する場合には、そのフォームウィンドウが開きます。カードは条件を持ちます。その条件が成立すると、カードスタックにそのカードが表示されます。一つの条件が真の場合は表示するカードの検索が停止されるので、真の条件を持つ他の全てのカードは表示されません。ローカルカードの場合は、名前フィールドにそのカードオブジェクトの名前を入力します。カードが属するカードスタックに移動するには、**カードスタック**へのボタン()をクリックします。

ローカルカードの設定ウィンドウには、以下のセクションがあります。

アクティベーション

アクティベート値のフィールドにカードをアクティベート(有効)にする値を入力します。この値が親の**カードスタック**ノードで選択されたデータソースの値と一致すると、そのカードが表示されます(親のカードスタックでアクティベート値を編集することもできますが、その値はここでの**アクティベート**値のフィールドでデフォルトになります)。

マージン

このセクションでは、必要に応じてカードの**水平**と**垂直**のマージンを調整することができます(デフォルト: 20 ピクセル)。

スケッチグリッド



スケッチグリッドセクションは、フォームのスケッチモードを選択した場合にのみ使用可能です。

このセクションでは、スケッチモードで表示できるグリッドの設定([グリッド線の表示とグリッドへのスナップ](#)を参照)とそのグリッドへのフォームオブジェクトのスナップについて説明しています。

列幅(デフォルト: 100 ピクセル)と**行高さ**(デフォルト: 20 ピクセル)のフィールドに値を入力して、グリッドサイズを指定します。

グリッドラインを左余白および上余白に揃えるには、**マージンに対するアライングリッド**のチェックボックスを選択します。

スナップゾーンのスライダーによって、グリッドにスナップするためにフォームオブジェクトのサイズを変更するときに必要な正確さを制御します。デフォルトでは、スナップゾーンは最大値に設定されているため、オブジェクトはグリッドにスナップする際にすばやくサイズ変更されます。必要に応じて、スナップゾーンを小さくするには、スライダーを**大**から**小**に動かします。

グリッドに対してのみスナップのチェックボックスを選択すると、フォームオブジェクトのサイズ変更がグリッドにのみスナップされ、他のフォームオブジェクトの境界にはスナップされません。

含まれているフォームオブジェクトのためのグリッドレイアウト



含まれているフォームオブジェクトのためのグリッドレイアウトのセクションは、フォームのグリッドモードが選択されている場合にのみ使用可能です。

このセクションには、グリッド内の列用と行用の二つのテーブルがあります。各テーブルの**列**と**行**の列には、それぞれ左上から 1 から始まる列番号および行番号があります。フォーム内の各行と各列の空白をどのように埋めるかを制御することができます。各テーブルには、**幅**(列用テーブル)または**高さ**(行用テーブル)の列があり、**フィット**(デフォルト)、**自動拡大縮小**、**固定**のオプションを含むリストとなっています。

自動拡大縮小のオプションを使用すると、グリッド内の列または行を展開して、ユーザーがフォームのサイズを大きくしたときにフォーム内のスペースを取ることができます。**フィット**または**固定**が設定された列または行では、自動拡大縮小はしません。

固定オプションは、グリッドレイアウトに列または行の特定の幅または高さを指定し、そのサイズはテーブルの 3 番目の**サイズ**の列に指定します。他のオプションの場合には、この 3 番目の列は無視されます。追加された幅(ピクセル)または高さ(ピクセル)は、フォームウィンドウの列または行のヘッダーに表示されます。

列を継承のリストから、列設定を継承するフォームオブジェクトを選択します。デフォルトは**なし**です。つまり、列の設定は継承されません。

外観

このセクションでは、カードのテキスト色と背景色と背景画像の外観を制御できます。

テキスト色のリストから、カードのテキストに使用する色を選択します。**継承**(デフォルト; フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

背景色のリストから、カードの背景として使用する色を選択します。**透明**(デフォルト)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

背景画像のリストから、カードの背景として使用する色を選択します。**透明**(デフォルト)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

背景画像リストから、カード内の画像として使用する背景画像を選択します。デフォルトはなしで、背景画像は使われません。画像を画像ライブラリに追加して背景画像として使用するには、**ライブラリに対する画像を追加してここで使用**のボタン(+)をクリックします。**エクスポート**ボタン(📄)をクリックすると、背景画像を PNG ファイルで保存することができます。

状態では、ユーザーがアプリケーションを実行するときにカードの初期状態を制御できます。デフォルトでは、カードは可視表示されて有効になっています。カードが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

ファイルインポート

ファイルインポート(📁)は、アプリケーションが何らかの目的(例えば、データ入力の実行)で使用するインポートファイルをブラウジングして選択するための特別なフォームオブジェクトです。アプリケーションビルダーのファイルライブラリにファイルを追加することもできます(詳細については、[ライブラリブランチ](#)を参照)。**名前**フィールドにファイルインポートオブジェクトの名前を入力します。

ボタンテキストのフィールドで、参照するボタンに表示するテキストを入力します。

ダイアログタイトルのフィールドには、ファイルインポートダイアログのタイトルを入力します。ダイアログのタイトルは、ボタンのツールチップでもあります。

ファイルタイプのリストに、ファイルブラウザーが表示するファイルのタイプを追加します。**追加**ボタン(+)をクリックすると、ファイルタイプとその関連拡張子のリストを含むダイアログボックスが開きます。この追加によって、ブラウザーに表示するファイルリストをフィルタリングすることができます。

リストのコマンドを整理したり削除するには、**上へ移動**(↑)、**下へ移動**(↓)、および**削除**(✖)のツールバーボタンを使用します。

デフォルトでは**ファイル名の入力を許可**のチェックボックスが選択されているため、ユーザーは一致するファイルのリストからファイルを選択するだけでなく、ブラウザーでファイル名を入力することができます。

また、**設定**ウィンドウには、以下のセクションがあります。

ファイル行先

このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されるツリーが含まれています。ノードは、何らかのファイルの行先を表すか、または子ノードを持ちます。リストには、**宣言**ノードの下に宣言をする**ファイル**ノードが含まれています。また、ファイルのブラウズをサポートする**モデル**の設定も含まれています。このような設定は、通常、例えば**補間機能**や**ジオメトリ**の**インポート**ノードなどでの設定で表示される、**参照**ボタン付きのテキストフィールドです。ツリー内では、**ファイル名**というサブノード(例えば、**補間**ノードの下)として表示されます。ファイルの行先を表すノードを選択すると、ツリーの下にある**ソースとして使用**のツールバーボタン(📄)が有効になります。

ノードを右クリックし、**ソースとして使用**を選択することもできます。ノードをソースとして選択すると、そのノードが**選択ソース**の下に選択されたソースとして表示されます。ファイルにアクセスするためのファイルスキームが、**使用のアクセス**の横に表示されます。例えば、タグ int1 を持つ**補間**ノードの場合、スキームは upload:/// int1 / filename です。また、**ファイル行先**セクションのヘッダーにある**作成**ボタン(+)をクリックして、新規変数宣言を作成してそれをファイルの行先として使用することもできます。**ファイルを作成および使用**のダイアログボックスが開き、ファイル名を指定できます。名前は、既存のファイル名と競合することはできません。ツリーの下**のノード編集**ツールバーボタン(🔧)をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。

位置およびサイズ

このセクションには、親フォームのグリッドにあるファイルインポートオブジェクトの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、ファイルインポートオブジェクトの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォーム**ツールバーの**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

ファイルブラウザの幅を指定することもできます。**幅**フィールドに幅(ポイント単位で)を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ 0 を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。**高さ**の値は、ソフトウェアによって決められます。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、ファイルブラウザの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、ファイルインポート内のテキストと背景の外観を制御できます。

テキスト色のリストから、ファイルインポートオブジェクトのテキストに使用する色を選択します。**継承**(デフォルト: フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。




背景色のリストから、ファイルインポート内の背景として使用する色を選択します。透明（デフォルト）、定義済みの基本色のどれかを選択します。また、カスタムを選択して、カラーパレットからカスタムカラーを選択することもできます。

ファイルインポート内のテキストのフォントとフォントサイズには、フォームノードのフォント設定がデフォルトとして使用されます。フォントリストから、デフォルトフォントまたはリストにある使用可能なフォントを選択してください。必要に応じて、フォントサイズのコンボボックスでフォントサイズ（ポイント単位）を選択するか入力します。デフォルトでは、フォントのデフォルトサイズが使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体（イタリック体）を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにファイルインポートオブジェクトの初期状態を制御できます。デフォルトでは、ファイルインポートオブジェクトは可視表示されて有効になっています。ファイルインポートオブジェクトが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

イベント


特定のタイプのフォームオブジェクトでは、データ入力などのイベントが発生したときに実行するメソッドを指定できます。**データ変更時**のリストには、なし（デフォルト）と使用可能なメソッドが含まれています。イベントのローカルメソッドを追加するには、**データ変更時**のリストの右側にある**ローカルメソッド作成**のボタン（）をクリックするか、またはファイルインポートオブジェクトを右クリックします。すると、**データ変更時**リストで選択したメソッドが**ローカルメソッド**に変更されます。ローカルメソッドをメソッドエディターウィンドウで開くには、**ソースへ**のボタン（）をクリックします。空の **onDataChange** エディターウィンドウが開かれ、そこでローカルメソッドを定義することができます。（このように最初にローカルメソッドを作成したとき、このウィンドウは自動的に開きます。）ローカルメソッドを削除するには、**ローカルメソッドを削除**のボタン（）をクリックします。ファイルインポートオブジェクトから呼び出されるメソッドは、ユーザーが選択した新規のクライアントファイル名を持った文字列の引数をサポートしています。また、ファイルインポートオブジェクトを右クリックしてローカルメソッドを作成するか、（**メソッド編集**または**ローカルメソッド編集**を選択して）コマンドに関連付けられたメソッドを開きます。

ファイルインポートオブジェクトで Ctrl + Alt を押しながらクリックした場合、

- **データ変更時**リストがメソッドに設定されている場合には、メソッドエディターウィンドウが開かれます。
- **データ変更時**リストがなしに設定されている場合には、（必要に応じて）ローカルメソッドを作成してそのリストに**ローカルメソッド**が設定され、ローカルメソッドのエディターウィンドウが開かれます。

データが変更されるとトリガーされるイベントでは、新しいデータ値がデータソースに格納された後にイベントが発生されます。

情報カードスタック


情報カードスタック（）は、最後に実行されたときの計算時間（ソリューションがまだ利用されていない場合）やいくつかの他のソリューションステータス情報といったような、アプリケーションに関する情報を表示するカードが組み込まれたカードスタックを表示するフォームオブジェクトです。カードスタックは、一度に 1 枚ずつ表示するために、カードスタック内のカード間をめぐって入れ替えることができます。どのカードを表示するかを制御するデータソースに、カードスタックを関連付けます。各カードには、カードスタックのデータソースと比較する値を指定します。カードスタックには、値が一致する最初のカードが表示されます。




どのカードもデータソースと一致しない場合には、何も表示されません。情報カードスタックオブジェクトの名前は、名前フィールドに入力します。

設定ウィンドウには、以下のセクションがあります。

アクティブ情報カード選択

このセクションでは、アクティブ情報カード選択のデータソースを定義します。このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類のデータ

を表すか、または子ノードを持ちます。カードスタックでは、通常、**パラメーター**の下の使用可能なパラメーター、**変数**の下の変数、および**宣言**ブランチの下で定義されたデータノードのみが表示されます。使用可能なデータノードのリストを拡張するには、**アクティブ情報カード選択**セクションのヘッダーで**モデルビルダーヘスイッチしデータアクセスをアクティブ**のボタン()をクリックして、モデルビルダーに切り替わってから、アクセスしたいデータのある**モデルビルダー**ブランチのノードを選択します。このボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることが可能な設定箇所の横に**データソース**選択チェックボックス(緑色の四角形)が表示されます。そのチェックボックスをクリックして選択すると、カードスタックに使用可能なソースノードとしてそのデータが挿入されます。


アクティブ情報カード選択のソースを表すノードを選択すると、ツリーの下にある**ソース**として**使用**ツールバーボタン()が有効になります。ノードを右クリックして、**ソース**として**使用**を選択することもできます。また、**アクティブ情報カード選択**セクションのヘッダーの**作成**ボタン()をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。**変数**を**作成**および**使用**のダイアログボックスが開き、ソースのデータタイプ(該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下での**ノード編集**ツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。ノードをソースとして選択すると、そのノードは**選択**ソースの下に選択されたソースとして表示されます。





複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースのデフォルト値は期待通りとは限りません。あるフォームオブジェクトのデフォルト値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。

情報カード

このセクションのテーブルには、情報カードに関連するアクティブ値(**アクティブ値**の列)、アイコン(**アイコン**の列)、およびカードに表示するテキスト(**テキスト**の列)が含まれています。スタックが表示するカードは、このセクションで入力するアクティブ値によって決定されます。その際、この値はソースの値と照合されます。

情報カード追加のボタン()をクリックして、カードをテーブルに追加します。**情報カード**のダイアログボックスが開き、そこで定義済みのカードを追加することができます。**追加**>>および<<**削除**のボタンをクリックして、それぞれ情報カードを追加または削除することができます。**カスタムカード**のボタンをクリックすると、**情報カード編集**のダイアログボックスが開かれ、そこで新しいカードのためにアクティブ値を定義し、アイコンを選択し、カードに表示するテキストを定義し、テキストの色を選択(フォームのテキスト色を使用する場合は**継承**を選択)することができます。**OK**をクリックすると、そのカードの設定が保存され、情報カードスタック内のカードに追加されます。

選択したカードを削除するには、**削除**ボタン()をクリックするか、テーブル内のそのカードの入力内容を右クリックします。

情報カード編集ボタン()をクリックすると、表示される**情報カード編集**ダイアログボックスでそのカードを編集することができます。このダイアログボックスでは、アクティブ値の定義、アイコンの選択、カードに表示するテキストの定義、およびテキストの色の選択(フォームのテキスト色を使用する場合は**継承**を選択)が可能です。**OK**をクリックすると、そのカードの設定が保存されます。

位置およびサイズ

このセクションには、親フォームのグリッドにある情報カードスタックの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、情報カードスタックの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。そ

の場合、フォームツールバーのスケッチセクションの中のアレンジメニューでアラインメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アラインメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

情報カードスタックの幅と高さを指定することもできます。デフォルトでは、それらは自動的に決められます。幅を調整するには、幅のフィールドに幅(ポイント単位で)を入力します。水平アラインメントのリストから全幅を選択した場合は、最小幅のリストが表示されます。(注) 最小幅を自動的に計算するには自動を選択します(通常は最小サイズ0を意味します)。または、マニュアルを選択して、その下のテキストフィールドに最小幅を指定します。高さを調整するには、高さのフィールドに高さ(ポイント単位で)を入力します。垂直アラインメントのリストから全幅を選択した場合は、最小高さのリストが表示されます。(注) 最小高さを自動的に計算するには自動を選択します(通常は最小サイズ0を意味します)。または、マニュアルを選択して、その下のテキストフィールドに最小高さを指定します。

(注) 下線部分は、原文を修正しております。

さらに、スケッチモードでは、位置 x および位置 y フィールドを使用して、情報カードスタックの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、行、列、行スパン、および列スパンの値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。セルマージンのリストから、以下を選択します。

- なし: フォームオブジェクトの周りに余白を使用しません。
- 親フォーム参照(デフォルト): 親フォームに設定されている余白を使用します。
- カスタム: フォームオブジェクトの余白を水平および垂直のテキストフィールドに指定します。

外観

このセクションでは、情報カードスタック内のテキストと背景の外観を制御できます。


テキスト色のリストから、情報カードスタック内のテキストに使用する色を選択します。継承(デフォルト; フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、カスタムを選択して、カラーパレットからカスタムカラーを選択することもできます。

背景色のリストから、情報カードスタック内の背景として使用する色を選択します。透明(デフォルト)、定義済みの基本色のどれかを選択します。また、カスタムを選択して、カラーパレットからカスタムカラーを選択することもできます。

情報カードスタック内のテキストのフォントとフォントサイズには、フォームノードのフォント設定がデフォルトとして使用されます。フォントリストから、デフォルトまたはリストにある使用可能なフォントを選択してください。必要に応じて、フォントサイズのコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントのデフォルトサイズが使用されます。

太字のフォントを使用するにはボールドのチェックボックスを選択し、斜体(イタリック体)を使用するにはイタリックのチェックボックスを選択します。あるいは、そのテキストに下線を付けるにはアンダーラインのチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときに情報カードスタックの初期状態を制御できます。デフォルトでは、情報カードスタックは可視表示されて有効になっています。情報カードスタックが非表示または使用不可になるように初期状態を設定する場合は、可視または有効化のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。




配列入力 () は、配列入力 (ベクトル入力) を入力するための入力テーブルを持つように定義されたフォームオブジェクトです。**配列入力** は配列をデータソースとしてサポートしています。オプションのラベル、シンボル、単位を追加することもできます。配列入力オブジェクトの名前は、**名前** フィールドに入力します。

長さ フィールドには、配列の長さを正の整数 (デフォルト: 3) として入力します。最大長は 1000 です。

ベクトル表示 のリストでは、**テーブル** (デフォルト) を選択して配列コンポーネントをテーブルとして表示するか、**成分** を選択して各配列コンポーネントをラベル付きの別々の入力フィールドとして表示するかを選択します。

また、**設定** ウィンドウには、以下のセクションがあります。

ソース

このセクションでは、配列入力のソースを定義します。このセクションには、**アプリケーションビルダー** ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類のデータを表すか、または子ノードを持ちます。配列入力では、例えば、**宣言** の下で定義した配列変数をリストに含みます。データを表すノードを選択すると、ツリーの下にある **ソース** として **使用** ツールバーボタン () が有効になります。それをクリックするか、代わりに Enter キーを押すか、ダブルクリックするか、ノードを右クリックして **ソース** として **使用** を選択して、選択ソースとして追加します。データソースの初期値が配列入力オブジェクトに表示されます。また、**ソース** セクションのヘッダーの **作成** ボタン () をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。**変数を作成および使用** のダイアログボックスが開き、ソースのデータタイプ (該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下ノード **編集** ツールバーボタン () をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。

ノードをソースとして選択すると、そのノードは **選択** ソースの下に選択されたソースとして表示されます。



複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースのデフォルト値は期待通りとは限りません。あるフォームオブジェクトのデフォルト値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。




初期値 のリストから、**データソース参照** を選択して選択したデータソースで指定された値を使用するか、または **カスタム** 値を選択してその下の関連するテーブルに配列のコンポーネントの初期値を入力します。

レイアウトオプション

このセクションでは、オプションのラベルと単位を配列入力に追加するための設定を行います。

ラベル位置 のリストを使用して、ラベルを配置します。オプションは、**上** (デフォルト)、**左**、および **ラベルなし** です。最初の二つのオプションでは、**ラベルテキスト** のフィールドに所望のラベルを入力することができます。

ラベルの位置がテーブルの上にある場合は、**シンボルを含める** のチェックボックスを選択することによって、テーブルの左側にシンボルを含めることができます。ラベルの位置がテーブルの左であり、シンボルの位置と重なっている場合には、この設定を見ることはできません。**シンボル (LaTeX エンコーディッド)** のフィールドには、LaTeX のシンタックスを使用してシンボルを入力します。

最後のオプションとして、**単位を含む** のチェックボックスを選択すると、テーブルの右側に単位記号が追加されます。単位を追加するには、**量を選択** ボタン () をクリックして **物理量** ダイアログボックスを開き、使用する物理量を検索します。また、ダイアログボックスの上部にあるテキストフィールドに検索文字列を入力し、**フィルター** ボタン () をクリックして物理量のリストをフィルタリングすることもできます。例えば、**電位** と入力して **フィルター** ボタンをクリックすると、ある種の可能性のある物理量のみがリスト表示されます。または、**カスタム単位** ボタン () をクリックして、テキストフィールドに単位 (例えば m/s^2) を入力します (物理量が **カスタム単位** になります)。



配列入力をコンポーネント(成分)として表示することを選択した場合、**ラベルテキストフィールド**は**コンポーネントラベル**のテーブルに変わり、各コンポーネントのラベルごとに一つの行が表示されます。同様に、**シンボル (LaTeX エンコーデッド)**のフィールドは、コンポーネントシンボルのそれぞれに対して一つの行を持つ**コンポーネントシンボルの**テーブルに変わります。

位置およびサイズ

このセクションには、親フォームのグリッドにある配列入力の全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、配列入力の水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

配列入力の幅を指定することもできます。**幅**フィールドに幅(ポイント単位で)を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ 0 を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。**高さ**フィールドは、配列入力のフィールドの高さがソフトウェアによって決められるため設定変更できません。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、配列入力の絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に従って更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、配列入力内のテキストと背景の外観を制御できます。

テキスト色のリストから、**フォームノード**の設定からテキストの色を継承する**継承**(デフォルト)を選択するか、**黒**などの定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

背景色のリストから、配列入力内の背景として使用する色を選択します。**透明**(デフォルト)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

配列入力フィールド内のテキストのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォント**リストから、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォント**

サイズのコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体(イタリック体)を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときに配列入力の初期状態を制御できます。デフォルトでは、配列入力は可視表示されて有効になっています。配列入力が非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

イベント

特定のタイプのフォームオブジェクトでは、データ入力などのイベントが発生したときに実行するメソッドを指定できます。**データ変更時**のリストには、**なし**(デフォルト)と使用可能なメソッドが含まれています。イベントのローカルメソッドを追加するには、**データ変更時**のリストの右側にある**ローカルメソッド作成**のボタン(+)をクリックするか、または配列入力オブジェクトを右クリックします。すると、**データ変更時**リストで選択したメソッドが**ローカルメソッド**に変更されます。ローカルメソッドをメソッドエディターウィンドウで開くには、**ソースへ**のボタン(📄)をクリックします。空の `onDataChange` エディターウィンドウが開かれ、そこでローカルメソッドを定義することができます。(このように最初にローカルメソッドを作成したとき、このウィンドウは自動的に開きます。)ローカルメソッドを削除するには、**ローカルメソッドを削除**のボタン(✖)をクリックします。配列入力オブジェクトから呼び出されるメソッドは、ユーザーが選択した新規のクライアントファイル名を持った文字列の引数をサポートしています。また、配列入力オブジェクトを右クリックしてローカルメソッドを作成するか、(**メソッド編集**または**ローカルメソッド編集**を選択して)コマンドに関連付けられたメソッドを開きます。

配列入力オブジェクト上で `Ctrl + Alt` を押しながらクリックした場合、

- **データ変更時**リストがメソッドに設定されている場合には、メソッドエディターウィンドウが開かれます。
- **データ変更時**リストが**なし**に設定されている場合には、(必要に応じて)ローカルメソッドを作成してそのリストに**ローカルメソッド**が設定され、ローカルメソッドのエディターウィンドウが開かれます。

データが変更されると、新しいデータ値がデータソースに格納された後にイベントが発生されます。

ラジオボタン




ラジオボタン(📻)フォームオブジェクトは、固定数のオプションからなるラジオボタン(オプションボタン)のグループを表示し、そこから一つを選択することができます。通常、いくつかのオプションがある場合に利用すると便利です(多くのオプションがある場合には、代わりにリストボックスやコンボボックスを使用することも検討してください)。ラジオボタンオブジェクトの名前は、**名前**フィールドに入力します。

方向リストから**垂直**(デフォルト)または**水平**を選択すると、ラジオボタンを縦または横に並べることができます。

設定ウィンドウには、以下のセクションがあります。

ソース

このセクションでは、ラジオボタンのデータソースを定義します。このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類 of データを表すか、または子ノードを持ちます。ラジオボタンでは、モデルの変数、およびアプリケーションビルダーの**宣言**の下にある変数を選択することができます。そこには**単位セット**も含まれます。使用可能なデータノードのリストを拡張するには、**ソースセクション**のヘッダーで**モデルビルダーへスイッチしデータアクセスをアクティブ**のボタン(📄)をクリックし、モデルビルダーに切り替わってから、アクセスしたいデータのある**モデルビルダー**内のノードを選択します。このボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることが可能な設定箇所の横に**データソース選択**チェックボックス(緑色の四角形)が表示されます。そのチェックボックスをクリックして選択すると、ラジオボタンに使用可能なソースノードとしてそのデータが挿入されます。

データを表すノードを選択すると、ツリーの下にあるソースとして使用ツールバーボタン()が有効になります。それをクリックするか、代わりにEnter キーを押すか、ダブルクリックするか、ノードを右クリックしてソースとして使用を選択して、選択ソースとして追加します。また、ソースセクションのヘッダーの作成ボタン()をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。変数を作成および使用のダイアログボックスが開き、ソースのデータタイプ(該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下ノード編集ツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。





ノードをソースとして選択すると、そのノードは**選択ソース**の下に選択されたソースとして表示されます。



複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースのデフォルト値は期待通りとは限りません。あるフォームオブジェクトのデフォルト値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。

初期値のリストから、ラジオボタンの初期値を定義する方法を選択します。オプションには、**最初の許可される値**、**データソース参照**(デフォルトとして、選択したデータソースで指定された値を使用)、**カスタム値**があります。**カスタム値**のオプションの場合、**値**のリストには現在フォームオブジェクトに表示されている許可される値が表示され、これは選択可能な選択リストとその値に依存します。データソースが、許可される値のリストを持つ埋め込みモデルの設定である場合は、これらの値も**値**リストに含まれます。例えば、選択された初期値が選択リストから削除されたために無効になった場合、初期値の**値**リストでは、**無効なデフォルト値**の後にその値が続けて記載されるテキストが示された状態となります。

選択リスト

選択リストのセクションでは、ラジオボタンに許可された値を与える**選択リストノード**を追加します。その有効な値の各々は、ラジオボタンの一つ一つに対応して表わされます。選択したデータソースが許可された値のセットを持つリストである場合、それらの値のサブセットのみがラジオボタンとして表示されます。選択された選択リスト内の他の値は全て無視されます。利用可能な**選択リストノード**は、**選択元**というリスト領域に表示されます。**選択対象追加**  のボタンをクリックすると、選択した**選択リストノード**が**選択**というリスト領域に追加挿入されます。**選択対象を削除**  のボタンをクリックすると、選択済みの**選択リストノード**が**選択**というリスト領域から削除されます。また、領域内の**選択リストノード**をダブルクリックして、それを**選択元**の領域から**選択**の領域に、あるいはその逆に移動することができます。**新規選択リスト追加**のボタン()をクリックして**選択リスト**ウィンドウを開き、新しい選択リストを定義することができます。そのウィンドウのテーブルで、**値**の列に許可されている値、**表示名**の列に対応する名前を入力します。**OK**をクリックすると、**アプリケーションビルダー**ツリーの**宣言ノード**の下**の選択リストノード**()に新しい選択リストが追加され、同時に**選択**の領域にも追加されます。

ソースセクションで、許可された値のリストをデータソースとして持つプロパティを選択すると、そのプロパティは最初に**選択**のリスト領域に置かれたノードになります。これを**選択元**のリスト領域に移動すると、許可された値のリストがクリアされます。それを再度元に戻すか、プロパティの値リストに属する値を持ったカスタム選択リストを追加することも可能です。プロパティリストと**選択リストノード**が両方とも**選択**のリスト領域にある場合、それらはマージされます。その際に同一の値があった場合には、**選択**のリスト領域内で最初にくる項目の記述が選ばれます。この方法を利用して、プロパティリスト内の項目の一つを名前変更することができます。許可された値のリストもある埋め込みモデル内の別のプロパティにソースを切り替えた場合には、以前の**プロパティ**リストノードが**選択元**の領域リストと**選択**の領域リストの両方から削除され、新規ノードが**選択**の領域リストに追加されます。

位置およびサイズ

このセクションには、親フォームのグリッドにあるラジオボタンの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、ラジオボタンの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**を選択します。
- **垂直アライメント**のリストから、**中間**、**上**、**下**を選択します。

グリッドモードでは、フォームオブジェクトが自動的に拡大されてセルの水平または垂直方向(該当する場合)の全幅までサイズ変更される、**全幅**を選択することもできます。

アラインメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アラインメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

幅フィールドと**高さフィールド**は、ラジオボタンの寸法がソフトウェアによって決められるため設定変更できません。

さらに、スケッチモードでは、**位置 x** および **位置 y** フィールドを使用して、ラジオボタンの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、ラジオボタン内のテキストと背景の外観を制御できます。

テキスト色のリストから、**フォームノード**の設定からテキストの色を継承する**継承**(デフォルト)を選択するか、**黒**などの定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

背景色のリストから、ラジオボタン内の背景として使用する色を選択します。**透明**(デフォルト)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。



ラジオボタン内のテキストのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォント**リストから、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体(イタリック体)を使用するには**イタリック**のチェックボックスを選択します。あるいは、そのテキストに下線を付けるには**アンダーライン**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにラジオボタンの初期状態を制御できます。デフォルトでは、ラジオボタンは可視表示されて有効になっています。ラジオボタンが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。


イベント

特定のタイプのフォームオブジェクトでは、データが入力されたといったようなイベントが発生した場合に実行するメソッドを指定することができます。**データ変更時**のリストには、**なし**(デフォルト)と使用可能なメソッドが含まれています。イベントのローカルメソッドを追加するには、**データ変更時**のリストの右側にある**ローカルメソッド作成**のボタン(+)をクリックします。すると、**データ変更時**リストで選択したメソッドが**ローカルメソッド**に変更され、ローカルメソッドのエディターウィンドウが開かれます。そこで、ローカルメソッドの内容を定義することができます。そのローカルメソッドをメソッドエディターウィンドウで開くには、

ソースへのボタン()をクリックします。ローカルメソッドを削除するには、**ローカルメソッドを削除**のボタン()をクリックします。また、ラジオボタンを右クリックしてローカルメソッドを作成するか、(**メソッド編集**または**ローカルメソッド編集**を選択して)コマンドに関連付けられたメソッドを開きます。ラジオボタンを Ctrl + Alt を押しながらかlickした場合、

- **データ変更時**リストがメソッドに設定されている場合には、メソッドエディターウィンドウが開かれます。
- **データ変更時**リストが**なし**に設定されている場合には、(必要に応じて)ローカルメソッドを作成してそのリストに**ローカルメソッド**が設定され、ローカルメソッドのエディターウィンドウが開かれます。

選択入力

選択入力()は、アプリケーション内のいくつかのジオメトリックエンティティの選択入力のためのフォームオブジェクトです。選択入力オブジェクトは、COMSOL マルチフィジックスモデルでの選択設定と似ています。ユーザーは選択をアクティブにすることができます。例えば、選択入力エンティティのリストに選択範囲をコピーして貼り付けることができます。






また、選択範囲をグラフィックスオブジェクトのソースとして追加して、選択入力コンポーネントなしでそのグラフィックスオブジェクト内で直接ジオメトリックエンティティを選択できるようにすることもできます。その場合、グラフィックスオブジェクトは入力のみを使用する必要があり、ユーザーは選択をアクティブにする必要はありません。

選択入力オブジェクトの名前は、**名前**フィールドに入力します。

設定ウィンドウには、次のセクションがあります。

ソース




このセクションでは、ソースとして使用する選択を定義します。このセクションには、**モデルビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類 of データを表すか、または子ノードを持ちます。選択入力では、ツリーにモデルからの**明示的な**選択ノードが含まれており、それを選択のソースとして選択することができます。この選択は、アプリケーションが使用するために利用可能であり、ユーザーによって更新された選択も含まれます。データを表すノードを選択すると、ツリーの下にある**ソースとして使用**ツールバーボタン()が有効になります。それをクリックするか、代わりに Enter キーを押すか、ダブルクリックするか、ノードを右クリックして**ソースとして使用**を選択して、選択ソースとして追加します。**ソース**セクションのヘッダーの**作成**ボタン()をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。**変数を作成および使用**のダイアログボックスが開き、ソースのデータタイプ(該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下**ノード編集**ツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。

ノードをソースとして選択すると、そのノードは**選択ソース**の下に選択されたソースとして表示されます。



複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースのデフォルト値は期待通りとは限りません。あるフォームオブジェクトのデフォルト値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。

アクティブ時のグラフィックス

アプリケーションのユーザーがアクティブ化スイッチを ON に設定すると、選択したジオメトリックエンティティがハイライト表示されているグラフィックスオブジェクトに選択入力項目を接続できます。ツリーからグラフィックスオブジェクトを選択し、**グラフィックス使用**のボタンをクリックして**選択グラフィックス**に追加します。また、ユーザーは、グラフィックスオブジェクトで直接選択することもできます。グラフィックスオブジェクトが標準のプロットツールバーを含むように設定されている場合、そのツールバーには**選択対象をズーム**()、**ボックス選択**(3D 時 )、および**ボックス選択解除**()といったボタンも含まれています。複数の選択入力オブジェクトが同じグラフィックスオブジェクトに接続されている場合、一度に選択入力項目の一つだけしかアクティブになりません。



ユーザーは、アクティブ化スイッチを OFF に設定することはできません。別の選択入力 that アクティブになっている場合、または関連するグラフィックスオブジェクトが選択設定以外のソースを取得した場合に、スイッチはオフになります。

位置およびサイズ

このセクションには、親フォームのグリッドにある選択入力オブジェクトの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、選択入力フォームオブジェクトの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

選択入力の幅を指定することもできます。**幅**フィールドに幅(ポイント単位で)を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ 0 を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。**高さ**フィールドは、選択入力のフィールドの高さがソフトウェアによって決められるため設定変更できません。

さらに、スケッチモードでは、**位置 x** および **位置 y** フィールドを使用して、選択入力の絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。



ホバー選択は、ウェブクライアントではサポートされていません。代わりに、マウスをクリックしたときに、何が選択されるかをハイライトで示す**事前選択**(preselection)が表示されます。この事前選択は、そこからマウスを動かすと消えます。下層を構成しているオブジェクトを選択するには、マウスホイールを使用します。ホバー選択の場合とは対照的に、最上位層も第 1 層として含まれます。最も近いオブジェクトを簡単に選択できるように、事前選択がない場合はマウスポインタの下の部分での最上位層のオブジェクトが選択されます。これは、クリックする前にマウスを移動してしまうと、事前選択以外のオブジェクトが選択されてしまう可能性があることを意味します。

外観

このセクションでは、選択入力内のテキストと背景の外観を制御できます。

テキスト色のリストから、フォームノードの設定からテキストの色を継承する**継承**(デフォルト)を選択するか、**黒**などの定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。


背景色のリストから、選択入力内の背景として使用する色を選択します。**透明**(デフォルト)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

選択入力内のテキストのフォントとフォントサイズには、フォームノードのフォント設定がデフォルトとして使用されます。フォントリストから、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体(イタリック体)を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときに選択入力の初期状態を制御できます。デフォルトでは、選択入力は可視表示されて有効になっています。選択入力为非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

テキスト


テキスト()フォームオブジェクトは、デフォルトのテキストを含めたテキストフィールドとして事前に定義されたフォームです。情報を提供するスタティックなテキストとして利用するか、または、例えばユーザーがメモやコメントを追加できるように編集可能なテキストとして利用することができます。テキストオブジェクトの名前は、**名前**フィールドに入力します。

ユーザーがテキストを編集して追加できるようにするには、**編集可能**のチェックボックスを選択します。デフォルトでは、テキストはスタティックです。

デフォルトでは、**テキストを折畳み**のチェックボックスが選択されています。これをクリックしてオフにすると、テキストの折り畳み機能が無効になります。テキストが定義された寸法に収まらない場合には、スクロールバーが表示されます。







設定ウィンドウには、以下のセクションがあります。

ソース

このセクションでは、テキストオブジェクトのソースを定義します。このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類のデータを表すか、または子ノードを持ちます。使用可能なデータノードのリストを拡張するには、**ソース**セクションのヘッダーで**モデルビルダー** **ヘスイッチしデータアクセスをアクティブ**のボタン()をクリックし、モデルビルダーに切り替わってから、アクセスしたいデータのある**モデルビルダー**内のノードを選択します。このボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることが可能な設定箇所横に**データソース選択**チェックボックス(緑色の四角形)が表示されます。




そのデータをテキストオブジェクトの使用可能なソースノードとして含めるには、チェックボックスをクリックして選択します。

テキストが読み取り専用になるように**編集可能**のチェックボックスがクリアされている場合は、メインの**モデルノード**および各**スタディノード**の下にある以下の**情報**ノードのいずれかをソースとして使用することができます。

- **モデル>情報**()の下の**予測計算時間**ノード()： 予測計算時間は、アプリケーション開発者が **root** ノードの**設定**ウィンドウの**予想**フィールドに入力できる値です。
- **モデル>情報**()の下の**前回の計算時間**ノード()： このノードには、最後に計算されたスタディの最後に測定された計算時間が表示されます。
- **各モデル>スタディ>情報**()の下の**前回の計算時間**ノード()： このノードには、スタディの最後に測定された計算時間が表示されます。

最初にアプリケーションを起動すると、最後に測定された時間がリセットされるため、前回の計算時間は表示されません。

(注) 原文では「**Not available yet** が表示される」となっていますが、弊社では確認できていません。

データを表すノードを選択すると、ツリーの下にあるソースとして使用ツールバーボタン()が有効になります。それをクリックするか、代わりに Enter キーを押すか、ダブルクリックするか、ノードを右クリックしてソースとして使用を選択して、選択ソースとして追加します。また、ソースセクションのヘッダーの作成ボタン()をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。変数を作成および使用のダイアログボックスが開き、ソースのデータタイプ(該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下ノード編集ツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。

ノードをソースとして選択すると、そのノードは選択ソースの下に選択されたソースとして表示されます。



複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースのデフォルト値は期待通りとは限りません。あるフォームオブジェクトのデフォルト値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。

初期値のリストからデータソース参照(デフォルト)を選択すると、ソースからのテキストが使用されます。代わりに、カスタム値を選択すると、その下にある値のテキストフィールドに入力した内容をテキストに表示させることができます。

位置およびサイズ

このセクションには、親フォームのグリッドにあるテキストオブジェクトの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、テキストオブジェクトの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、フォームツールバーのスケッチセクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

テキスト入力の幅と高さを指定することもできます。幅フィールドに幅(ポイント単位で)を、高さフィールドに高さ(ポイント単位で)を入力します。水平アライメントのリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ0を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。垂直アライメントのリストから**全幅**を選択した場合は、**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します(通常は最小サイズ0を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、テキスト入力の絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。

- **親フォーム参照** (デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、テキストと背景の外観を制御できます。

テキスト色のリストから、このテキストに使用する色を選択します。**継承** (デフォルト: フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

背景色のリストから、このテキストの背景として使用する色を選択します。**透明** (デフォルト)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

テキストアライメントのリストから、**左** (デフォルト)、**中心**、または**右**のテキストの配置を選択します。

このテキストのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォント**リストから、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ (ポイント単位) を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体 (イタリック体) を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにテキストオブジェクトの初期状態を制御できます。デフォルトでは、テキストオブジェクトは可視表示されて有効になっています。テキストオブジェクトが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

イベント

テキストが変更されたときにアプリケーションによって実行されるコードメソッドを追加することができます。このイベントは、新しいデータ値がデータソースに格納された後に発生されます。**データ変更時**のリストには、**なし** (デフォルト) と使用可能なメソッドが含まれています。イベントのローカルメソッドを追加するには、**データ変更時**のリストの右側にある**ローカルメソッド作成**のボタン (+) をクリックします。すると、**データ変更時**リストで選択したメソッドが**ローカルメソッド**に変更されます。(このように最初にローカルメソッドを作成すると、エディターウィンドウが自動的に開かれます。) 選択したメソッドをエディターウィンドウで開くには、**ソースへ**のボタン (📄) をクリックします。空の **onDataChange** エディターウィンドウが開かれ、そこでローカルメソッドを定義することができます。ローカルメソッドを削除するには、**ローカルメソッドを削除**のボタン (✖) をクリックします。

また、テキストオブジェクトを右クリックしてローカルメソッドを作成するか、(**メソッド編集**または**ローカルメソッド編集**を選択して) コマンドに関連付けられたメソッドを開きます。

テキストオブジェクト上で Ctrl + Alt を押しながらクリックした場合、


- **データ変更時**リストがメソッドに設定されている場合には、メソッドエディターウィンドウが開かれます。
- **データ変更時**リストが**なし**に設定されている場合には、(必要に応じて) ローカルメソッドを作成してそのリストに**ローカルメソッド**が設定され、ローカルメソッドのエディターウィンドウが開かれます。




リストボックス

リストボックス (☐) フォームオブジェクトは、リストボックスを表します。一つのリストをソースとして使用する場合、Shift キーを押しながらクリックするか Ctrl キーを押しながらクリックして、そのリスト内の複数の項目を選択できます。他のソースについては、リストから一つの値しか選択できません。リストボックスオブジェクトの名前は、**名前**フィールドに入力します。

設定ウィンドウには、以下のセクションがあります。

ソース

このセクションでは、リストボックスのデータソースを定義します。このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類のデータを表すか、または子ノードを持ちます。リストボックスでは、モデルの変数とパラメータ、および宣言の下で追加された配列 1D などの変数をデータソースとして使用でき、そこには**単位セット**も含まれます。使用可能なデータノードのリストを拡張するには、**ソースセクション**のヘッダーで**モデルビルダーへスイッチしデータアクセスをアクティブ**のボタン()をクリックして、モデルビルダーに切り替わってから、アクセスしたいデータのある**モデルビルダー**ブランチ内のノードを選択します。このボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることが可能な設定箇所横に**データソース**選択チェックボックス(緑色の四角形)が表示されます。そのチェックボックスをクリックして選択すると、リストボックスに使用可能なソースノードとしてそのデータが挿入されます。

データを表すノードを選択すると、ツリーの下にある**ソースとして使用**ツールバーボタン()が有効になります。それをクリックするか、代わりに Enter キーを押すか、ダブルクリックするか、ノードを右クリックして**ソースとして使用**を選択して、選択ソースとして追加します。また、**ソース**セクションのヘッダーの**作成**ボタン()をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。**変数を作成および使用**のダイアログボックスが開き、ソースのデータタイプ(該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下**ノード編集**ツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。

ノードをソースとして選択すると、そのノードは**選択ソース**の下に選択されたソースとして表示されます。







複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースのデフォルト値は期待通りとは限りません。あるフォームオブジェクトのデフォルト値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。

初期値のリストから、リストボックスの初期値を定義する方法を選択します。オプションには、**最初の許可される値**、**データソース参照**(デフォルトとして、選択したデータソースで指定された値を使用)、**カスタム値**があります。**カスタム値**のオプションの場合、**値**のリストには現在フォームオブジェクトに表示されている許可される値が表示され、これは選択可能な選択リストとその値に依存します。データソースが、許可される値のリストを持つ埋め込みモデルの設定である場合は、これらの値も**値**リストに含まれます。例えば、選択された初期値が選択リストから削除されたために無効になった場合、初期値の**値**リストでは、**無効なデフォルト値**の後にその値が続けて記載される**テキスト**が示された状態となります。

データソースが文字列配列の場合、選択ソースのデフォルトとして空の配列を設定する**空配列**オプションが**初期値**リストに含まれます。値の選択方法については、更に追加される選択肢があります。**次で値を選択**のリストから**ダイアログ**を選択して、ダイアログボックスを使用してリストボックスを表示することができます。この**ダイアログタイトル**は、その下のフィールドに入力します。あるいは、**リストボックス**を選択して複数選択リストボックスを使用することもできます。**ダイアログ**オプションの場合は、下にボタンがあるリストが使用され、ユーザーがリスト内で項目の追加、削除、および移動に使用することができます。これは、(標準の複数選択リストボックスと比較して、)項目が多数あるリストでどの項目が選択されているかをよりよく管理したいような場合に適しています。

選択リスト

選択リストのセクションでは、リストボックスに許可された値を与える**選択リストノード**を追加します。選択したデータソースが許可された値のセットを持つリストである場合、それらの値のサブセットのみがリストボックスの許可された値として表示されます。選択された選択リスト内の他の値は全て無視されます。利用可能な**選択リストノード**は、**選択元**というリスト領域に表示されます。**選択対象追加**  のボタンをクリックすると、選択した**選択リストノード**が**選択**というリスト領域に追加挿入されます。**選択対象を削除**  のボタンをクリックすると、選択済みの**選択リストノード**が**選択**というリスト領域から削除されます。また、領域内の**選択リストノード**をダブルクリックして、それを**選択元**の領域から**選択**の領域に、あるいはその逆に移動することができます。**新規選択リスト追加**のボタン()をクリックして**選択リスト**ウィンドウを開き、新しい選択リストを定義することができます。そのウィンドウのテーブルで、**値**の列に許可されている値、**表示名**の列に対応する名前を入力します。**OK**をクリック

すると、アプリケーションビルダーツリーの宣言ノードの下の選択リストノード()に新しい選択リストが追加され、同時に選択の領域にも追加されます。

ソースセクションで、許可された値のリストをデータソースとして持つプロパティを選択すると、そのプロパティは最初に選択のリスト領域に置かれたノードになります。これを**選択元**のリスト領域に移動すると、許可された値のリストがクリアされます。それを再度元に戻すか、プロパティの値リストに属する値を持ったカスタム選択リストを追加することも可能です。プロパティリストと**選択リストノード**が両方とも**選択**のリスト領域にある場合、それらはマージされます。その際に同一の値があった場合には、**選択**のリスト領域内で最初にくる項目の記述が選ばれます。この方法を利用して、プロパティリスト内の項目の一つを名前変更することができます。許可された値のリストもある埋め込みモデル内の別のプロパティにソースを切り替えた場合には、以前のプロパティリストノードが**選択元**の領域リストと**選択**の領域リストの両方から削除され、新規ノードが**選択**の領域リストに追加されます。

位置およびサイズ

このセクションには、親フォームのグリッドにあるリストボックスの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、リストボックスの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチセクション**の中の**アレンジメニュー**でアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

リストボックスの幅と高さを指定することもできます。**幅フィールド**に幅(ポイント単位で)を、**高さフィールド**に高さ(ポイント単位で)を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ0を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。**垂直アライメント**のリストから**全幅**を選択した場合は、**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します(通常は最小サイズ0を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、リストボックスの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、リストボックス内のテキストの外観を制御できます。

テキスト色のリストから、テキストに使用する色を選択します。継承(デフォルト; フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、カスタムを選択して、カラーパレットからカスタムカラーを選択することもできます。

リストボックス内のテキストのフォントとフォントサイズには、フォームノードのフォント設定がデフォルトとして使用されます。フォントリストから、デフォルトフォントまたはリストにある使用可能なフォントを選択してください。必要に応じて、フォントサイズのコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントのデフォルトサイズが使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体(イタリック体)を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにリストボックスの初期状態を制御できます。デフォルトでは、リストボックスは可視表示されて有効になっています。リストボックスが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

イベント

リストボックス内のデータが変更されたときにアプリケーションによって実行されるコードメソッドを追加することができます。このイベントは、新しいデータ値がデータソースに格納された後に発生されます。データ変更時のリストには、なし(デフォルト)と使用可能なメソッドが含まれています。イベントのローカルメソッドを追加するには、**ローカルメソッド作成**のボタン(+)をクリックします。すると、データ変更時リストで選択したメソッドが**ローカルメソッド**に変更されます。選択したメソッドをエディターウィンドウで開くには、**ソースへ**のボタン(📄)をクリックします。空の `onDataChange` エディターウィンドウが開かれ、そこでローカルメソッドを定義することができます。ローカルメソッドを削除するには、**ローカルメソッドを削除**のボタン(✖)をクリックします。また、リストボックスを右クリックしてローカルメソッドを作成するか、(メソッド編集またはローカルメソッド編集を選択して)コマンドに関連付けられたメソッドを開きます。

リストボックス上で **Ctrl + Alt** を押しながらクリックした場合、

- **データ変更時**リストがメソッドに設定されている場合には、メソッドエディターウィンドウが開かれます。
- **データ変更時**リストがなしに設定されている場合には、(必要に応じて)ローカルメソッドを作成してそのリストに**ローカルメソッド**が設定され、ローカルメソッドのエディターウィンドウが開かれます。

テーブル

テーブル(📊)フォームオブジェクトは、行と列を持つテーブルを表します。テーブルオブジェクトの名前は、名前フィールドに入力します。テーブルの列には、ヘッダーが含まれます。ヘッダーを削除するには、**ヘッダー表示**のチェックボックスをオフにします。




COMSOL デスクトップでのグローバルパラメータや変数を含むテーブルの場合と同じように、ユーザーが新しい値を入力したときにテーブルに新しい行を追加させるには、**新しい行の自動追加**のチェックボックスを選択します。

テーブルの行をソート可能にするには、**ソート可能**のチェックボックスをオンにします。これにより、ユーザーはテーブルの列ヘッダーをクリックして行をソートすることができます。列ヘッダーをクリックするたびに、行のソートが昇順、降順、およびソートなし(テーブルの元の行の順序)の順に実行されます。




また、設定ウィンドウには、以下のセクションがあります。

ソース

このセクションでは、テーブルデータのソースを定義します。このセクションには、**アプリケーションビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類 of データを表すか、または子ノードを持ちます。テーブルでは、**宣言**の下に追加した配列をテーブルのデータソースとして使用できます。データを表すノードを選択すると、ツリー

の下にある**テーブルに追加**ツールバーボタン()が有効になります。それをクリックするか、代わりに Enter キーを押すか、ダブルクリックするか、ノードを右クリックして**テーブルに追加**を選択することもできます。選択したデータソースがテーブルに表示されるデータのソースになり、そのデータソースの初期値がテーブル入力オブジェクトに表示されます。また、**ソース**セクションのヘッダーの**作成**ボタン()をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。**変数を作成および使用**のダイアログボックスが開き、ソースのデータタイプ(該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下の**ノード編集**ツールバーボタン()をクリックすると、対応するノードに移動します。

下のテーブルでは、**ヘッダ**の下のヘッダーテキスト、**幅**の下の幅(ピクセル単位)、および**編集**の下のテーブルデータが編集可能(デフォルト)か否かの各列の行を編集することができます。**自動拡大縮小**のチェックボックスを選択したテーブルの列は、水平方向に自動拡大縮小します。テーブルの列を自動拡大縮小できるようにするには、テーブルを追加するフォームでグリッドモードを使用する必要があります。また、**位置およびサイズ**(下記参照)の**水平アライメント**のリストから**全幅**を選択する必要があります。**アライメント**の列のリストを使用して、列のデータ体裁の配置を調整することができます。**左**(デフォルト)、**中心**、**右**を選択します。右端の列の**データソース**には、各列のデータのソースが表示されます。

上へ移動および**下へ移動**ボタン( および )をクリックして、列の順序を移動して並べ替えます。選択した列を削除するには、**削除**ボタン()をクリックします。

データソースに関する操作

テーブルのデータソースを操作する場合、例えば、メソッドを使用してソースをテーブルに変更するときや、テーブルの一部である特定のデータソースに対して set コマンドを実行するときにも、以下の規則が適用されます。以下の規則では、テーブルの**最初のデータソース**がテーブルの行数を制御するソースです。テーブルで使用される他の全てのソースには、デフォルトの要素値が埋め込まれるか、あるいは、最初のデータソースの長さに合わせてクロップされます。最初のデータソースとして設定したツリーノードには、鍵のシンボルが表示された状態となります。

- データをクリアするには、最初のデータソースをクリアしさえすればよい。
- 一つの行を追加するには、最初のデータソースに新しい行を与えさえすればよい。
- 最後の行を削除するには、最初のデータソースの最後の行を削除しさえすればよい。
- 一つの行を挿入するには、最初のデータソースに行を挿入し、それを他のプロパティに対して挿入する必要があります。
- 一つの行を削除するには、最初に他の全てのソースに対してその行を削除し、最後に最初のデータソースに対してその行を削除する必要があります。


ツールバー




このセクションでは、ツールバーのテーブルに項目を追加し、以下のテーブルオプションを有効にすることができます。

位置のリストから、ツールバーを配置するテーブルとの相対的な位置を、**下**(デフォルト)、**上**、**左**、または**右**から選択します。




二つのアイコンサイズから選択することができます。**アイコンサイズ**リストから、**小**(デフォルト)または**大**を選択します。

下側のテーブルでは、一つまたは複数のボタンを追加してツールバーのテーブルを作成できます。

ツールバーアイテム追加のボタン()をクリックすると、**ツールバー項目**ダイアログボックスが開きます。このダイアログボックスでは、次のテーブルコマンドをツールバーボタンとして一つ以上選択して追加できます。

移動()の下では、**上へ移動**()および**下へ移動**()ボタンをダブルクリックして、行を移動させるために対応するボタンを追加します。

修正()の下では、

- **追加**ボタン()をダブルクリックして、行を追加するボタンを追加します。
- **削除**ボタン()をダブルクリックして、選択した行を削除するボタンを追加します。
- **テーブル消去**ボタン()をダブルクリックして、テーブル全体の全てのコンテンツをクリアするボタンを追加します。

ファイル(📁)の下では、ファイルからロード(📄)、テーブル消去、ファイルからのロード(📄)、ファイルに保存(💾)のボタンをダブルクリックして、それぞれについて、ファイルからテーブルデータを読み込むため、最初にテーブルをクリアしてから新しいコンテンツをファイルから読み込むため、およびファイルにテーブルデータを保存するために対応するボタンを追加します。ユーザーがテーブルデータを読み込んで保存することができるファイルタイプは、テキストファイル(.txt)、CSV ファイル(.csv)、データファイル(.dat)、およびライセンスに LiveLink™ for Excel®が含まれている場合には Microsoft Excel® ファイル(.xlsx)です。使用できるデータ区切り文字は、CSV ファイルの場合はカンマ、セミコロン、およびタブです。その他の Excel® 以外のファイルの場合はスペースとタブです。

これらの各グループの下にあるボタンをいっぺんに全部追加するには、**移動**、**修正**、および**ファイル**ボタンをダブルクリックします。または、別の方法として、ダイアログボックスの**追加**ボタンと**削除**ボタンをクリックして、それぞれ選択したボタンを追加および削除することができます。**カスタムアイテム**のボタンをクリックすると、**カスタムツールバーアイテム編集**ダイアログボックスが開かれます(**カスタムツールバーアイテム編集**ダイアログボックスを参照)。

OK をクリックして**ツールバー項目**ダイアログボックスを閉じると、選択したコマンドがツールバーのテーブルにボタンとして追加されます。再び**ツールバー項目**ダイアログボックスを開くと、左側の領域にはまだ追加していないテーブルコマンドのみが含まれた状態で表示されます。

セパレーター追加のボタン(🔗)をクリックすると、ツールバーのボタンのグループ間にセパレータを追加することができます。

カスタムツールバーアイテム編集ダイアログボックスでカスタムツールバーボタンの外観や動作を変更するには、テーブル内のボタンを選択して**編集**ボタン(🔧)をクリックします。**上へ移動**および**下へ移動**ボタン(↑ および ↓)をクリックして、ツールバーのボタンの順序を移動して並べ替えることができます。**削除**ボタン(✖)をクリックすると、選択したボタンが削除されます。

テーブルには、追加された各項目の行が含まれており、**名前**、**アイコン**、**テキスト**、および**ツールチップ**の各列にそれぞれ、名前、アイコン、テキスト、およびツールチップが表示されます。

位置およびサイズ

このセクションには、親フォームのグリッドにあるテーブルの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、テーブルの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォーム**ツールバーの**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

テーブルの高さを指定することもできます。**高さ**フィールドに高さ(ポイント単位で)を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します(通常は最小サイズ 0 を意味します)。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。**幅**フィールドは、テーブルの幅がソフトウェアによって決められるため設定変更できません。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、テーブルの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**(デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、テーブル内のテキストの外観を制御できます。

テキスト色のリストから、テキストに使用する色を選択します。**継承**(デフォルト; フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

テーブル内のテキストのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォント**リストから、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体(イタリック体)を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにテーブルの初期状態を制御できます。デフォルトでは、テーブルは可視表示されて有効になっています。テーブルが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

イベント

特定のタイプのフォームオブジェクトでは、データが入力されたといったようなイベントが発生した場合に実行するメソッドを指定することができます。**データ変更時**のリストには、**なし**(デフォルト)と使用可能なメソッドが含まれています。イベントのローカルメソッドを追加するには、**データ変更時**のリストの右側にある**ローカルメソッド作成**のボタン(+)をクリックします。すると、**データ変更時**リストで選択したメソッドが**ローカルメソッド**に変更され、新規メソッドのためのエディターウィンドウが開かれます。選択したメソッドをメソッドエディターウィンドウで開くには、**ソース**へのボタン(📄)をクリックします。空の **onDataChange** エディターウィンドウが開かれ、そこでローカルメソッドを定義することができます。ローカルメソッドを削除するには、**ローカルメソッドを削除**のボタン(✖)をクリックします。また、テーブルオブジェクトを右クリックしてローカルメソッドを作成するか、(**メソッド編集**または**ローカルメソッド編集**を選択して)コマンドに関連付けられたメソッドを開きます。








テーブルオブジェクト上で Ctrl + Alt を押しながらクリックした場合、

- **データ変更時**リストがメソッドに設定されている場合には、メソッドエディターウィンドウが開かれます。
- **データ変更時**リストがなしに設定されている場合には、(必要に応じて)ローカルメソッドを作成してそのリストに**ローカルメソッド**が設定され、ローカルメソッドのエディターウィンドウが開かれます。

データが変更されると、新しいデータ値がデータソースに格納された後にイベントが発生されます。

カスタムツールバーアイテム編集ダイアログボックス

カスタムボタンまたはトグルボタンをツールバーオブジェクト、テーブル、またはグラフィックスオブジェクトに追加するには、いくつかの方法があります。ボタンアイテムまたはトグルボタンアイテムを作成し、**カスタムツールバーアイテムを編集**のダイアログボックスを開いて定義するには、次のいずれかの手順を使用します。**カスタムツールバーアイテムを編集**ダイアログボックスには、カスタムボタンまたは関連するコマンドやメソッドを実行するトグルボタンを作成するための設定とツールがあります。



- ツールバーオブジェクトの場合は、**ツールバー項目**セクションの項目のテーブルの下にある**項目追加** () または **トグルアイテム追加** () のボタンをクリックするか、フォームウィンドウ上のツールバーを右クリックして**項目追加** () または **トグルアイテム追加** () のボタンを選択します。また、**ツールバーオブジェクトの設定**ウィンドウで、カスタムツールバーボタンまたはトグルボタンを右クリックして**編集**を選択することによっても開くことができます。
- テーブルオブジェクトの場合は、**設定**ウィンドウの**ツールバー**セクションの**ツールバーアイテム追加**のボタン () をクリックします。次に、**ツールバー項目**ダイアログボックスで、**カスタムアイテム**ボタンまたは**カスタムトグルアイテム**ボタンをクリックして、**カスタムツールバーアイテムを編集**ダイアログボックスを開きます。**テーブルオブジェクトの設定**ウィンドウで、カスタムツールバーボタンまたはトグルボタンを右クリックして**編集**を選択することによっても開くことができます。
- **グラフィックスオブジェクト**の場合は、**設定**ウィンドウの**ツールバー**セクションの下部にある**項目追加** () または **トグルアイテム追加** () のボタンをクリックします。また、**グラフィックスオブジェクトの設定**ウィンドウで、カスタムツールバーボタンまたはトグルボタンを右クリックして**編集**を選択することによっても開くことができます。

カスタムツールバーアイテムを編集ダイアログボックスには、次のページがあります。

一般

名前フィールドに、参照するために使用するツールバーボタンの名前を入力します。

テキストフィールドに、ボタンにラベルとして表示されるテキストを入力します。

アイコンリストから、アイコンなしの場合は**なし**を選択するか、既存の画像ファイルからアイコンを選択します。**ライブラリ**に対する**画像を追加してここで使用**のボタン () をクリックすると、アイコンとして使用する画像をブラウズして選択することができます。**エクスポート**ボタン () をクリックすると、アイコンを画像ファイルとして保存できます。アイコンとテキストラベルを使用すると、ツールバーの項目にアイコンとラベルの両方が表示されます。

ツールチップフィールドには、ボタンまたはトグルボタンのツールチップとして表示される説明テキストを入力します。**ツールバー項目**ダイアログボックスでは、テキストラベルが空の場合、追加されたボタンがツールチップによって表示されます。

キーボードショートカットを追加するには、**キーボードショートカット**フィールドをアクティブにしてから、キーボードでそのキーボードショートカットを実際に入力します。

キーボードショートカットでは、普通の文字だけでなく、CTRL + SHIFT + D などの修飾キーを使用する必要があります。ショートカットには、Ctrl キー (CTRL)、Alt キー (ALT)、Shift キー (SHIFT) を使用できます。Ctrl キーは OS X のコマンドとして解釈されることに注意してください。ショートカットで次のキーを使用しないでください。

- Backspace: ショートカットを消去するために使用できるため
- Delete: ショートカットを消去するために使用できるため
- Escape
- Alt 単独使用: ファイルメニューのショートカットとの競合を避けるため





他のキーボードショートカットを無効にすることも可能ですので、使用するショートカットキーの組み合わせを選択する際には注意してください。

ソース



ソースページは、ツールバーのトグルボタンでのみ使用できます。

ここでは、ツールバーのトグルボタンの状態のソースを指定します。ソースは、**宣言**の下に作成される文字列やブーリアン変数にすることができ、それをツリーから選択して**ソースとして使用**ボタン()をクリックします。または、Enter キーを押すか、ダブルクリックするか、ノードを右クリックして**ソースとして使用**を選択して、選択ソースとして追加します。**新規変数を作成しソースとして使用**のボタン()をクリックすることもできます。ソースを指定したら、次の設定を使用して初期状態を定義します。

初期値リストから、**データソース参照**(デフォルト)を選択してデータソースの初期値を使用するか、**カスタム値**を選択して**初期状態**リストを使用して初期状態を定義します。**選択**(デフォルト)または**クリア**から選択します(選択されたものがオンで、クリアされているものはオフです)。

実行コマンド選択

このセクションには、**アプリケーションビルダー**ウィンドウのツリーのフィルター処理されたツリーが含まれています。ノードは、コマンドをサポートしているか、または子ノードを持っています。一つまたは複数のコマンドをサポートするノードを選択すると、ツリーの下ツールバーでそれに対応するボタンが有効になります。また、ノードを右クリックして、そのノードの使用可能なコマンドのリストを取得することもできます。選択したノードのコマンドをクリックする(または Enter キーを押すか、ダブルクリックして**実行**、**プロット**、**値設定**などのデフォルトコマンドでコマンドを追加する)と、ツリーの下テーブルの最後の行にコマンドとノードが表示されます。このテーブルには、実行する全てのノードが含まれます。テーブルの下にあるツールバーを使用して、コマンドを削除したり移動したりすることができます。

モデルブランチでは、**パラメーターノード**の下パラメーターなど、何らかの種類データ値を表すノードの全てが、**値設定**のコマンドをサポートしています。**値設定**コマンドをテーブルに追加すると、3 番目の列の**引数**が有効になります。この列に、設定する値を入力します。配列を表すデータの場合は、中括弧とコンマを使用して配列要素を入力します。例えば、{1, 2, 3}を入力して、値が 1,2,3 の 3 要素配列を設定します。配列と行列を入力する方法の詳細については、**配列 1D 文字列ノード**を参照してください。**補関数ノード**の下にある**ファイル名ノード**のように、ファイルのインポートを表すノードの場合は、**インポートファイル**のコマンドを使用できます。








ツリーには、**モデルブランチ**に加えて、**アプリケーションツリー**からのいくつかのブランチが含まれています。

- **フォームブランチ**: **フォームノード**は、フォームをアプリケーションのメインフォーム(つまり、アプリケーションウィンドウのコンテンツはこのフォームになります)として設定する**表示**コマンドと、フォームを別のダイアログウィンドウとして表示する**ダイアログとして表示**をサポートしています。
- **GUI コマンドブランチ**: このブランチの下にあるコマンドは、三つのサブカテゴリにグループ化されています。
 - **ファイルコマンド**: これには、**アプリケーションを保存**(現在の名前でアプリケーションを保存する); **名前を付けてアプリケーションを保存**(ユーザーがアプリケーションを適切な場所に保存できるようにするファイルブラウザーダイアログを開く); **サーバー上でアプリケーションを保存**; **サーバー上でアプリケーションを別名で保存**; **ファイルを開く**(引数列で有効な URI パスを使用して指定されたアプリケーションファイルリソースを開く); **名前を付けてファイルを保存**(同様に、ユーザーが引数列で指定した名前でファイルを保存できるようにする); および**アプリケーションを終了**(実行中のアプリケーションを閉じる)が含まれています。アプリケーションが COMSOL サーバー上で実行されている場合、**サーバー上でアプリケーションを保存**と**サーバー上でアプリケーションを別名で保存**のコマンドは、COMSOL サーバーアプリケーションライブラリに新しいアプリケーションとして現在の状態を保存します。
 - **グラフィックスコマンド**: 画面にわたって**ズーム**、**現在のビューをリセット**、**シーンライト**、**透明度**、**印刷**、**全選択**、および**選択対象消去**のコマンドがあります。全てのグラフィックスコマンドで、コマンドに引数として適用するグラフィックスオブジェクトの名前を追加します。
 - **モデルコマンド**: ここでは、**全解消去**と**全メッシュ消去**のコマンドが利用できます。

上記のいずれかのノードをダブルクリックまたは右クリックして、**実行コマンド**を追加します。


- **宣言ブランチ**: このブランチには、**アプリケーションビルダー**ウィンドウの**宣言ブランチ**の下に追加した変数宣言がタイプ別にグループ化されています。パラメーターと同様に、**値設定コマンド**がサポートされています。
- **メソッドブランチ**: **メソッドノード**は、**実行コマンド**をサポートしています。
- **ライブラリブランチ**: **サウンド**の下で、コマンドシーケンスで再生するサウンドファイルを選択できます。

ツリーの下にあるボタンの一つをクリックすると、現在選択されているコマンドが下のテーブルの**コマンド列**に表示されます。また、**アイコン列**と**引数列**もあり、後者ではコマンドで使用される任意の引数を入力できます。

コマンドのリストの下にある**ローカルメソッド作成**ボタン()をクリックすると、テーブル内のコマンドのリスト全体が同等のコードを含む新しいローカルメソッドに変換され、そのローカルの onClick メソッドのエディターウィンドウが開かれます。ローカルメソッドを作成していないときは、**新しいメソッドに変換**ボタン()をクリックして、テーブル内のコマンドのリスト全体を同等のコードを含む新しい**メソッドノード**に変換します。この操作の後、コマンドのリストには、作成された**メソッドノード**の1回の**実行**操作だけが設定された形になります。**コマンド**の下で一つのメソッドを選択するか、リスト内に一つのメソッドしかない場合、**メソッド**へのボタン()をクリックしてそのメソッドのエディターウィンドウに移動することができます。**引数編集**のボタン()の詳細については、**宣言とコマンドシーケンスでの初期値と引数の編集**を参照してください。**上へ移動**()、**下へ移動**()、**削除**()のツールバーボタンを使用して、リストからコマンドを整理したり削除することができます。(また、ローカルメソッドを削除することもできます。)

OK をクリックして**カスタムツールバーアイテム編集**のダイアログボックスを閉じ、ツールバーにボタンを追加します。

スライダー

スライダー()は、スライダーコントロールを使用して数値入力を選択するためのフォームオブジェクトです。スライダーオブジェクトの名前は、**名前**フィールドに入力します。

値タイプのリストから、スライダーのデータソースに於けるデータのタイプに応じて、**整数**または**実数**(デフォルト)を選択します。

最小値(デフォルト: 0)および**最大値**(デフォルト: 1)のフィールドに最小値と最大値を入力して、スライダーでカバーされるデータの範囲を定義します。

ステップ数のフィールド(デフォルト: 5)には、スライダーのステップ数(解像度)を入力します。


方向のリストから**水平**(デフォルト)または**垂直**を選択して、スライダーの向きを水平から垂直、またはその逆に変更します。

ツールチップフィールドには、ユーザーがスライダー上にカーソルのポインタを置いたときにツールチップとして表示されるテキストを入力します。




値に単位を追加を選択し、関連するテキストフィールドに単位を入力することによって、スライダーの値の数値に単位を追加することができます。単位は、ソースが文字列宣言とパラメータである場合にのみ適用されます。

また、**設定**ウィンドウには、以下のセクションがあります。

ソース

このセクションでは、スライダーのデータソースとして使用する変数を定義します。このセクションには、**アプリケーションビルダー**ウィンドウと**モデルビルダー**ウィンドウのツリーのフィルター処理されたビューが表示されます。ノードは、何らかの種類のデータを表すか、または子ノードを持ちます。スライダーでは、モデルの変数からのスカラー値、および**宣言**の下で定義された変数をデータソースとして使用できます。使用可能なデータノードのリストを拡張するには、**ソース**セクションのヘッダーで**モデルビルダーへスイッチしデータアクセスをアクティブ**のボタン()をクリックして、モデルビルダーに切り替わってから、アクセスしたいデータのある**モデルビルダー**ブランチ内のノードを選択します。このボタンをアクティブにすると、選択したノードの**設定**ウィンドウには、含めることが可能な設定箇所の横に**データソース**選択チェックボックス(緑色の四角形)が表示されます。

そのチェックボックスをクリックして選択すると、スライダーに使用可能なソースノードとしてそのデータが挿入されます。

データを表すノードを選択すると、ツリーの下にあるソースとして使用ツールバーボタン()が有効になります。それをクリックするか、代わりにEnter キーを押すか、ダブルクリックするか、ノードを右クリックしてソースとして使用を選択して、選択ソースとして追加します。また、ソースセクションのヘッダーの作成ボタン()をクリックして、新規変数宣言を作成してそれをソースとして使用することもできます。変数を作成および使用のダイアログボックスが開き、ソースのデータタイプ(該当する場合)、名前、および初期値を選択できます。名前は、既存の変数宣言と競合することはできません。ツリーの下ノード編集ツールバーボタン()をクリックすると、対応するノードに移動します。必要に応じて、プログラムがモデルビルダーに切り替えます。

ノードをソースとして選択すると、そのノードは**選択ソース**の下に選択されたソースとして表示されます。



複数のフォームオブジェクトで同じデータソースを使用しようとすると、いくつかの奇妙な副作用が発生する可能性があります。ソースのデフォルト値は期待通りとは限りません。あるフォームオブジェクトのデフォルト値が他のフォームオブジェクトの一つに対して無効であると、深刻なエラーが発生することもあります。

初期値のリストから**データソース参照**を選択すると、選択されているデータソースで指定された値が使用されます。また、**カスタム値**を選択すると、その下にある**値フィールド**にスライダー(デフォルト値: 0)のための初期値(起動時の値)を入力することができます。

単位

メソッドのリストから、次のいずれかの方法を選択してスライダーの単位を指定します。

- **単位なし**: デフォルトオプション。
- **値に単位を追加**: **単位式**のフィールドに入力した単位が、スライダーの設定位置に対応する数値に追加されます。
- **単位セットから単位を追加**: スライダーは、**宣言**の下に追加された**単位セット**ノードから単位を追加します(**単位セットノード**を参照)。単位セットのリストから使用する単位セットを指定し、**単位リスト**のリストから使用する単位を指定します。単位リストには、選択された単位セットの全ての定義済みプロパティとその単位がリストされます。**単位なし**の選択肢もあります。

外観

外観では、ユーザーがアプリケーションを実行するときにスライダーの初期状態を制御できます。デフォルトでは、スライダーは可視表示されて有効になっています。スライダーが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

位置およびサイズ

このセクションには、親フォームのグリッドにあるスライダーの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、スライダーの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォーム**ツールバーの**スケッチ**セクションの中の**アレンジ**メニューでアラインメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なること

があり、位置が少しずつ異なるためです。アラインメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

スライダーの幅を指定することもできます。幅フィールドに幅(ポイント単位で)を、高さフィールドに高さ(ポイント単位で)を入力します。水平アラインメントのリストから全幅を選択した場合は、最小幅を指定することができます。最小幅を自動的に計算するには自動を選択します(通常は最小サイズ0を意味します)。または、マニュアルを選択して、その下のテキストフィールドに最小幅を指定します。高さフィールドは、スライダーの高さがソフトウェアによって決められるため設定変更できません。

さらに、スケッチモードでは、位置 x および位置 y フィールドを使用して、スライダーの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、行、列、行スパン、および列スパンの値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン(グリッドモードの場合のみ)では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。セルマージンのリストから、以下を選択します。

- なし: フォームオブジェクトの周りに余白を使用しません。
- 親フォーム参照(デフォルト): 親フォームに設定されている余白を使用します。
- カスタム: フォームオブジェクトの余白を水平および垂直のテキストフィールドに指定します。

イベント

特定のタイプのフォームオブジェクトでは、データが入力されたといったようなイベントが発生した場合に実行するメソッドを指定することができます。データ変更時のリストには、なし(デフォルト)と使用可能なメソッドが含まれています。イベントのローカルメソッドを追加するには、データ変更時のリストの右側にあるローカルメソッド作成のボタン(+)をクリックするか、またはスライダーオブジェクトを右クリックします。すると、データ変更時リストで選択したメソッドがローカルメソッドに変更され、新規メソッドのための空の onDataChange エディターウィンドウが開かれます。後に、そのローカルメソッドをメソッドエディターウィンドウで開くには、ソースへのボタン(📄)をクリックします。ローカルメソッドを削除するには、ローカルメソッドを削除のボタン(X)をクリックします。スライダーから呼び出されるメソッドは、データソースの新しい値を保持する倍精度の引数をサポートしています。また、スライダーオブジェクトを右クリックしてローカルメソッドを作成するか、(メソッド編集またはローカルメソッド編集を選択して)コマンドに関連付けられたメソッドを開きます。

スライダー上で Ctrl + Alt を押しながらクリックした場合、

- データ変更時リストがメソッドに設定されている場合には、メソッドエディターウィンドウが開かれます。
- データ変更時リストがなしに設定されている場合には、(必要に応じて)ローカルメソッドを作成してそのリストにローカルメソッドが設定され、ローカルメソッドのエディターウィンドウが開かれます。

データが変更されると、新しいデータ値がデータソースに格納された後にイベントが発生されます。

デフォルトでは、ドラッグ中にトリガーのチェックボックスが選択されています。スライダーをドラッグしている間にイベントがトリガーされます。イベントに接続されているメソッドが実行に時間がかかる場合、スライダーは低速に見えることがあります。ドラッグ中にトリガーのチェックボックスをオフにしている場合には、スライダーをドラッグして解除するときだけイベントがトリガーされます。

ハイパーリンク

ハイパーリンク(🌐)フォームオブジェクトは、追加情報または関連情報を含むウェブページへのハイパーリンクを追加する機能を提供してくれます。

ハイパーリンクオブジェクトの名前、名前フィールドに入力します。

テキストフィールドには、クリック可能なハイパーリンクとしてフォームに表示されるテキストを入力します。

URL フィールドには、有効なウェブアドレスまたは電子メールアドレスを入力します。

有効なウェブアドレスとしては、www.comsol.com のように、http:// または https:// プレフィックスは通常必要ありません。ウェブページがユーザーのデフォルトブラウザで開かれます。

電子メールアドレスの場合は、mailto:info@comsol.com のように mailto 形式を使用して、電子メールを送信するためのハイパーリンクを提供します。また、指定された件名とメッセージ本文を含めることもできます。ハイパーリンクは、ユーザーのデフォルトの電子メールアプリケーションプログラムを起動し、To フィールドが指定アドレスに設定されている新規メッセージを準備します。COMSOL アプリケーションから電子メールをインタラクティブに送信する方法は、組み込みの電子メールメソッドを使用する方法とは異なる点に注意してください。

設定ウィンドウには、以下のセクションがあります。

位置およびサイズ

このセクションには、親フォームのグリッドにあるハイパーリンクオブジェクトの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、ハイパーリンクオブジェクトの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向（該当する場合）の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向（該当する場合）の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチ**セクションの中の**アレンジ**メニューでアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

ハイパーリンクオブジェクトの幅を指定することもできます。**幅**フィールドに幅（ポイント単位で）を、**高さ**フィールドに高さ（ポイント単位で）を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します（通常は最小サイズ 0 を意味します）。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。**垂直アライメント**のリストから**全幅**を選択した場合は、**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します（通常は最小サイズ 0 を意味します）。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、ハイパーリンクオブジェクトの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン（グリッドモードの場合のみ）では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**： フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**（デフォルト）： 親フォームに設定されている余白を使用します。
- **カスタム**： フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、ハイパーリンクのテキストと背景の外観を制御できます。


背景色のリストから、このテキストの背景として使用する色を選択します。**透明**(デフォルト)、定義済みの基本色のどれかを選択します。また、**カスタム**を選択して、カラーパレットからカスタムカラーを選択することもできます。

このテキストのフォントとフォントサイズには、**フォームノード**のフォント設定がデフォルトとして使用されます。**フォントリスト**から、**デフォルトフォント**またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ**のコンボボックスでフォントサイズ(ポイント単位)を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ**が使用されます。

太字のフォントを使用するには**ボールド**のチェックボックスを選択し、斜体(イタリック体)を使用するには**イタリック**のチェックボックスを選択します。

状態では、ユーザーがアプリケーションを実行するときにハイパーリンクオブジェクトの初期状態を制御できます。デフォルトでは、ハイパーリンクオブジェクトは可視表示されて有効になっています。ハイパーリンクオブジェクトが非表示または使用不可になるように初期状態を設定する場合は、**可視**または**有効化**のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されません。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。







ツールバー

ツールバー()は、他のフォームオブジェクトと同じようにフォーム上に配置されるツールバーです。これによって、ウィンドウのタイトルよりも下側にメニューボタンのバーが表示されます。このメニューボタンをクリックすると一連のコマンドが実行されます。これは他のメニューと同じように機能しますが、サブメニューはサポートされません。フォームウィンドウのツールバーオブジェクトを右クリックし、**項目追加**を選択してアクション項目をツールバーに追加する(**項目ノード**を参照)か、**セパレーター追加**を選択してツールバーボタンのグループ間にセパレーターを追加します(**セパレーターノード**を参照)。**ツールバー項目**のセクションで、ツールバーのコンテンツを追加したり編集したりすることもできます。ツールバーオブジェクトの名前は、名前フィールドに入力します。

二つのアイコンサイズから選択することができます。**アイコンサイズ**リストから、**小**(デフォルト)または**大**を選択します。

設定ウィンドウには、以下のセクションがあります。

ツールバー項目

このセクションのテーブルには、名前、アイコン、およびテキストの列に、現在のツールバー項目とセパレーターがリスト表示されます。**テキスト**の列で直接テキストを編集することができます。テーブルに設定してある項目を右クリックし、コンテキストメニューから移動、削除、またはカスタムボタンのための編集を選択できます。編集を選択した場合には、表示される**カスタムツールバーアイテム編集**ウィンドウで編集します。また、テーブルの下にある**上へ移動**()、**下へ移動**()、**削除**()、および**編集**()のボタンを使用することもできます。**項目追加**ボタン()をクリックすると、項目をツールバーに追加することができます。**セパレーター追加**ボタン()をクリックすると、セパレーターが追加されます。

テーブルには、追加された各項目の行が含まれており、**名前**、**アイコン**、**テキスト**、および**ツールチップ**の各列にそれぞれ、名前、アイコン、テキスト、およびツールチップが表示されます。

位置およびサイズ

このセクションには、親フォームのグリッドにあるツールバーの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、ツールバーの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向(該当する場合)の全幅までサイズ変更されます。

アラインメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバー**の**スケッチセクション**の中の**アレンジメニュー**でアラインメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アラインメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。**幅と高さ**のフィールドは、ツールバーの寸法がソフトウェアによって決められるため設定変更できません。

さらに、スケッチモードでは、**位置 x** および **位置 y** フィールドを使用して、ツールバーの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および **列スパン** の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。

セルマージン

セルマージン (グリッドモードの場合のみ) では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン** のリストから、以下を選択します。

- **なし**: フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照** (デフォルト): 親フォームに設定されている余白を使用します。
- **カスタム**: フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

外観

このセクションでは、ツールバー内のテキストと背景の外観を制御できます。

テキスト色 のリストから、テキストに使用する色を選択します。**継承** (デフォルト; フォームオブジェクトが配置されているフォームの設定を使用)、定義済みの基本色のどれかを選択します。また、**カスタム** を選択して、カラーパレットからカスタムカラーを選択することもできます。


背景色 のリストから、ツールバーのテキストの背景として使用する色を選択します。**透明** (デフォルト)、定義済みの基本色のどれかを選択します。また、**カスタム** を選択して、カラーパレットからカスタムカラーを選択することもできます。

テキストのフォントとフォントサイズには、**フォームノード** のフォント設定がデフォルトとして使用されます。**フォントリスト** から、**デフォルトフォント** またはリストにある使用可能なフォントを選択してください。必要に応じて、**フォントサイズ** のコンボボックスでフォントサイズ (ポイント単位) を選択するか入力します。デフォルトでは、フォントの**デフォルトサイズ** が使用されます。

太字のフォントを使用するには**ボールド** のチェックボックスを選択し、斜体 (イタリック体) を使用するには**イタリック** のチェックボックスを選択します。

状態 では、ユーザーがアプリケーションを実行するときにツールバーの初期状態を制御できます。デフォルトでは、ツールバーは可視表示されて有効になっています。ツールバーが非表示または使用不可になるように初期状態を設定する場合は、**可視** または **有効化** のチェックボックスをオフにします。メソッドを使用して、可視にするか有効にすることもできます。フォームエディターでは、フォームオブジェクトの状態は外観の変更によって示されます。非表示に設定されているオブジェクトは、フォームエディターで選択する際には可視化されています。

スペーサー

スペーサー () フォームオブジェクトは、ユーザーインターフェースでは表示されません。これは、スペースの絶対値のサイズを定義するもので、隣接するフォームオブジェクトがそれらのコンテンツを表示するのに十分なスペースを保証するために使用されます。通常、テーブルやプロットの横にスペーサーを使用して、それらが適切に表示されるようにします。ユーザーがウィンドウのサイズをスペーサーのサイズより小さくするように変更すると、スクロールバーが表示されてウィンドウの有効サイズは維持されます。スペーサーオブジェクトの名前は、**名前** フィールドに入力します。

設定ウィンドウには、以下のセクションがあります。

位置およびサイズ

このセクションには、親フォームのグリッドにあるスペーサーの全てのレイアウト設定が含まれています。

グリッドモードでは、次のリストを使用して、スペーサーの水平および垂直の配置を制御できます。

- **水平アライメント**のリストから、**中心**、**左**、**右**、または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向（該当する場合）の全幅までサイズ変更されます。
- **垂直アライメント**のリストから、**中間**、**上**、**下**または**全幅**を選択します。これにより、フォームオブジェクトが自動的に拡大され、セルの水平または垂直方向（該当する場合）の全幅までサイズ変更されます。

アライメントを指定する必要性は、グリッドモードでの作業に於いて、フォームオブジェクトがグリッドセル内でどのように配置されるかを制御するような場合に最も明白です。フォームオブジェクトを整列させることはスケッチモードでも役立ちます。その場合、**フォームツールバーのスケッチセクション**の中の**アレンジメニュー**でアライメントツールを使用できます。Windows クライアント以外のクライアントでアプリケーションを実行する場合、フォームオブジェクトがアプリケーションビルダーで表示されている通りに正確に配置されないことがあります。これは、フォームオブジェクトが他のクライアントではサイズが異なることがあり、位置が少しずつ異なるためです。アライメントを指定することによって、フォームオブジェクトを全てのクライアントに整列させられるようにすることができます。

スペーサーの幅を指定することもできます。**幅**フィールドに幅（ポイント単位で）を入力し、**高さ**フィールドに高さ（ポイント単位で）を入力します。**水平アライメント**のリストから**全幅**を選択した場合は、**最小幅**を指定することができます。最小幅を自動的に計算するには**自動**を選択します（通常は最小サイズ 0 を意味します）。または、**マニュアル**を選択して、その下のテキストフィールドに最小幅を指定します。**垂直アライメント**のリストから**全幅**を選択した場合は、**最小高さ**を指定することができます。最小高さを自動的に計算するには**自動**を選択します（通常は最小サイズ 0 を意味します）。または、**マニュアル**を選択して、その下のテキストフィールドに最小高さを指定します。

さらに、スケッチモードでは、**位置 x** および**位置 y** フィールドを使用して、スペーサーの絶対位置を指定できます。グリッドモードでは、オブジェクトを配置しているグリッド位置を、**行**、**列**、**行スパン**、および**列スパン**の値として表示できます。



これらの位置とサイズの設定は、フォームウィンドウで実行されるオブジェクトのサイズ変更や移動などの描画操作に追従して更新されます。



幅と高さの設定は、スペーサーが占める領域を指定します。通常は、幅または高さのいずれかを非常に小さな値に設定し、もう一方を 1 方向の希望するサイズに設定します。二つの方向に対して全幅設定を使用すれば、2 方向のサイズに適用させることができます。

セルマージン

セルマージン（グリッドモードの場合のみ）では、フォームオブジェクトの周りのマージンを制御できます。デフォルトでは、マージンは親フォームから取得されます。**セルマージン**のリストから、以下を選択します。

- **なし**：フォームオブジェクトの周りに余白を使用しません。
- **親フォーム参照**（デフォルト）：親フォームに設定されている余白を使用します。
- **カスタム**：フォームオブジェクトの余白を**水平**および**垂直**のテキストフィールドに指定します。

メソッドの操作

この章では、アプリケーションまたはモデルにおいて、コードが記述されたメソッドを作成するために使用可能なアプリケーションビルダーのツールについて説明しています。これらのツールは、言語要素、自動コード記録、シンタックスチェック、検索ツール、デバッグツール、およびメソッドとユーティリティクラスの作成、編集、テスト、およびデバッグのためのその他の機能を提供してくれます。



以下に、この章の構成を示します。

- [概要](#)
- [メソッドの作成](#)
- [アプリケーションのデバッグ方法](#)



概要

アプリケーションビルダーは、アプリケーションが組み込み機能を拡張するタスクを実行するために使用するコードを記述する目的でデザインされたツールとエディターウィンドウを提供しています。

メソッドエディターウィンドウを開く


メソッドリボンツールバーからメソッドエディターツールをアクティブにします。メソッドエディターウィンドウを開くには、タブをクリックするか、新規メソッドを作成するか、メソッドノードをダブルクリックするか、メソッドノードを右クリックして**編集**を選択します。複数のメソッドエディターウィンドウを開くことができます。メソッドエディターウィンドウのタブには、 (アプリケーションメソッド用) または  (モデルメソッド用) とメソッドの名前が表示されます。

コーディングとメソッドの概要

アプリケーションビルダーでは、カスタムコードがメソッドノードで表されます。コードを記述するには、最初にメソッドノードを作成する必要があります。メソッドノードを作成するには、リボンツールバーの**新規メソッド**のボタン()をクリックするか、**アプリケーションビルダー**ウィンドウのメソッドノード()を右クリックして**新規メソッド**を選択します。

新しいメソッドノードは、アプリケーションビルダーツリーのメソッドノードの下に表示され、アプリケーションメソッドを含んでいます。このようなメソッドは、全てのアプリケーションメソッドおよびフォームオブジェクトで使用することから、グローバルな利用が可能です。また、フォームの一部のフォームオブジェクトにローカルメソッドを作成することもできます。ローカルメソッドは、定義されているオブジェクトの外部でのアクセスや表示はできません。これらのローカルメソッドは、チェックボックスの設定(データ)が変更されたような場合に、フォームオブジェクト内のイベントに接続します。メソッドノードは、コマンドシーケンスまたはフォームオブジェクトのイベントからメソッドを参照するために使用されます。モデルビルダーのモデルで使用の場合は、モデルメソッドのための**モデルメソッド**ノードを作成することもできます。モデルノードは、アプリケーションメソッドと同じように機能しますが、モデルビルダーで実行して COMSOL マルチフィジックスモデルの一部を拡張またはカスタマイズすることができます。

メソッドノードには、COMSOL Multiphysics®コードと Java®コードが含まれています。アプリケーションビルダーウィンドウでダブルクリックするか、右クリックして**編集**を選択して、検査および編集を行うことができます。メソッドを開くと、メソッドエディターウィンドウにそのメソッドが表示されます。

エディタータブの中でコードに加えられた変更は、タブを閉じるかコードをコンパイルした後にアプリケーションビルダーモデルに格納されます。コードのシンタックスをコンパイルしてチェックするには、リボンツールバーの**シンタックスチェック**ボタン()をクリックします。コードは、ApplicationMethod Java クラスを定義します。このクラスには、アプリケーションがそれを実行するときにコマンドシーケンスが呼び出す execute というメソッドがあります。典型的な使用例は、メソッドノードを作成し、execute メソッドにコードを記述し、このコードをアプリケーションのユーザーインターフェースのフォームオブジェクトがトリガーできるコマンドシーケンスにリンクすることです。フォームオブジェクトの状態に影響するようなメソッドは、即座に適用されません。

デフォルトの設定では、execute メソッド内に含まれるコードのみが表示されます。全てのコードを表示するには、**コード全表示**のチェックボックスをオンにします。これは、**環境設定**ダイアログボックスの**メソッド**セクションにおける環境設定です。

アプリケーションの実行中に execute メソッドがトリガされる際に、model はそのアプリケーションの**モデルオブジェクト**です。このメソッドでは、フィジックスのフィーチャにアクセスしてパラメータを変更することができます。

アプリケーションビルダーウィンドウ

アプリケーションビルダーウィンドウは、アプリケーションビルダーを使用しているときに、デフォルトでデスクトップの左端に配置されているウィンドウです。そのツリーには、アプリケーション内の全てのコンポーネント（フォーム、メソッド、ファイルなど）のノードが含まれています。

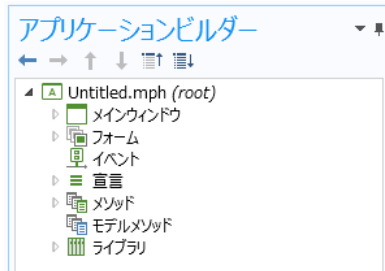


Figure 4-1: アプリケーションビルダーウィンドウ

ツリー内のメソッドノードをダブルクリックすると、そのコードがエディタータブに表示されます。ノードに関連付けられたエディタータブがすでに開いている場合は、タブにフォーカスが置かれます。ツリー内のノードを右クリックすると、コンテキストメニューが開きます。このメニューには編集オプションがあり、これによってエディターが開かれもします。コンテキストメニューには、上へ移動、下へ移動、コピー、複製など、モデルビルダーツリーノードの標準オプションも含まれています。

エディタータブが開いている場合にツリー内のメソッドノードを選択すると、ノードに関連付けられているエディタータブにフォーカスが移ります。同様に、エディタータブをクリックしてそこにフォーカスされると、それに対応するノードがアプリケーションビルダーウィンドウで選択されます。

メソッドウィンドウ




各メソッドにおいて、メソッドウィンドウにはメソッドのコードが含まれています。手動でコードを入力したり、モデリング手順のコードを記録したり、モデルコード、モデル方程式、および言語要素ウィンドウからコードを挿入することができます。メソッドエディターウィンドウでは、シンタックスハイライトを使用します。これは、環境設定ダイアログボックスで定義することができます（シンタックスハイライトとコメントを参照）。メソッドエディターウィンドウでは、標準のキーボードショートカットとコンテキストメニューを使用して、コードの一部を選択、コピー、カット、ペースト、および削除することができます。コード行をトリプルクリックするか、メソッドエディターウィンドウの左側にある行番号をクリックすると、コード行全体を選択することができます。







メソッドツールバー

メソッドのコンテキストリボンツールバーは、メソッドの作成、編集、およびデバッグのためのほとんどの機能へのアクセスを提供しています。それは、メソッドエディターウィンドウがアクティブな場合に使用できます。このセクションでは、メソッドツールバーのボタンについて簡単に説明しています。

メインセクション




このセクションには、別のウィンドウに移動して新規フォームとメソッドを作成するための以下のボタンが含まれています。

- **モデルビルダーボタン** (): アプリケーションビルダーからモデルビルダーウィンドウ、および標準の COMSOL デスクトップに切り替えます。
- **新規フォームボタン** (): 新規フォームウィザードを使用して新しいフォームを作成します。 [フォームの操作と新規フォームウィザードの使用](#) を参照してください。
- **新規メソッドボタン** (): モデルに新しいメソッドノードを作成し、そのコードを新しいエディタータブで開きます。 [メソッドノードとメソッドエディターウィンドウ](#) を参照してください。

- **データアクセスボタン** (): モデルに依存するデータとプロパティ、および実行中のアプリケーションからデータアクセス機能を通して変更可能なアプリケーション固有のプロパティを追加します。[データアクセス](#)を参照してください。
- **メソッドを記録ボタン** (): 埋め込みモデルの操作の記録セッションを開始して、新しいメソッドを作成します。その後、埋め込みモデルはそのメソッド内でコードとして利用できます。記録が開始されると、そのボタンは**記録停止ボタン** () に変わり、クリックすると記録が終了します。[コードの記録](#)を参照してください。
- **設定ボタン** (): **設定**ウィンドウを開いたり閉じたりします。
- **プレビューボタン** (): アプリケーションのフォームとメソッドのライブプレビューのための**プレビュー**ウィンドウを、表示または非表示にします。**プレビュー**ウィンドウでは、スクロールしてアプリケーション内の全てのフォームとメソッドのプレビューをみることができます。例えば、フォームとやりとりするメソッドを操作している場合などに便利です。**プレビュー**ウィンドウでフォームのプレビューを表示するには、**アプリケーションビルダー**ウィンドウでフォームまたはメソッドノードを選択します。
- **エディターツールボタン** (): **エディターツール**ウィンドウを表示または非表示にします。エディターツールウィンドウでは、よく使う COMSOL マルチフィジックスのモデル操作を選択してメソッドに挿入したり、それらに基づいてフォームオブジェクトを生成したりすることができます。[モデルコードとフォームオブジェクトの追加](#)を参照してください。

ライブラリセクション

このセクションには、外部コードとユーティリティクラスを挿入するための以下のボタンが含まれています。

- **ユーティリティクラスボタン** (): モデルに新しい**ユーティリティクラス**ノードを作成し、そのコードを新しいエディタータブで開きます。[ユーティリティクラスノード](#)を参照してください。
- **外部 Java ライブラリボタン** (): モデルに新しい**外部 Java ライブラリ**ノードを作成します。[外部 Java ライブラリノード](#)を参照してください。
- **外部 C ライブラリボタン** (): モデルに新しい**外部 C ライブラリ**ノードを作成します。[外部 C ライブラリノード](#)を参照してください。








編集セクション


編集セクションには、次のボタンがあります。

- **保存状態に戻すボタン** (): メソッドを最後に保存してから行った変更を破棄し、保存したバージョンに戻します。

コードセクション


このセクションには、コードをより直接的に扱う以下のボタンが含まれています。

- **言語要素ボタン** (): **言語要素**ウィンドウを表示または非表示にします。ここで、言語要素を選択してメソッドに挿入することができます。[言語要素の追加](#)を参照してください。
- **モデル表現ボタン** (): **モデル表現**ウィンドウを表示または非表示にします。ここでは、COMSOL マルチフィジックスのモデル表現を選択してメソッドに挿入することができます。[モデル表現の追加](#)を参照してください。
- **シンタックスチェックボタン** (): 作成した全てのメソッドのシンタックスをチェックします。メソッドのシンタックスエラーと警告は、**エラーと警告**ウィンドウに表示されます。[エラーと警告ウィンドウ](#)を参照してください。
- **ノードへボタン** (): 選択したソースコードのモデルエンティティに対応する組み込みモデルのノードに移動します。[ソースコードがマップされるノードへ移動](#)を参照してください。
- **コードを記録ボタン** (): メソッドのコードとして後で使用できる埋め込みモデル操作の記録セッションを開始します。記録が開始されると、ボタンは**記録停止ボタン** () に変わり、クリックすると記録が終了します。[コードの記録](#)を参照してください。
- **ショートカットを使用ボタン** (): 特定のフォームの表現でショートカットとして使用するローカルメンバーフィールド変数を作成します。[ショートカットの使用](#)を参照してください。

- ローカル変数を作成ボタン(


デバッグセッション

このセッションには、メソッドをデバッグするためのボタンが含まれています。アプリケーションのデバッグ方法を参照してください。ここに含まれるボタンを、以下に示します。

- 継続ボタン(


ブレークポイントセッション

このセッションには、すべてのブレークポイントを削除および無効にするための次のボタンが含まれています。

- 全て削除ボタン(




テストセッション

このセッションには、アプリケーションをテストするための次のツールが含まれています。



- アプリケーションをテストボタン(

ビューセッション

ビューセッションには、アプリケーションビルダーのデスクトップウィンドウでビューを再配置するための次のボタンがあります。


- タイル表示()と移動先()ボタン: アプリケーションビルダーのウィンドウを並べ替えます。
- デスクトップリセットボタン(): デスクトップのレイアウトをデフォルトの状態にリセットします。

メソッドノードとメソッドエディターウィンドウ

メソッドノード()とモデルメソッドノード()には、ユーザ定義のメソッドが含まれます。新しいメソッドノードを追加するには、メソッドノードを右クリックして新規メソッドを選択します。新しいモデルメソッドノードを追加するには、モデルメソッドノードを右クリックして、新規モデルメソッドを選択します。メソッドツールバーの新規メソッドのボタンをクリックすることもできます。新規メソッドのダイアログボックスが開いたら、名前フィールドにメソッドのラベルを入力し、メソッドタイプのリストからアプリケーションメソッドまたはモデルメソッドのメソッドのタイプを選択します。

- アプリケーションメソッドでは、アプリケーションビルダーを使用して作成されるアプリケーション内のボタン、ウィンドウ、およびその他のコンポーネントに接続する機能が提供されます。アプリケーションメソッドはグローバルメソッドであり、全てのアプリケーションメソッドとフォームオブジェクトからのアクセスが可能です。
- モデルメソッドでは、現在のセッションでモデルビルダーによって表されるモデルオブジェクトを直接変更することができます。例えば、いくつかのマニュアルステップから成るモデリングタスクを自動化するために使用することができます。モデ

ルメソッドはグローバルメソッドであり、他の全てのメソッドからのアクセスが可能です。アプリケーションメソッドはモデルメソッドを呼び出すことはできますが、その逆はできません。



メソッドを編集するには、メソッドノードをダブルクリックするか、右クリックして**編集**()を選択します。エディターウィンドウが開かれ、そこでメソッドのコードを編集することができます。**コード全表示**の環境設定に応じて、メソッド宣言のみか、あるいは全部のクラスが表示されます。エディターでは、コードの一部をハイライト表示して右クリックすることによって、カット、コピー、ペースト、および削除を行うことができます。また、Ctrl + C などの標準的なキーボードショートカットを使用してコピーしたり、Ctrl + A を押してエディターウィンドウ内の全てのコードを選択したり、Delete キーを使用することもできます。このコードは、ApplicationMethod という Java クラスを拡張し、次のシグネチャを持つメソッドが一つだけ必要です。


```
public void execute () {
    model.physics ( "es" ). feature ( "ccn1" ). set ( "ConstitutiveRelationD", 1,
    model.modelData ( ). getString ( "dielectricModel" ) );
}
```

このメソッドでは、COMSOL マルチフィジックスの API で使用可能な操作を実行することができます。拡張クラスを介して利用可能ないくつかのメソッドとメンバがあります。中でも、model というメンバとしてアクセス可能なモデルオブジェクトが最も重要です。上記のコード例では、アプリケーション自身のデータ(Electrostatics インタフェース)から dielectricModel という名前のデータフィールドを読み込み、それを組み込みモデルのフィジックスフィーチャのパラメータに設定しています。

メソッドノードの**設定**ウィンドウで追加できる入力と出力についての詳細は、[メソッドノード](#)または[モデルメソッドノード](#)を参照してください。

ユーティリティクラスノード

ユーティリティクラスノード()には、他のメソッドから呼び出せるメソッドを持つユーティリティクラスが含まれています。新しい**ユーティリティクラスノード**を追加するには、**ライブラリノード**()を右クリックし、**ユーティリティクラス**を選択するか、リボンツールバーの**ユーティリティクラス**ボタンをクリックします。ユーティリティクラスの名前は、**名前**フィールドに入力します。

ユーティリティクラスを使用すると、アプリケーション内のメソッド間で Java コードを共有したり、アプリケーション間で実装をコピーしたりすることができます。ユーティリティクラスで宣言されたメソッドは、アプリケーションの他のメソッドから呼び出すことができます。ユーティリティクラスコードを編集するには、**ユーティリティクラスノード**をダブルクリックするか、右クリックして**編集**()を選択します。エディターウィンドウが開かれ、そこでユーティリティクラスのコードを編集することができます。**コード全表示**の環境設定に応じて、メソッド宣言のみか、あるいは全部のクラスが表示されます。

例えば、アプリケーション内の全てのジオメトリを作成する次のメソッドを考えてみましょう。

```
public static void runAllGeom () {
    model.geom ( ). run ( );
}
```



ユーティリティクラスの名前は、ユーティリティクラスノードのタグと同じでなければなりません。**ユーティリティクラスノード**の名前が util1 であるとする、上記のメソッドを呼び出すには、その他のメソッドで util1.runAllGeom (); と入力します。



ユーティリティクラスの名前を変更する場合は、ユーティリティクラスの名前がタグと一致するようにソースコードを更新する必要があります。それ以外の場合、コードはコンパイルされません。

コード全表示の環境設定を有効にすると、コードエディターは、ユーティリティクラスが ApplicationLanguageBase という名前の抽象基本クラスを拡張していることが表示されます。この継承により、(前述のように) モデルオブジェクトにアクセスすることが可能になり、他の便利なメソッドを使用することができます。また、この継承を削除して、別のユーティリティクラスやその他のアクセス可能なクラスをインスタンス化したり拡張することもできるクラスを作成することができます。ユーティリティクラスのパッケージは builder であり、この名前を維持することをお勧めします。

外部 Java ライブラリノード

外部 Java ライブラリノード()を使用して、メソッドに含めるコンパイル済みの Java ライブラリをインポートします。Java ライブラリ (JAR ファイル) は、サードパーティの Java 開発ツールで作成することができます。新しい**外部 Java ライブラリ**を追加するには**ライブラリノード**()を右クリックし、**外部 Java ライブラリ**を選択するか、リボンツールバーの**外部 Java ライブラリ**のボタンをクリックします。表示するラベルは、**ラベル**フィールドに入力します。

外部 Java ライブラリノードの設定ウィンドウには、以下のセクションがあります。

ライブラリをインポート

ファイル名のフィールドにインポートする Java ライブラリファイルの名前を入力するか、**参照**をクリックして Java ライブラリファイル (JAR ファイル) を選択します。次に、**開く**をクリックして、ライブラリをアプリケーションにインポートします。インポート中に、JAR ファイルがアプリケーションにコピーされます。アプリケーションを保存すると、そこには JAR ファイル全体のコピーも含まれています。また、JAR ファイルで宣言されているパッケージの import ステートメントは、アプリケーション内の全ての既存のメソッド、将来のメソッド、およびユーティリティクラスのソースコードに追加されます。そして、アプリケーション内の全ての Java コードが、クラスパス上に追加された JAR ファイルとともに再コンパイルされます。



将来、コードをコンパイルする場合、JAR ファイルはクラスパス上にあります。



ライブラリのアップデート版を使用するには、もう一度**インポート**ボタンをクリックします。アプリケーションにはライブラリのコピーが含まれているため、元の JAR ファイルのみをアップデートするだけでは不十分です。

外部 Java ライブラリノードを削除すると、全ての Java ソースコードがアップデートされて、JAR ファイルで定義されたインポートが除外され、アプリケーション内の全ての Java コードが再コンパイルされます。

外部 C ライブラリノード

外部 C ライブラリノード()を使用して、例えばメソッドに含める C コードに基づいて、コンパイルされたネイティブコードライブラリをインポートすることができます。新しい**外部 C ライブラリノード**を追加するには、**ライブラリノード**()を右クリックして**外部 C ライブラリ**を選択するか、リボンツールバーの**外部 C ライブラリ**ボタンをクリックします。表示するラベルを**ラベル**フィールドに入力し、外部 C ライブラリの名前を**名前**フィールドに入力します。

任意のプログラミング言語で書かれ、アプリケーションのターゲットプラットフォーム用にコンパイルされたネイティブライブラリをインポートすることができます。これは、例えば、コンピュータに接続された特定のハードウェア (測定機器など) にアクセスしたり、負荷の重い計算を高速化するのに便利です。呼び出すメソッドは、特定のプラットフォームの C 言語呼び出し規約に従わなければなりません。多くのプログラミング言語では、この要件を満たすメソッドを生成することができます。

設定ウィンドウで、サポートするつもりのプラットフォームに対応するセクションを展開し、インポートしたい共有ライブラリを参照して選択します。Windows®オペレーティングシステムでは、共有ライブラリのファイル拡張子は通常 DLL です。Linux®では、拡張子は通常.so です。macOS では、通常.dylib です。ファイルシステム内のファイルを変更して、新しいバージョンをアプリケーションにインポートしたい場合には、**リロード**ボタンをクリックします。アプリケーションにインポートする別のファイルを選択するには、**編集**ボタンをクリックします。

外部 C ライブラリノードの設定ウィンドウには、以下のセクションがあります。

Windows, 64 ビットのためにライブラリをインポート

このセクションはデフォルトで拡張されています。**参照**をクリックし、64 ビット Windows®用のコンパイル済みネイティブコードライブラリファイル (DLL ファイル) を検索し選択します。

Linux, 64 ビットのためにライブラリをインポート

参照をクリックし、64 ビット Linux®のコンパイル済みネイティブコードライブラリファイル (.so ファイル) を検索し選択します。

OS X, 64 ビットのためにライブラリをインポート

参照をクリックし、64 ビット MacOS のコンパイル済みネイティブコードライブラリファイル(dylib ファイル)を検索し選択します。

ロード中

デフォルトでは、**サポート外のプラットフォームへのアップロードを許可**のチェックボックスが選択されているため、サポートされていないプラットフォームで共有ライブラリを読み込むことができます。このようなプラットフォームでは、メソッドからライブラリにアクセスしようとするとエラーが発生します。チェックボックスをクリアすると、サポートされていないプラットフォームでアプリケーションを起動しようとするとエラーメッセージが直接表示され、そのアプリケーションは起動しません。

外部 C ライブラリの使用

アプリケーションでライブラリを使用するには、メソッドまたはユーティリティクラスからライブラリを呼び出す必要があります。ApplicationMethod クラスまたは ApplicationLanguageBase クラスで使用可能な External external (String libraryTag) メソッドを使用して、以下のメソッドによる External タイプのオブジェクトを返します。

TABLE 4-1: EXTERNAL オブジェクトのメソッド

メソッド	説明
long invoke (String method, Object... arguments)	指定された引数を使用して、ライブラリ内の名前付きネイティブメソッドを呼び出します。文字列は char * に変換されます。メソッドから返された値を返します。
long invokeWideString (String method, Object... arguments)	指定された引数を使用して、ライブラリ内の名前付きネイティブメソッドを呼び出します。文字列は wchar_t * に変換されます。メソッドから返された値を返します。
void close ()	ライブラリを解放し、リソースを解放します。このメソッドを呼び出さない場合、外部ライブラリがもはや必要なくなったときに自動的に呼び出されます。

構文は次のようになります。

```
length = external ( "native1" ). invoke ( "stringLength", "Foo" );
```

これは、int string Length (char * str) として宣言された C 関数を前提としています。次のコードを使用することもできます。

```
External library = external ( "native" );  
seven = library. invoke ( "sum", 3, 4 );  
two = library. invoke ( "sum", 3, -1 );  
library. close ( );
```

これは、int 関数の int (int a, int b) として宣言された C 関数を前提としています。同じライブラリへの呼び出しをいくつか行う必要がある場合、後者の構文はより効率的です。

サポートされる引数のタイプ

メソッドは最大六つの引数を持つことができ、64 ビットプラットフォーム上の 64 ビット値である整数(またはネイティブポインタ) 値を(オプションで) 返すことができます。Java では、戻り値は常に long です。

全ての引数タイプがサポートされているわけではなく、いくつかの制限が適用されます。特に、浮動小数点(float) 引数は倍精度(double) 引数に変換されます。ネイティブ関数を浮動小数点(float) 引数で宣言してはいけません。浮動小数点タイプのデータを転送する必要がある場合には、サポートされている浮動小数点(float) 配列を代わりに使用できます。

TABLE 4-2: サポートされる引数のタイプ

JAVA タイプ	C/C++ タイプ	注釈
boolean	bool	
byte	char	
char	wchar_t	システムに依存。Linux では 32 ビット、MacOS では 16 ビット、Windows では 16 ビット。
short	short	

TABLE 4-2: サポートされる引数のタイプ

JAVA タイプ	C/C++ タイプ	注釈
int	int	
long	long long	
String	const char * or const wchar_t *	C 側で取得するバージョンは、使用する invoke インタフェースのバージョンによって異なります。文字列は null で終端されているため、変更しないでください。上記の wchar_t のタイプの注釈を比較してください。
float	double	引数は自動的に倍精度に拡張されます。
double	double	
boolean[]	bool *	
byte[]	char *	
short[]	short *	
int[]	int *	
int[][]	int **	
long[]	long long *	どのプラットフォームでも制限はありません。
String[]	char ** or wchar_t **	String[] 引数を使用すると、文字列を Java に戻すことができます。上記の String のタイプの注釈を比較してください。
float[]	float *	
double[]	double *	
double[][]	double **	

配列タイプの場合、外部関数は指定された配列の値を変更することができます。更新された値は、Java プログラムに戻されます。呼び出し先(callee)が転送された配列のサイズを変更することはできないことに注意してください。呼び出し元(caller)と呼び出し先(callee)は、追加の引数を使用するか、または大文字小文字を使用してサイズの伝達を行います。呼び出し先(callee)が割り当てられたバッファの外側のメモリを書き込んだり読み込んだりすると、プログラムの動作は特定されておらず、プログラム終了が突然起こる可能性があります。倍精度配列タイプの場合には、ポインタではなく値だけが変更されることに注意してください。

String []タイプの引数の場合、プログラムは配列のポインタのうちの一つを変更して、新しく null 終端文字列を指すようにすることができます。この文字列は Java に戻されます。一つの文字列の最大長は 65,535 文字です。大量のデータを転送するには、バイト配列を使用します。文字列配列内の対応する要素を 0 に設定すると、Java に null 値を返すことができます。

より複雑なタイプは、Java 側でバイト配列にシリアル化(直列化)する必要があります。バイト配列は、呼び出し先(callee)によってデシリアライズ(直列化復元)できます。

外部関数は、整数(int)または内部オブジェクト(void *)へのポインタを返すことができます。Java では、この値は長整数タイプ(long)値として返されます。これは、64 ビットのメモリアドレスを保持するのに十分な幅です。戻り値として文字列または倍精度が必要な場合は、長さ 1 の配列であり、外部関数によって埋め込むことができる正しいタイプを代わりに渡す必要があります。



アプリケーションの作成者として、適切なタイプの引数が正しい順序で指定されていることを確認する必要があります。チェックは行われずに、間違いを犯すと、簡単にプログラムがクラッシュしたり、定義されていない動作が起こることがあります。特に、整数引数と浮動小数点引数を扱う場合は、特別な注意が必要です。sum (int a, int b)として宣言された関数は、invoke ("sum", 1, 2)によって呼び出されなければならないのに対して、sum (double a, double b)として宣言された関数は、invoke ("sum", 1.0, 2.0)によって呼び出されなければなりません。

デバッグ

アプリケーションを操作するとき、ネイティブコードデバッガ(Microsoft®VisualStudio®など)を使用することができます。まず、デバッグシンボルでライブラリをビルドする必要があります。次に、このライブラリをアプリケーションにインポートし、デ

バグを COMSOL マルチフィジックスのプロセスにアタッチします。デバッグしたいネイティブ関数にブレークポイントを設定します。COMSOL マルチフィジックスでアプリケーションを実行またはテストします。ソフトウェアが外部ライブラリをロードすると、デバッグは、ブレークポイントをロードされるライブラリに配置しておいたソースコードと一致させて、関数が呼び出されたときに中断(ブレーク)させることができます。詳細については、ネイティブデバッグシステムのドキュメントを参照してください。

外部ライブラリを使用する場合の注意とヒント

- ネイティブコードからメモリにアクセスするときは、十分注意してください。アプリケーション全体のメモリが簡単に破損する可能性があり、定義外の動作やクラッシュの発生をきたします。
- アプリケーションでサポートするプラットフォームごとに個別のライブラリを提供する必要がありますが、特定のアプリケーションが COMSOL サーバーでサポートされている全てのプラットフォームをサポートする必要はありません。現在サポートされているプラットフォームは、AMD64 アーキテクチャを使用する全てのケースに於いて、Windows 64 ビット、Linux 64 ビット、および MacOS 64 ビットです。
- ほとんどのコンパイラが作成する共有ライブラリは、コンパイラとともに配布される別の共有ライブラリによって提供されるランタイム環境に依存します。所望のターゲットコンピュータに適切なランタイム環境がインストールされていることを確認する必要があります。
- 使用したいネイティブコードライブラリが複雑すぎて、単一の共有ライブラリ(DLL, so または dylib)に収まらない場合は、(カスタムインストーラを使用して)ターゲットコンピュータにライブラリを個別に配備する必要があります。そのときに、アプリケーションには、複雑なライブラリを呼び出す小さなラッパーライブラリが含まれています。技術的には、システム管理者権限が必要でないという条件では、複雑なライブラリをインストールするメソッドを使用して、複雑なライブラリのインストーラ全体を大規模でシンプルなライブラリに含めることが可能です。

外部関数の例

以下の手順では、ライブラリを作成およびインポートしてメソッドで呼び出す方法の例を示しています。

1 ライブラリの作成

以下のソースコードは、呼び出したい C++ で書かれた簡単な外部関数を定義しています。#ifdef ステートメントは、クロスプラットフォーム対応のソースコードを作成するためのものですが、通常、共有ライブラリを作成する際にはコンパイラ固有の設定が必要になるため、コンパイラのドキュメントを参照する必要があります。ヘッダーファイル test.h は次の通りです。

```
// test.h : Declares the exported function
#ifdef _MSC_VER
    #define TESTDLL_API __declspec ( dllexport )
#else
    #define TESTDLL_API __attribute__( ( __visibility__( "default" ) ) )
#endif extern "C" {
    TESTDLL_API int testSum( int a, int b );
}
```

関数定義 test.cpp は次のようになります。

```
// test.cpp : Defines the exported function for the library
#include "test.h"

// This is an example of an exported function.
TESTDLL_API int testSum ( int a, int b ) {
    return a + b ;
}
```

gcc コンパイラを使用して、コマンドシェルに次のように入力します。

```
gcc -shared -o test.so -fPIC test.cpp
```

Microsoft Visual Studio のようなグラフィカルな C++ビルド環境を使用している場合、おそらく上記のコードを DLL プロジェクトに挿入するのが最も簡単です。

2 ライブラリのインポート

外部 C ライブラリノードをタグ `native1` で追加し、ステップ 1 で作成ライブラリを適切なプラットフォームにインポートします。

3 メソッドの呼び出し

メソッドノードを追加し、次のコードを `Execute` メソッドのボディに挿入します。

```
long sum = external ( "native1" ). invoke ( "testSum", 1, 2 );  
alert ( "1 + 2 =" + sum );
```


フォームにボタンを追加し、実行するコマンドとして上記で追加したメソッドを選択します。ライブラリがビルドされているのと同じプラットフォームにアプリケーションが構築された場合は、**アプリケーションをテスト**をクリックしてメソッドをテストすることができます。それ以外の場合は、アプリケーションを保存し、正しいプラットフォーム上の COMSOL Multiphysics セッションで実行します。いずれの場合も、アプリケーションウィンドウのボタンを押すと、予定通りのダイアログが表示されるはずですが。

ファイルスキームとファイル処理

サーバーからクライアントへのファイル転送

COMSOL サーバーからクライアントアプリケーションにファイルを転送するには、**宣言**の下に**ファイル**ノードを追加しないで、以下の手順を実行してください。

- 例えば、`temp :/// output. data` にデータを書き込みます。
- `downloadtoclient` アクションの引数として `temp :/// output. data` を追加します。

`downloadtoclient` アクションの引数を定義するときには、**引数編集**ボタン () をクリックして**引数編集**のダイアログボックスを開きます (**宣言とコマンドシーケンスでの初期値と引数の編集**を参照)。ダイアログボックスでは、**ファイルスキーム**リストからその下のいずれかのファイルスキームを選択することができます。次に、**ファイル名**のフィールドにファイル名 (`output. data` など) を指定します。`embedded :///` ファイルスキームの場合、**アプリケーションファイルリソース選択**の下のリストから、アプリケーション内の既存のファイルを選択します。**OK** をクリックしてダイアログボックスを閉じると、**引数**フィールドに選択したファイルスキームとファイル名が入力されます。

ファイルスキーム

通常、クライアントファイルシステム上のファイルを参照するファイルパスが指定されているモデルまたはアプリケーションのどこでも、代わりに`<scheme> :/// ...` のようなスキームシンタックスを使用して、サーバー上の他の場所から取得しようとするファイルを参照することができます。次の表には、使用可能なスキームと、アプリケーションビルダーと COMSOL サーバーから実行するときのファイルの場所が示されています。

TABLE 4-3: ファイルスキーム

スキーム	参照先	アプリケーションビルダーのデフォルトパス	COMSOL サーバーのデフォルトパス
<code>embedded:///</code>	ファイルライブラリを使用してモデルに埋め込まれたファイル (読み取り専用)。	MPH ファイルの内部	MPH ファイルの内部
<code>upload:///</code>	ファイル宣言にアップロードされたファイル。	<code>user:///</code> または <code>temp:///</code> ターゲットによる。	<code>user:///</code> または <code>temp:///</code> ターゲットによる。
<code>temp:///</code>	ランダムなテンポラリディレクトリにあるファイル。起動されたアプリケーションインスタンスごとに一意。	<code>%TEMP%</code> または <code>/ tmp</code> のランダムサブディレクトリ	<code>.comsol / v53aserver / service / users / [username]</code> のランダムサブディレクトリ
<code>common:///</code>	全てのユーザーとアプリケーションで共有されるディレクトリ内のファイル。	<code>.comsol / v53a / applications / files / common</code> の場所	<code>.comsol / v53aserver / applications / files / common</code> の場所
<code>user:///</code>	現在のユーザーの全てのアプリケーションによって共有されるディレクトリ内のファイル。	<code>.comsol / v53a / applications / files / user</code> の場所	<code>.comsol / v53server / applications / files / user/[username]</code> の場所

temp スキームに保存されたファイルは、同じユーザーであっても、同じアプリケーションを複数回実行する間にずっとは維持されません。user スキームに保存されたファイルは、他のアプリケーションであってもずっと維持されて、同じユーザーがずっとアクセスすることができます。common スキームに保存されたファイルは、ずっと維持されて、全てのアプリケーションが全てのユーザーからアクセスできます。

次の環境設定を使用して、個別にデフォルトの場所を変更することができます。

TABLE 4-4: デフォルトの場所の環境設定

スキーム	アプリケーションビルダーの環境設定	COMSOL サーバーの環境設定
temp:///	ファイル>テンポラリファイル用フォルダ	ファイル>テンポラリファイル用フォルダ
common:///	ファイル>コモンファイル用フォルダ	ファイル>アプリケーションライブラリファイル用フォルダ
user:///	ファイル>ユーザーファイル用フォルダ	ファイル>アプリケーションユーザーファイル用フォルダ

URI シンタックス

ファイルスキーム文字列は、RFC 2396 (<http://www.ietf.org/rfc/rfc2396.txt>) で定義されている URI (Uniform Resource Identifier) です。ファイルスキームは、RFC 2396 で許可されているものに加えて、パスセグメント内の予約されていないいくつかの Unicode 文字を許可しています。つまり、ファイル名やディレクトリ名でスペースをエスケープする必要はありません。エスケープされたオクテットシーケンス (例えば、%20) は、UTF-8 としてデコードされます。現在、ファイルスキームは authority コンポーネントを使用していません。つまり、それらは常に三つのスラッシュを持つ <scheme> : /// <path> という形式です。

クライアントファイルシステムとのファイルの送受信

アプリケーションビルダーは、アプリケーションの実行時にクライアントファイルシステムとの間でファイルをやりとりするためのいくつかの方法を提供してくれています。ウェブクライアントの実行時とネイティブクライアントの実行時の両方で、ファイルを取得する次の方法があります。

- ファイルインポートフォームオブジェクト ([ファイルインポート](#)を参照) を使用して、ユーザーにファイルを要求します。ユーザーは、クライアントファイルシステム上のファイルをブラウズし、その後、サーバーにアップロードされ、アプリケーションおよびそのメソッドで利用可能となります。このメソッドは、例えば、実行時にユーザーから CAD ファイルまたは実験データを提供するために利用することができます。
- 任意のアプリケーションメソッドから fileOpen 関数を呼び出すことができます。それは、メソッド、モデル、またはアプリケーションに埋め込まれたサーバーからの任意のファイルを取り出し、クライアント上の関連アプリケーションを使用して開きます。このメソッドは、例えば、クライアント上で PDF ドキュメントを開く、またはクライアント上でモデルからエクスポートされたテキストファイルや画像を表示するために使用することができます。
- fileOpen に似た方法で、任意のアプリケーションメソッドから fileSaveAs 関数を呼び出すこともできます。サーバーから任意のファイルを受け取り、ユーザーがファイルを保存するためにクライアントの場所を参照することができる **名前を付けて保存** のダイアログボックスを表示します。このメソッドは、ウェブブラウザー内のリンクからファイルをダウンロードする場合と同じです。
- **名前を付けてアプリケーションを保存** と **サーバー上でアプリケーションを別名で保存** のコマンドは、フォームのボタンやメニュー項目のコマンドシーケンスツリーで使用することができます。これらのコマンドは、ユーザーがアプリケーション全体が保存されるクライアントパスを指定できる **名前を付けて保存** のダイアログボックスを表示します。アプリケーションが COMSOL サーバーで実行されている場合、これらのコマンドは現在の状態を COMSOL サーバーのアプリケーションライブラリに新規アプリケーションとして保存します。

メソッドの作成

以下のセクションでは、コードを記述してメソッドを作成するためのツールについて説明しています。

- シンタックスハイライトとコメント
- コード補完とツールチップのヘルプ
- コードの折り畳み
- 言語要素の追加
- モデル表現の追加
- モデルコードとフォームオブジェクトの追加
- ソースコードがマップされるノードへの移動
- コードの記録
- ショートカットの使用
- 他のメソッドの直接的な呼び出し
- 変数宣言で定義されたプロパティの使用
- テキストの検索
- 字下げと空白フォーマット
- ブレースマッチング

COMSOL マルチフィジックスで利用可能なモデルオブジェクトで操作するための組み込みメソッドについては、*COMSOL Multiphysics Programming Reference Manual* を参照してください（フィジックスインタフェースは除かれた内容が掲載されています）。

シンタックスハイライトとコメント

メソッドのコード内の異なる言語要素は、異なるスタイルで表示されます。例えば、次のコードスニペットを考えてみましょう。


```
Code3 X
1 for (int i = 0; i < 10; i++) {
2   model.result().numerical("eval").set("expr", "var" + i);
3   model.result().numerical("eval").appendResult();
4 }
```

このコードスニペットには、三つのスタイルが含まれています。

- キーワード（for, int など）は、青色の太字のフォントで表示されています。
- 文字列リテラルは赤色のフォントで表示されています。
- 残りのコードは黒色のフォントで表示されています。

環境設定のダイアログボックスで、シンタックスハイライトのテーマを設定することができます。**ファイル > 環境設定 > メソッド** と選択して、メソッド固有の環境設定を表示します。**シンタックスをハイライト** の下のテーマリストには、**モダン**（デフォルト）と**クラシック** の二つの定義済みテーマが含まれています。**ユーザー定義** を選択すると、キーワード、数字、文字列、言語要素など、コードの個々の部分に色を割り当てることのできるシンタックスハイライトモードを定義することができます。各コードごとのカラー選択ボタンをクリックして開かれるカラーパレットからカラーを選択します。**カスタムカラー** のボタンをクリックした場合は、RGB 値を使用して定義された色が追加されます。

コメントの追加

フォームエディターで作業するコードにコメントを追加するには、コメントの前に // を付けます。ショートカット Ctrl + 7 を使用するか、メソッドウィンドウで右クリックして**コメントを切替え**  を選択することで、選択した行のコメントをオン/オフに切り替えることができます。

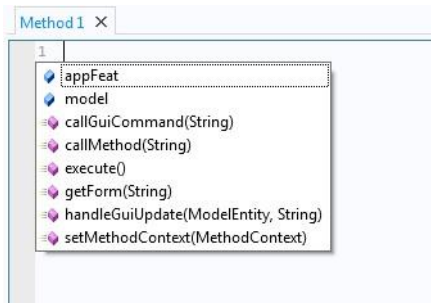
コード補完とツールチップのヘルプ

コード補完

メソッドエディタウィンドウで Java コードを入力するときに、部分的なコードの補完候補を要求することができます。可能な補完候補のリストは、開かれる別の補完リストに表示されます。状況によっては、リスト内で項目が選択されたときに、その詳細情報がさらに別のウィンドウに表示される場合もあります。コード補完は、キーボードショートカット Ctrl + Space (または Ctrl + /) を使用していつでも要求することができます。また、ピリオドを入力した時点で自動的に表示されます。これは、通常、フィールドにアクセスしているときに使用可能なメソッドの中から選択できる必要があるからです。

アプリケーションビルダーは、以下のタイプの補完候補をサポートしています。

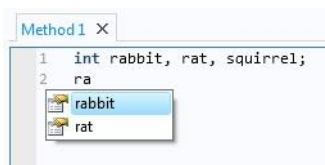
新規ステートメント： 単純な例では、下図に示すように何も入力せずに Ctrl + Space を押すことです。



補完リストには、メンバーフィールド (例えば appFeat や model など) と、呼び出すことができるメンバーメソッド (例えば callGuiCommand など) が含まれています。以下の二つの方法で補完を選択することができます。

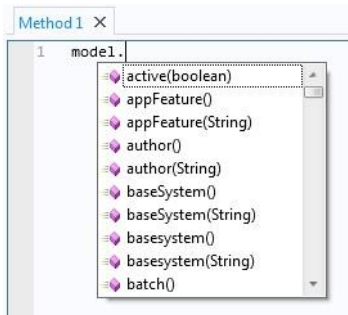
- 矢印キーを使用してリスト内の項目を選択し、Tab キーを押して選択を確定します。
- テキストを入力します。リストは自動的に更新され、そのテキストで始まる補完だけがリストに掲載されます。例えば、appF と入力すると、appFeat だけがリストに残ります。Tab キーを押して選択を確定します。

部分ステートメント： 変数、フィールド、またはメソッド名の先頭を入力し、Ctrl + Space を押すと、提案される補完が表示されます。

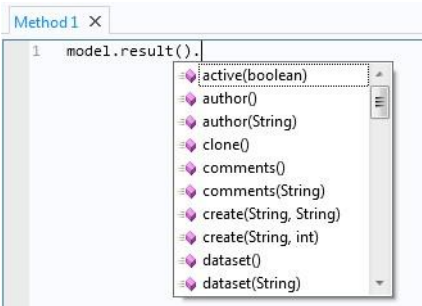


上図では、プレフィックス ra と一致する変数のみが表示されています。この例は、ローカル変数が補完候補にも現れることを示しています。

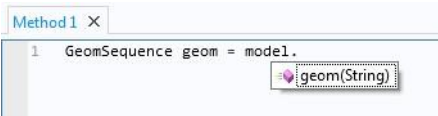
クラスメンバー：モデルオブジェクトの既知の Java クラスの特定したい変数または式を入力する場合には、クラス内のメンバーの補完候補を下図のように取得することができます。



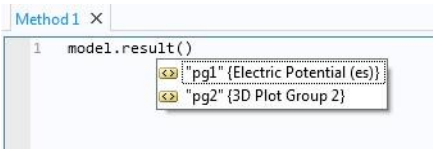
この補完は、下図に示すように、一連の呼び出しでも機能します。



代入と宣言のクラスメンバー：代入と変数宣言では、予想されるタイプを使用して補完のリストをフィルタリングすることができます。代入ステートメントの左側に宣言されているのと同じタイプを持つ補完だけが、表示されています。

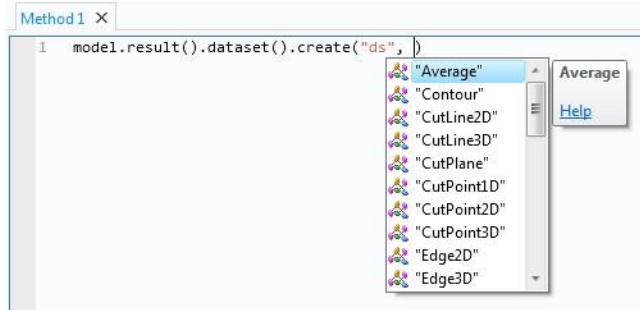


モデルオブジェクト内のタグ：補完のための式が埋め込みモデルの具体的なエンティティを特定させたい場合、補間はモデルオブジェクト API 内のメソッドに対して使用することができ、下図に示すように、リストのタグには最初の引数が取られます。

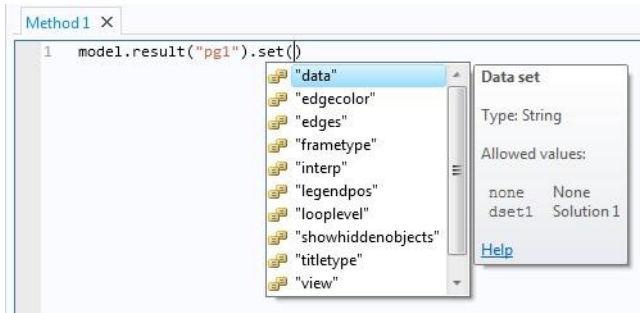


上記のリストは、埋め込みモデルに二つのプロットグループ (pg1 と pg2) が含まれているアプリケーションによるものです。補完リストには、タグと対応するプロットグループの名前が含まれています。タグだけをコードに転送するには、リスト内の補完を選択して Tab キーを押します。

新規エンティティを作成するときのモデルオブジェクトのタイプ： 補完のための式が埋め込みモデルの具体的なエンティティを特定させたい場合、補間は、新しいノードを作成するときモデルオブジェクトのオペレーションタイプに対して使用することができます。



プロパティ名： 補完のための式が埋め込みモデルの具体的なエンティティを特定させたい場合、補間は、get および set 操作内の最初の引数として使用されるプロパティ名に対して使用することができます。これらは、モデルオブジェクトのほとんどのフィーチャーで使用することができます。



上記の例は、埋め込みモデルのplotグループ pg1 で使用できるプロパティを示しています。リスト内の項目を選択すると、そのプロパティの説明とそのデータタイプが表示されます。また、利用可能な場合は、ドキュメントへのリンクと許可された値のセットが表示されることもあります。許容された値のセットが文字列の有限の配列である場合には、その許容された値が表示されます。

プロパティ値： 補完のための式が埋め込みモデルの具体的なエンティティを特定させたい場合、補間は、モデルオブジェクトのほとんどのフィーチャーで使用可能な set 操作内の第 2 引数として使用されるプロパティ値に対して使用することができます。



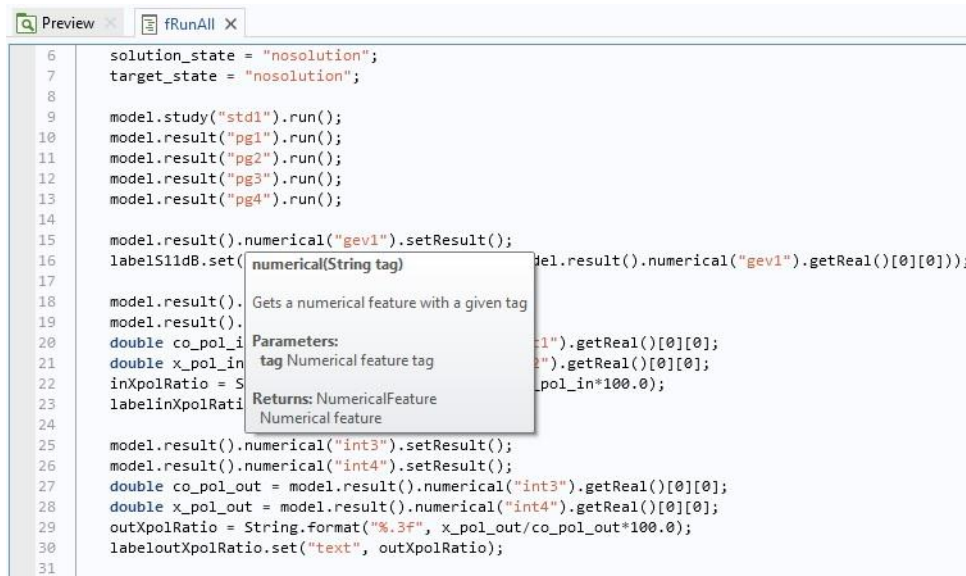
リスト内の項目を選択すると、プロパティ値の説明と、利用可能な場合はドキュメントへのリンクが表示されます。

メソッドエディターのコードのためのツールチップ

メソッドのコードのさまざまな部分にカーソルを移動すると、コードのその部分に関する情報を提供するツールチップが表示されます。コードのさまざまな部分について、以下のツールチップが表示されます。

プロパティ名

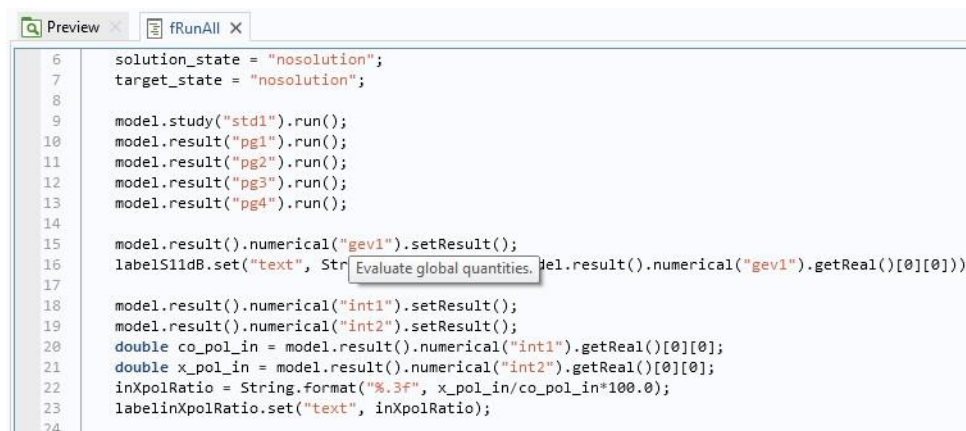
モデルまたはアプリケーションオブジェクトのプロパティのプロパティ名にマウスポインタを置くと、コード補完が使用されたときのプロパティの情報と同様のプロパティに関する情報が表示されます。例えば、`model.result().numerical("gev1").setResult();` の中で `numerical` をホバーリングすると、下図のツールチップが表示されます。



```
6 solution_state = "nosolution";
7 target_state = "nosolution";
8
9 model.study("std1").run();
10 model.result("pg1").run();
11 model.result("pg2").run();
12 model.result("pg3").run();
13 model.result("pg4").run();
14
15 model.result().numerical("gev1").setResult();
16 labelS11dB.set("numerical(String tag) model.result().numerical("gev1").getReal()[0][0]);
17
18 model.result().
19 model.result().
20 double co_pol_in = model.result().numerical("int1").getReal()[0][0];
21 double x_pol_in = model.result().numerical("int2").getReal()[0][0];
22 inXpolRatio = String.format("%.3f", x_pol_in/co_pol_in*100.0);
23 labelinXpolRatio
24
25 model.result().numerical("int3").setResult();
26 model.result().numerical("int4").setResult();
27 double co_pol_out = model.result().numerical("int3").getReal()[0][0];
28 double x_pol_out = model.result().numerical("int4").getReal()[0][0];
29 outXpolRatio = String.format("%.3f", x_pol_out/co_pol_out*100.0);
30 labeloutXpolRatio.set("text", outXpolRatio);
31
```

モデルエンティティ

この場合、同じプロパティのモデルエンティティ名 `"gev1"` にマウスを合わせると、モデルエンティティの目的に関する情報が記載された下図のツールチップが表示されます。



```
6 solution_state = "nosolution";
7 target_state = "nosolution";
8
9 model.study("std1").run();
10 model.result("pg1").run();
11 model.result("pg2").run();
12 model.result("pg3").run();
13 model.result("pg4").run();
14
15 model.result().numerical("gev1").setResult();
16 labelS11dB.set("text", String.format("%.3f", model.result().numerical("gev1").getReal()[0][0]));
17
18 model.result().numerical("int1").setResult();
19 model.result().numerical("int2").setResult();
20 double co_pol_in = model.result().numerical("int1").getReal()[0][0];
21 double x_pol_in = model.result().numerical("int2").getReal()[0][0];
22 inXpolRatio = String.format("%.3f", x_pol_in/co_pol_in*100.0);
23 labelinXpolRatio.set("text", inXpolRatio);
24
```

宣言

アプリケーションビルダーツリーの宣言で定義された変数の場合、その上にカーソルを置くと変数の説明がツールチップとして表示されます。下図の場合、それは文字列変数 inXpolRatio です。



```
6 solution_state = "nosolution";
7 target_state = "nosolution";
8
9 model.study("std1").run();
10 model.result("pg1").run();
11 model.result("pg2").run();
12 model.result("pg3").run();
13 model.result("pg4").run();
14
15 model.result().numerical("gev1").setResult();
16 labelS11dB.set("text", String.format("%.3f", model.result().numerical("gev1").getReal()[0][0]));
17
18 model.result().numerical("int1").setResult();
19 model.result().numerical("int2").setResult();
20 double co_pol_in = model.result().numerical("int1").getReal()[0][0];
21 double x_pol_in = model.result().numerical("int2").getReal()[0][0];
22 inXpolRatio = String.format("%.3f", x_pol_in/co_pol_in*100.0);
23 Input waveguide feed cross polarization ratio percentage
24
25 model.result().numerical("int3").setResult();
26 model.result().numerical("int4").setResult();
27 double co_pol_out = model.result().numerical("int3").getReal()[0][0];
28 double x_pol_out = model.result().numerical("int4").getReal()[0][0];
29 outXpolRatio = String.format("%.3f", x_pol_out/co_pol_out*100.0);
30 labeloutXpolRatio.set("text", outXpolRatio);
```

ショートカット

宣言で定義されたフォームオブジェクトへのショートカットについては、説明とそれに対応する Java コードがツールチップに表示されます。下図は、labelinXpolRatio という名前のショートカットの場合です。



```
6 solution_state = "nosolution";
7 target_state = "nosolution";
8
9 model.study("std1").run();
10 model.result("pg1").run();
11 model.result("pg2").run();
12 model.result("pg3").run();
13 model.result("pg4").run();
14
15 model.result().numerical("gev1").setResult();
16 labelS11dB.set("text", String.format("%.3f", model.result().numerical("gev1").getReal()[0][0]));
17
18 model.result().numerical("int1").setResult();
19 model.result().numerical("int2").setResult();
20 double co_pol_in = model.result().numerical("int1").getReal()[0][0];
21 double x_pol_in = model.result().numerical("int2").getReal()[0][0];
22 inXpolRatio = String.format("%.3f", x_pol_in/co_pol_in*100.0);
23 labelinXpolRatio.set("text", inXpolRatio);
24 Text label
25 Shortcut to Text label
26 app.form("mainForm").formObject("labelinXpolRatio");
27 double co_pol_out = model.result().numerical("int3").getReal()[0][0];
28 double x_pol_out = model.result().numerical("int4").getReal()[0][0];
29 outXpolRatio = String.format("%.3f", x_pol_out/co_pol_out*100.0);
30 labeloutXpolRatio.set("text", outXpolRatio);
31
```

コード生成と補完のための環境設定

環境設定のダイアログボックスのメソッドセクションでは、コードの生成と補完のオプションを指定することができます。

デフォルトでは、括弧を自動的に閉じるのチェックボックスが選択されています。これによって、{、[、または(の文字を入力すると、対応する閉じ括弧が自動的に挿入されます。このチェックボックスをクリアすると、閉じ括弧の追加が無効になります。

with ステートメントを使ってコンパクトコード生成のチェックボックスがデフォルトで選択されています。with ステートメントを使用して生成されたコードでは、set()、setIndex()、getString()などの呼び出しで使用するターゲットを設定しており、コードをよりコンパクトにしています。with ステートメントを使用せずに完全なコードを使用する場合は、チェックボックスをクリアします。

コードの折り畳み

デフォルトでは、メソッドのエディターウィンドウはコードの折り畳みが可能です。これにより、エディターでコードのセクションを非表示にしたり表示したりすることを選択できます。コードの折り畳みを使用すると、指定された時間に関連するコードのセクションのみを表示することで、多くのコードを含むメソッドを管理することができます。

コードセグメントを非表示にするには-(マイナス)ボタンをクリックし、コードセグメントを表示するには+(プラス)ボタンをクリックします。折り畳まれているとき、その領域のコードは省略記号(...)の付いたボックスとして置かれます。そこにカーソルを置くと、非表示のコードがツールチップとして表示されます。



```
16 L1=Math.max(L1,minlength);
17 int k=2;
18 while(k<MAXITERATIONS && Math.abs(f1)>fqtol ){
19     f2 = f1;
20     fq = frequency(L1);
21     f1 = fq-targetfq;
22     carry = L1;
23     L1=L1-f1*((L1-L2)/(f1-f2));
24     L2 = carry;
25     L1=Math.max(L1,minlength);
26     k=k+1;
27     setProgress(k*100/MAXITERATIONS);
28 }
29 L1=Math.round(L1*100)/100.00; // we won't get more than 2 decimal accuracy (mesh limits the accuracy rather than the secant method)
30 model.param().set("L1",L1);
31 if(Math.abs(f1)>fqtol) {
32     error("Computation terminated after "+toString(MAXITERATIONS)+" iterations. Frequency diff: "+toString(Math.abs(f1)));
33 }
34 }
35 else { // "Find prong length" check box is cleared in main form
36     startProgress(false, "Computing frequency.");
37     setProgress(25);
38     fq = frequency(model.param().evaluate("L1"));
39     setProgress(75);
40 }
```

Figure 4-2: 一部のコードセグメントを折り畳む前のメソッドエディターウィンドウ



```
16 L1=Math.max(L1,minlength);
17 int k=2;
18 while(k<MAXITERATIONS && Math.abs(f1)>fqtol ){...}
19 L1=Math.round(L1*100)/100.00; // we won't get more than 2 decimal accuracy (mesh limits the accuracy rather than the secant method)
20 model.param().set("L1",L1);
21 if(Math.abs(f1)>fqtol) {...}
22 }
23 else {...}
24 }
25 with(model.result("pg1"));
26 set("looplevel", new String[]{"?"}); // The first real eigenfrequency always the 7th computed eigenfrequency in solid mechanics
27 endwith();
28 useGraphics(model.result("pg1"), "graphics1");
29 zoomExtents("graphics1");
30 setProgress(100);
31 play_sound();
32 closeProgress();
```

Figure 4-3: いくつかのコードセグメントを折り畳んだ後の同じメソッド

コードの折り畳みをオフにするには、**環境設定**ダイアログボックスを開きます。**メソッド**のページで、**メソッドエディター**の下の**コードフォールディングを有効化**のチェックボックスをクリアします。

言語要素の追加

言語要素ウィンドウには、メソッドに挿入可能な配列や for ブロックの作成などといった、よく使う操作を実行するための共通言語の構成とテンプレートが含まれています。また、文字列ユーティリティ、変数の宣言と変換、進捗情報とエラーメッセージ、ファイル操作、ユーザーインターフェイスコンポーネント、電子メールの送信コード、およびその他の有用な言語要素を追加することもできます。言語要素を挿入するには、それをダブルクリックするか、右クリックして**テンプレートを挿入** (📄) を選択します。最上部の検索フィールドを使用して、利用可能な構成とテンプレートのリストをフィルタリングすることができます。

アクションによっては、テンプレート内に現在のエディター選択を使用するアクションがあります。— 例えば、現在の選択を**複数行コメント**のテンプレートに入れるなど。変更するのが最も好ましいテンプレートの部分が選択されるか、あるいはカーソルがそこに配置されます。

モデル表現の追加

モデル表現ウィンドウには COMSOL マルチフィジックスの表現が含まれており、それをメソッドエディターウィンドウに挿入して、メソッド内の結果や方程式のフィードバックに使用することができます。例えば、`model.result(). numerical("gev1"). set("expr", "x")` に x (x 座標) を挿入することができます。モデル表現をダブルクリックして挿入するか、右クリックして**式を挿入**を選択します。モデル表現は、メソッドエディターウィンドウにあるカーソルの箇所に挿入されます。

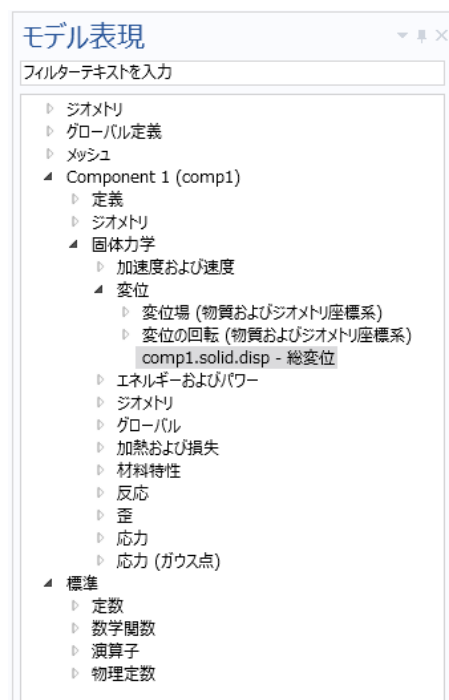


Figure 4-4: 利用可能なモデル表現の全体ビュー

ウィンドウの最上部にある検索フィールドを使用して、モデル表現のリストをフィルタリングすることができます。例えば、次の図は、si が含まれるモデル表現のリストを示しています。

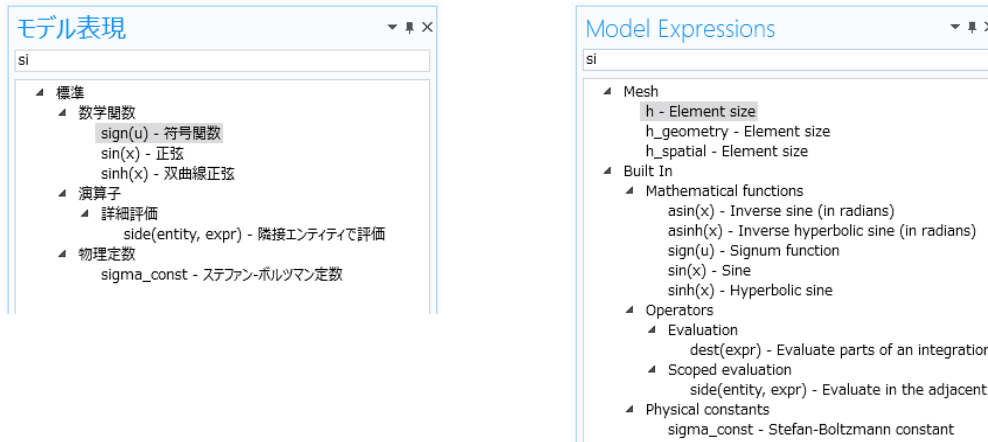


Figure 4-5: 使用可能なモデル表現のフィルタ表示

(注) 英語の言語設定の COMSOL では、同じモデルのフィルタリング結果は上図のようになります。この場合は、説明文などに対しても si が含まれるものが全て掲載されます。

挿入された表現と関数は、挿入ポイントが既に文字列中である場合を除いて、引用符で囲まれた文字列です (例えば "sigma_const")。文字列形式となっている理由は、挿入された表現がモデル表現であってコードではないことを明確にするためです。

モデルコードとフォームオブジェクトの追加

エディターツールウィンドウには、よく使うモデル操作 (例えば、フィーチャのデフォルト以外のプロパティを全て設定するなど) のコードと、モデル操作に対応する適用可能なフォームオブジェクトを作成するためのオプションが用意されています。

エディターツール

メソッドエディターで作業する場合、メソッドエディターウィンドウのカーソル位置にコードを挿入することができます。エディターツールウィンドウのツリーノードを右クリックし、モデル内のそのノードに適用可能なアクションから選択するか、ツリーの下に対応するボタンの一つをクリックします。あるいは、ダブルクリックするか Enter キーを押すと、ツリーノードを右クリックしたときに表示されるコンテキストメニューの一番上にあるアクション項目を挿入することができます。

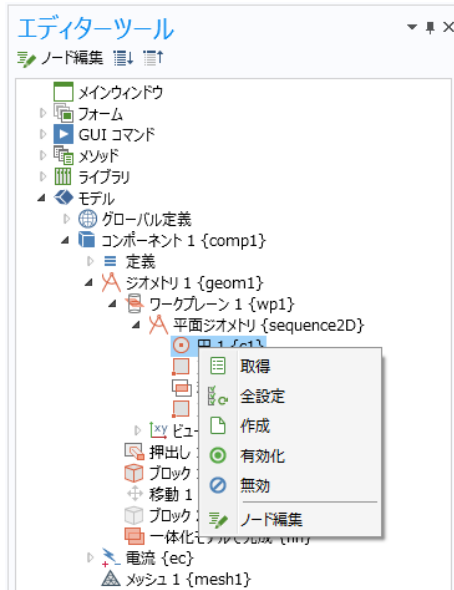


Figure 4-6: 現在のメソッドにコードを挿入するには、コンテキストメニューを使用します。

メソッドエディターで作業している場合は、次のアクションを使用することができます（全てのアクションが全てのノードに適用可能ではありません）。

- **取得** (📄): パラメータ、プロパティ、またはファイル名の値を取得するためのコード（例えば `model.param().get("w")` または `model.component("comp1").geom("geom1").feature("r1").getDouble("rot")`）を挿入、またはフィーチャへの参照（例えば `model.geom("geom1").feature("r1")`）を挿入します。
- **設定** (🔧): パラメータ、プロパティ、またはファイル名の値を設定するためのコード（例えば `model.param().set("w", 0)`；または `model.component("comp1").geom("geom1").feature("r1").set("rot", "0")`；）を挿入します。
- **全設定** (🔧): フィーチャのデフォルト以外のプロパティを全て設定するためのコードを挿入します。
- **作成** (📄): フィーチャを作成するためのコード（例えば `model.component("comp1").geom("geom1").create("r1", "Rectangle")`；）を挿入します。
- **実行** (▶): フィーチャを実行するためのコード（例えば `model.sol("sol1").run()`；）を挿入します。
- **有効化** (🟢) または **無効** (🔴): フィーチャを有効あるいは無効にするためのコード（例えばフィーチャを有効にするには `model.component("comp1").geom("geom1").feature("r1").active(true)`；）を挿入します。
- **ノード編集** (✎): メインのデスクトップのアプリケーションビルダーウィンドウでモデルブランチの下にあるノードを選択すると、その設定ウィンドウが開かれてそこで編集することができます。また、エディターツールウィンドウの最上部にあるノード編集ボタン(✎)をクリックすることもできます。

フォームエディターツール

フォームエディターで作業する場合、ツリーで選択した項目のアクションや表示に対応するフォームオブジェクトを挿入することができます。ツリーノードを右クリックし、個別のフォームオブジェクトまたは複数のフォームオブジェクトの組み合わせからなる項目の中から選択するか、ツリーの下にある対応するボタンの一つをクリックします。あるいは、ダブルクリックすると、ツリーノードを右クリックしたときに表示されるコンテキストメニューの一番上にあるアクション項目を挿入することができます。

フォームエディターで作業している場合は、次のアクションを使用することができます（全てのアクションが全てのノードに適用可能ではありません）。

- **入力** (I23) : テキストラベル、入力フィールド、および (該当する場合は) 単位オブジェクトを挿入して、その名前、およびモデルパラメータや宣言された変数のための入力フィールドを表示します。
- **出力** (I.23) : テキストラベルとデータ表示オブジェクトを挿入して、名前、およびモデルパラメータや変数、あるいは宣言ノードの下の変数の値を表示します。
- **グラフィックス** (I23) : プロットグループ、ジオメトリ、またはメッシュ用のグラフィックスオブジェクトを挿入します。
- **ボタン** (OK) : ファイルやビューコマンド用、あるいは、例えばレポートを作成するためのボタンオブジェクトを挿入します。

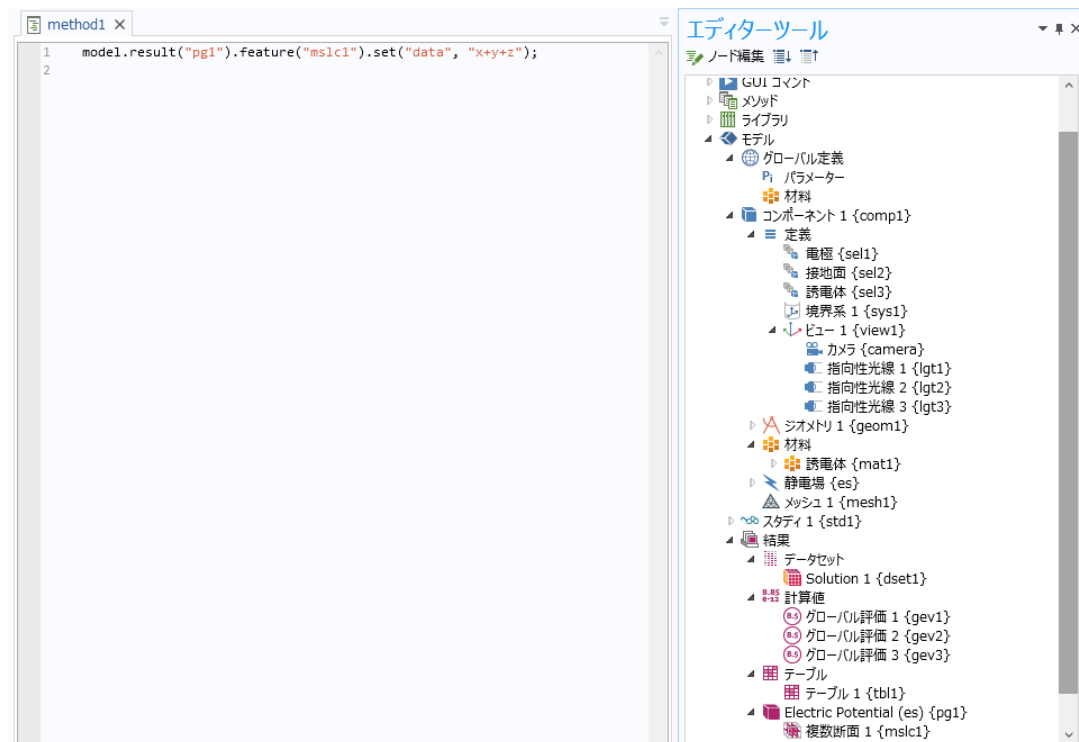
ソースコードがマップされるノードへの移動

メソッドリボンツールバーのノードへのボタン (I23) をクリックすると、カーソルを置いているソースコードが埋め込みモデルのエンティティにマッピングされます。可能であれば、対応するエンティティがエディターツールウィンドウに表示されます。このウィンドウは、まだ開いていなければ自動的に開かれます。また、埋め込みモデルを参照しているコードの一部にカーソルを置いている場合は、コードを右クリックしてノードへ (I23) を選択することもできます。

例えば、下図のコードを入力するとします。

```
method1 x
1 model.result("pg1").feature("mslc1").set("data", "x+y+z");
2
```

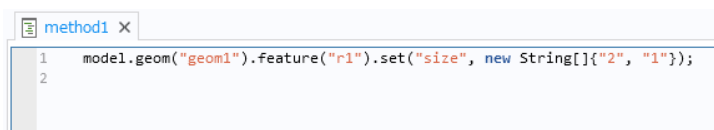
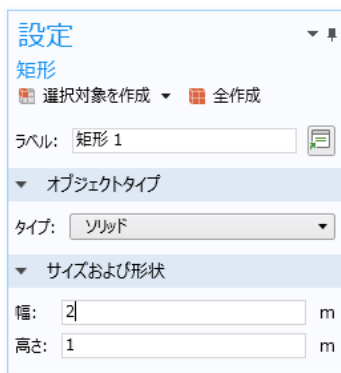
feature 上のどこかにカーソルを置き、ノードへをクリックします。すると、下図に示すように、エディターツールウィンドウが、該当するノード 複数断面 1 (Multislice 1) が選択された状態で開かれます。



result 上のどこかにカーソルを移動してノードへをクリックすると、エディターツールウィンドウ内で、代わりに Electric Potential のプロットグループが選択されます。

コードの記録

この機能によって、埋め込みモデルの変更に基いてコードを挿入することができます。アクションを挿入したい場所にカーソルを置きます。コードを記録(●)をクリックし、モデルビルダーウィンドウでアクションを実行してから、記録停止(■)をクリックします。記録されたコードは、メソッドエディターウィンドウに直接表示されます。例えば、矩形のジオメトリオブジェクトのサイズ設定を変更すると、下図のように、そのアクションの記録コードがメソッドエディターウィンドウに表示されます。



コードを記録しているとき、メソッドノード上の緑(注)の丸印は、記録されているコードが表示されるメソッドを示しており、そのメソッドのメソッドエディターウィンドウは赤い枠で囲まれた状態となります。

(注) 原文を修正しております。

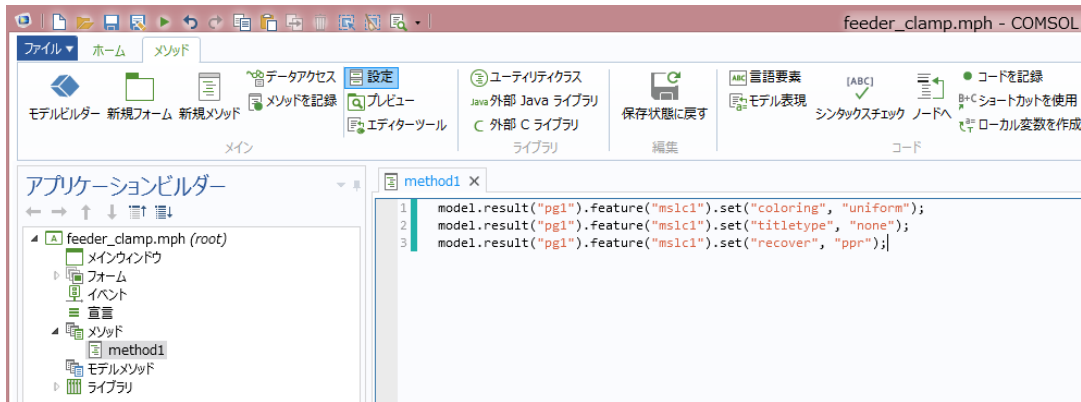
新規メソッドの記録

便利な機能として、ホームリボンツールバーに新規メソッドを作成して記録を開始するアクションもあります。メソッドを記録(●)をクリックし、メソッドを記録のダイアログボックスで、名前フィールドに新しいメソッドの名前を入力し、メソッドタイプリストからメソッドのタイプ — アプリケーションメソッド(デフォルト)またはモデルメソッド — を選択します。記録したいアクションを実行し、記録停止をクリックします。基本的に、メソッドを記録するためのコードは必要ありません。メソッドエディターウィンドウが開いている場合は、上記のコードを記録を使用する場合と同様に、記録されたコードが直接表示されます。新規メソッドが記録されている際、COMSOL デスクトップのモデルビルダーウィンドウとアプリケーションビルダーウィンドウが赤い枠で囲まれ、コード記録がアクティブになっていることが示されます。

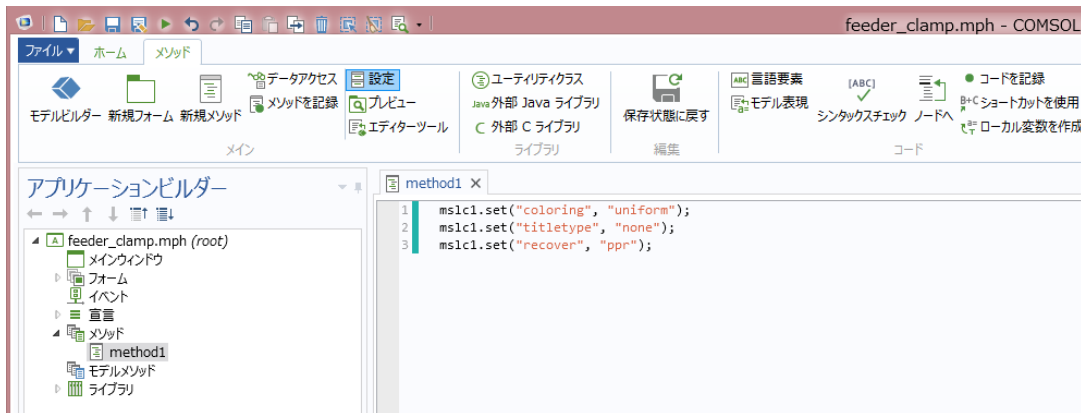
ショートカットの使用

メソッドリボンツールバーのショートカットを使用(B+C)をクリックする(または Ctrl + K キーを押す)と、メソッドエディターはカーソルでのソースコードをチェックします。特定のフォームの表現に対応する場合は、ショートカットを使用のダイアログボックスが表示されます。そこで、名前フィールドにショートカットの名前を入力します。OK をクリックすると、そのショートカットがショートカットノードの設定ウィンドウのリストに追加され、ソースコードがそのショートカットで表されるコードに置き換えられます。カーソルで選択したソースコードに複数の可能な表現がある場合には、選択表現のダイアログボックスが表示され、そこで、抽出リストから抽出する部分を選択することができます。表現は、現在埋め込みモデルに含まれているモデルエンティティに対応する必要があります。

例えば、下図のコードを入力して、カーソルが feature の第一オカレンス(構成要素)に置かれるようにします。



ショートカットを使用のボタンをクリックすると、ソースコードが下図のように変換されます。




ソースコードの feature 部分に対応する表現は、Results/pg1/mslc1 として定義されたショートカット mslc1 に置き換えられています。

カーソルがサポートされているタイプの表現に置かれていないとき、**ショートカットを使用**のボタンをクリックするとエラーメッセージが表示されます。サポートされる表現形式は、通常、モデルオブジェクト API が使用されるときに発生します。例えば、以下に示すオブジェクトのメソッド呼び出しのシーケンスの場合です。

```
<variable name> [ . <method name> ( <method arguments (optional)> ) ]
```

ローカル変数とそのタイプ宣言の作成

ローカル変数を宣言した際にそのタイプがわからない場合には、**ローカル変数を作成**のボタン() (または Ctrl + 1) をクリックして、正しいタイプの宣言を挿入することができます。また、ローカル変数もない場合には、タイプの宣言と共に固有のローカル変数が作成されます。例えば、次のように入力したとします。

```
x = model.component ( "comp1" ). geom ( ) ;
```

その後、**ローカル変数を作成**のボタンをクリックすると、変数のタイプ宣言が追加されます。

```
GeomList x = model.component ( "comp1" ). geom ( ) ;
```

単に model.component ("comp1"). geom () だけを追加した場合は、一意のローカル変数名も追加されます。

```
GeomList var3 = model.component ( "comp1" ). geom ( ) ;
```

他のメソッドの直接的な呼び出し

アプリケーションにメソッド method1 と method2 が含まれているとします。メソッドを呼び出すには、Java コードで直接その名前を使用します。これは、スーパークラスの ApplicationMethod (以下の例では method16) で宣言されている callMethod () の呼び出しと同じです。


```
package builder ;

import com. comsol. api. * ;
import com. comsol. model. * ;

public class method16 extends ApplicationMethod {

    public void execute () {
        // The following two lines are equivalent:
        method2 () ;
        callMethod ( "method2" ) ;
    }

}
```

他のメソッドのエディターウィンドウを開くには、コード内のメソッド名 (例えば method2) を Ctrl + Alt + ダブルクリックするか、右クリックして **メソッドへ** () を選択します。

変数宣言で定義されたプロパティの使用

宣言の下にある各ノードはデータタイプ (文字列、ブーリアンなど) に対応していますが、そのノードに対応した変数を Java コードで使用するために利用できます。いくつかの例外を除けば、通常の Java 変数のように動作します。例えば、宣言の下のノードに文字列プロパティ foo の宣言があるとします。次のコードは、それに値を割り当てています (この例では method16)。

```
package builder ;

import com. comsol. api. * ;
import com. comsol. model. * ;

public class method16 extends ApplicationMethod {

    public void execute () {
        foo = "bar" ;
    }

}
```

テキストの検索

Ctrl + F キーを押すと、**検索**ツールが開かれます。そこで、テキストを検索し、必要に応じてメソッド内で置換することができます。**検索**ツールでは、**全てのタブ**をクリックして、ユーザーインターフェイスコンポーネント、モデルエンティティタグ、操作識別子、ラベルなど、アプリケーション全体を検索することができます。例えば、テキスト Label の検索では全ての出現箇所を検索されるのに加え、メソッドを検索する場合でも全てのローカルメソッドを含む全てのメソッドを検索されます。**メソッド**のタブをクリックすると、メソッド内の文字列を検索し、オプションで置換することもできます。

また、**置換**のテキストフィールドに検索文字列を置き換えるための文字列を入力することもできます。

メソッドの中で検索の下では、現在のメソッドでのみ検索する場合は**現設定**をクリックし、全てのメソッドを検索する場合は**全て**をクリックします。**方向**の下で、**前方**または**後退**をクリックして、メソッドの検索方向を制御します。**大文字**、**小文字**を区別の


チェックボックスを選択すると、大文字と小文字を区別して検索することができます（検索文字列は大文字と小文字を含めて正確に一致する必要があります）。

次の検索文字列の該当文字列を検索するには、**次を検索**ボタンをクリックします。その右側にある**置換**ボタンをクリックすると、それを**置換**のテキストフィールドの文字列に置き換えます。**全てを置換**ボタンをクリックすると、検索文字列の全ての該当文字列を検索し、**置換**のテキストフィールドの文字列に置き換えます。

全て検索ボタンをクリックすると、検索が開始されて検索結果が**検索結果**ウィンドウに表示されます。ここで、各検索文字列が1行ごとに表示されます。行の一つをダブルクリックすると、その行が示す発生したメソッド内の検索結果をハイライト表示します。**メソッド**の列には検索文字列が表示されているメソッド名が表示され、**ライン**の列にはその検索文字列が表示されているメソッドの行番号が表示され、**テキスト**の列には検索文字列を含んだその行のテキストそのものが表示されます。

検索結果ウィンドウの左上隅にある**リフレッシュ**ボタン(🔄)をクリックすると、検索操作または検索置換操作を再度実行し、**検索結果**ウィンドウの内容を更新します。

字下げと空白フォーマット

コードが正しい字下げと空白フォーマットを使用するようにメソッドのコードを書式設定するには、Tab キーを押すか、メソッドエディターを右クリックして**字下げとフォーマット**()を選択します。空白の書式設定と挿入は、メソッドのコードの選択している部分、またはコードが選択されていない場合は現在の行に適用されます。デフォルトでは、字下げと空白フォーマットは、キーボードのフォーカスがメソッドエディターウィンドウを離れると自動的に行われます。**環境設定**ダイアログボックスの**メソッド**のページで**自動で字下げとフォーマット**のチェックボックスを使用して、メソッドエディターウィンドウを離れるときに自動で字下げとフォーマットを適用させるかどうかを指定することができます。このフォーマットは、モデルビルダーで記録できる自動生成コードに使用されているものと同じです。

以下のリストには、最もよく使う空白の規則を示しています。

- 左括弧の後または右括弧の前にスペースは入れません。
- 関数名と左括弧の間、または with と左括弧の間にスペースは入れません。
- if、while、for、catch などのキーワードの後には、一つのスペースが必要です。
- =、==、<=、*、+、%、&&、および||のような演算子の前後には、各一つのスペースが必要です。
- セミコロン文字の後には、一つのスペースが必要です。しかし、前にスペースは入れません。
- 少なくとも一つの空白でない文字が含まれている行には、その末尾に空白は入れません。

空白の書式設定を使わずに現在選択されている行を字下げするには、Shift + Tab キーを押します。

コードの書式設定に関係なく字下げを増減するには、Ctrl + Alt + i を押して字下げを増やし、Shift + Ctrl + Alt + i を押して字下げを減らします。

ブレースマッチング

メソッドエディターは、一致する括弧のセット（角括弧（大括弧）、波括弧（中括弧）、または丸括弧（括弧））を認識し、どちらかをクリックして選択したときに一致する括弧をハイライト表示します。次のキーボードショートカットを使用して、一致する括弧の間を移動し、括弧内の内容を選択します。

- Ctrl + M : 一致する括弧の間でカーソルを移動します。
- Ctrl + Shift + M : 二つの一致する括弧の間のテキスト範囲全体を選択します。

アプリケーションのデバッグ方法

コンパイルエラーの表示

メソッドにコンパイルエラーがある場合には、そのメソッドノードに赤い十字が表示されます。

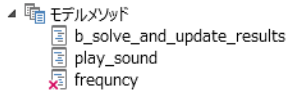


Figure 4-7: frequency メソッドにはコンパイルエラーが含まれています。他のメソッドにはエラーはありません。

フォームオブジェクトのローカルメソッドにコンパイルエラーがある場合には、ローカルメソッドのアイコンに赤い十字が表示されます。



Figure 4-8: この入力フィールドのローカルメソッドにコンパイルエラーの原因のコードが含まれていることを示しています。

メソッドエラーの状態ではそのメソッドを開き、コンパイルエラーを示すコード行を確認します。誤ったコードには下線が引かれており、その下線付きコードの上にマウスを置くとツールチップに問題の種類が表示されます。

デバッグツール

アプリケーションビルダーには、アプリケーションで使用される、またはモデルでモデルメソッドとして使用されるメソッドのコードをデバッグするためのツールが含まれています。この機能は、メソッドリボンツールバーのデバッグセクションとブレークポイントセクションから利用することができます。

アプリケーションのテスト実行を一時停止する必要があるコードの行に、マークを付けることができます。これは、例えば、メソッドが本当にifステートメントを入力するかを確認したり、メソッドが実際に実行されたかどうかを確認する場合などに利用することができます。行番号の左側の余白をクリックして、その行にブレークポイントを追加することによって、コードの行をマークすることができます。これによって、余白にブレークポイントを示す赤い丸印が表示されます。空の行の余白をクリックしてもブレークポイントは追加されないことに注意してください。— 行には、ブレークポイントが意味をなすためのステートメントが含まれていなければなりません。

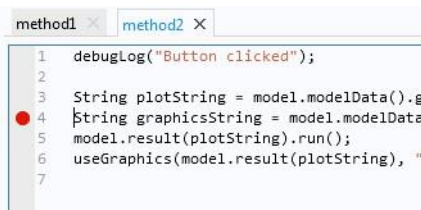


Figure 4-9: 4行目にブレークポイントを持つメソッドエディターウィンドウ

ブレークポイントを削除するには、それをもう一度クリックします。全てのブレークポイントは、メソッドリボンタブのブレークポイントセクションにある**全て解除**のボタン(🗑️)を使用すると、1回のアクションで削除することができます。場合によっては、アプリケーションのテスト実行時には既存のブレークポイントを無視するようになると便利です。このような場合には、**全て無効化**(🚫)のトグルボタンを使用して全てのブレークポイントを無効にすることができます。無効とされたブレークポイントは空の赤い円で示されます。



Figure 4-10: メソッド2のエディターの4行目の無効ブレークポイント

アプリケーションのテスト実行中にメソッドが実行されると、メソッドエディターウィンドウが開き、メソッドにブレークポイントが設定されて有効になっている場合には、メソッドはブレークポイントのその行で実行を一時停止します。その後、その行全体が黄色でハイライト表示され、**継続** (→)、**ステップ** (→|)、および**ヘステップ** (→|≡) のボタンが**デバッグ**リボンタブで有効になります。**継続**ボタンと**ステップ**ボタンは、デバッグ機能で一時停止されている場合にのみ使用できます。

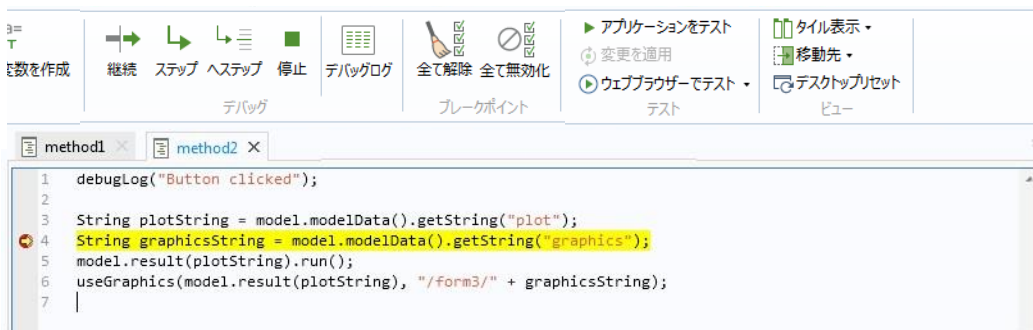


Figure 4-11: デバッグはブレークポイントで停止し、[継続]ボタンと[ステップ]ボタンを使用できます。
このコードでは、debugLog メソッドを使用してデバッグ情報をデバッグログに出力します。

継続ボタンをクリックする(または F5 キーを押す)と、メソッドは完了するか別のブレークポイントに到達するまで実行を続けます。**ステップ**ボタンをクリックする(または F6 キーを押す)と、メソッドは次の行まで実行されます。**ヘステップ**ボタンは、**ステップ**ボタンと同じように機能します。但し、その行に別のメソッドまたはユーティリティメソッドの呼び出しが含まれている場合は、**ヘステップ**ボタンをクリックする(または F7 キーを押す)と、現在のメソッドの次の行に進むのではなく、その呼び出すメソッドの最初の行に移動します。また、メソッドを最初に実行させる段階で有効な**停止**ボタン (■) もあります。これをクリックすると、現在のメソッドが強制的に停止されます。例えば、すぐにデバッグを停止したい場合や、メソッドが無制限ループになった場合などに利用することができます。

多くの場合、コード行に達したことを示すメッセージを出力したり、特定のオブジェクトの値を出力したりできると便利です。これは、debugLog メソッドを使用して行うことができます。これらのオブジェクトの文字列、倍精度、整数、ブーリアン、1D または 2D 配列(行列)のタイプを入力として受け取る debugLog メソッドがあります。これらのメソッドは、**デバッグログ**ウィンドウにメッセージを出力します。**メソッド**リボンタブの**デバッグログ** (≡) ボタンを使用して、**デバッグログ**ビューの表示を制御することができます。

エラーと警告ウィンドウ

メソッドのコンパイルやシンタックスのエラーと警告は、**エラーと警告**ウィンドウに表示されます。このウィンドウには、現在のエラーと警告の数が上部に表示されます。メソッドツールバーの**シンタックスチェック**のボタン (ABC) をクリックすると、シンタックスチェックが実行されてエラーと警告が表示されます。その下側の表では、**メソッド**と**ライン**の列にそれぞれ、エラーと警告が発生したメソッドとラインが表示されます。**メッセージ**の列には、特定のエラーまたは警告に関する情報が表示されます。警

告またはエラーの表の行をダブルクリックすると、それに対応するメソッドのエディターウィンドウで発生したコード箇所がハイライト表示されます。

メソッドのランタイムエラーの処理

アプリケーションでメソッドを実行すると、ランタイムエラーが発生する可能性があります。このようなエラーは、シンタックスチェックやコンパイル中に検出されないことがあります。このようなエラーが発生すると、エラーウィンドウが表示され、エラーの種類に関する情報が表示されます。**詳細**をクリックすると、クラスとメソッドの名前、行番号、メソッドへのリンクなどの追加情報が表示され、エディターウィンドウでデバッグできるようになります。

実行中のメソッドの停止

アプリケーションをテストするとき、メソッドは無限ループを完了させる、または収束させるために非常に長い時間がかかるかもしれません。Ctrl + Pause を押して、アプリケーションのテスト中にメソッドの実行を停止することができます。

COMSOL Multiphysics Application Builder Reference Manual 日本語訳版 (バージョン 5.3a)

2018 年 2 月 28 日 初版発行 (バージョン 5.3a)

著者: COMSOL AB. / COMSOL, Inc.

編集: 計測エンジニアリングシステム株式会社

Printed in Japan

ソフトウェアはライセンスの同意条項のもとでのみ使用またはバックアップが許可されます。
本書の一部または全部を著作権法の定める範囲を越え、無断で複写、複製、転載することを禁じます。



URL: <https://www.comsol.jp>



計測エンジニアリングシステム株式会社

〒101-0047 東京都千代田区内神田 1-9-5 井門内神田ビル 5F

TEL: 03-5282-7040/FAX: 03-5282-0808

URL: <http://www.kesco.co.jp>

E-mail: support@kesco.co.jp