Sizer for WinForms

2018.04.11 更新

グレープシティ株式会社

目次

はじめに

Sizer for WinForms により、コードを記述する必要なく、解像度に依存しないアプリケーションを作成します。Sizer for WinForms に備わっている強力な **C1SizerLight** コンポーネントと **C1Sizer** がこれを可能にします。含まれているすべてのコントロールを比例的にサイズ変更できるようになったため、**Windows フォーム**はどの解像度でもその外観を維持します。

C1Sizer は、.NET フレームワークが提供する基本的なレイアウト機能（Dock と Anchor プロパティ）を拡張する強力なグリッドレイアウトマネージャを備えたコンテナコントロールです。**C1Sizer** コントロールでは、バンドから成るグリッドを定義した後、これらのバンドに配置する各コントロールを追加できます。**C1Sizer** コントロールがサイズ変更されると、バンドが自動的に再計算され、含まれているコントロールはそれぞれの新しい位置に自動的に移動します。設計時にバンドを設定し、それらをスプリッタとして機能させるか、コントロールのサイズ変更時にサイズを一定に維持するように構成できます。

コンポーネントをプロジェクトに組み込む方法

コンポーネントの組み込み

Visual Studio では、ツールボックスにコンポーネントを追加しただけでは、プロジェクトにコンポーネントを追加したことにはなりません。プロジェクトの参照設定へ追加された時点でコンポーネントが組み込まれます。

以下のいずれかの操作を行うとプロジェクトへコンポーネントが組み込まれます。

1. フォームにコンポーネントを配置する
2. ソリューションエクスプローラ上で参照の追加を行う

プロジェクトに組み込まれているコンポーネントの一覧は、ソリューションエクスプローラで確認できます。また、各コンポーネントが使用している DLL もソリューションエクスプローラに登録される場合があります。詳細については、Visual Studio の製品ヘルプを参照してください。

本製品で使用しているコンポーネントの一覧を以下に示します。

ファイル	内容
C1.C1Sizer.2.dll	本体アセンブリ
C1.C1Sizer.4.dll	本体アセンブリ（※）
C1.C1Sizer.4.Design.dll	デザイナアセンブリ（※）

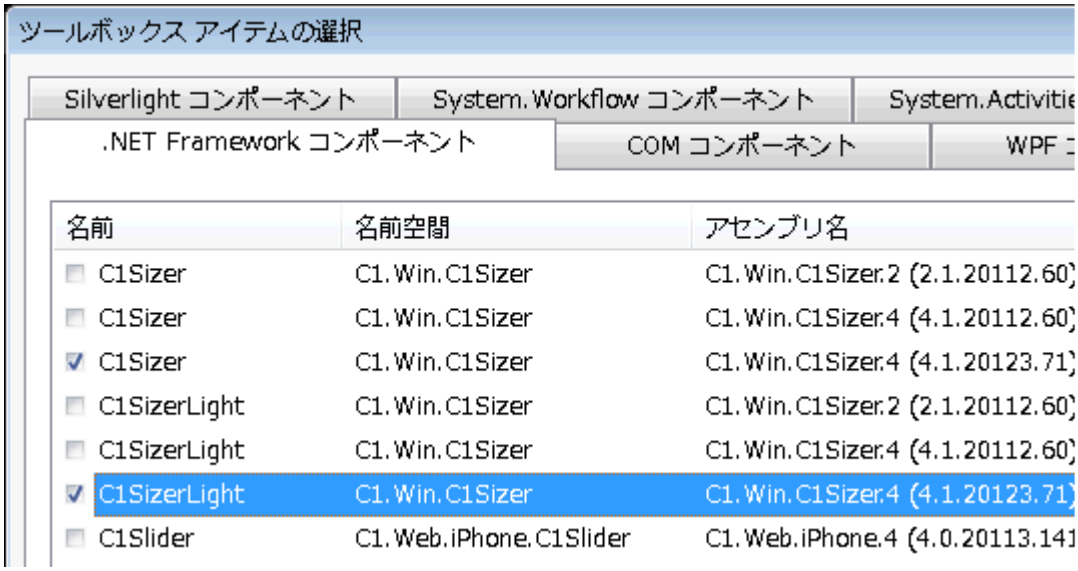
※ .NET Framework 4 以上でご利用いただけます。

フォームにコンポーネントを配置する方法

以下に、C1Sizer コントロールをツールボックスに追加し、フォームに配置する方法を示します。これにより、コンポーネントがプロジェクトに組み込まれます。

配置手順

1. [ツール]メニューから[ツールボックス アイテムの選択]を選択します。
2. [ツールボックス アイテムの選択]ダイアログの「.NET Framework コンポーネント」タブを選択します。
3. 使用するコンポーネントのチェックボックスを ON にして〈OK〉ボタンをクリックしてください。ツールボックスに指定したコントロールのアイコンが表示されます。



4. ツールボックスから指定したコントロールのアイコンを選択してフォームに配置します。ソリューションエクスプローラの参照設定に指定したコントロールの名前空間が追加されます。

コンポーネントのランタイムファイル

Sizer for WinForms のランタイムファイルは、C:\Program Files\ComponentOne\WinForms\Bin\ フォルダにインストールされる次のファイルです。

ファイル	内容
C1.C1Sizer.2.dll	本体アセンブリ
C1.C1Sizer.4.dll	本体アセンブリ(※)

※ .NET Framework 4 以上でご利用いただけます。

注意:本製品に含まれているファイルのうち、上記以外のファイルは配布できません。

製品の概要

Sizer for WinForms は、コードを記述しなくても解像度に依存しないアプリケーションを作成できるようにする2つのコンポーネントを提供します。

- **C1Sizer ('C1Sizer クラス' in the on-line documentation)**
  C1Sizer は、.NET Framework から提供される基本レイアウト機能（Dock プロパティと Anchor プロパティ）を拡張する強力なグリッドレイアウトマネージャを持つコンテナコントロールです。C1Sizer コントロールを使用すると、いくつかのテーブル領域で構成されるグリッドを定義し、それらのテーブル領域にスナップするコントロールを追加できます。C1Sizer コントロールがサイズ変更されると、テーブル領域が自動的に再計算され、内部のコントロールは自動的に新しい位置に移動します。設計時にテーブル領域を設定して、スプリッタとして動作したり、コントロールがサイズ変更されてもサイズが固定されるように調整できます。
- **C1SizerLight ('C1SizerLight クラス' in the on-line documentation)**
  C1SizerLight は、非ビジュアルコンポーネントです。これをフォームに追加すると、フォームのサイズと位置が記録されます。フォームがサイズ変更された場合、**C1SizerLight** は、フォーム上のすべてのコントロールを同じ割合でサイズ変更するため、どの解像度でもフォームの外観が維持されます。また、**C1SizerLight** は、フォーム上の全部または一部のコントロールのフォントをサイズ変更することもできます。

C1Sizer コントロールと **C1SizerLight**コンポーネントは、**Bounds ('Bounds プロパティ' in the on-line documentation)** プロパティを更新することによって、他のコントロールのサイズを変更します。コントロールによっては、**C1Sizer** と **C1SizerLight** は、サイズを変更するには、**Bounds** プロパティ以外の特定のプロパティを使用する必要があります。

下表には、**Bounds** プロパティ以外のプロパティ設定を必要とするコントロールとそのプロパティの設定を示します。

コントロール	プロパティ設定
TextBox	TextBox コントロールは、**MultiLine = True** の場合にのみ高さの設定を可能にします。そうでない場合には、フォントに合わせて高さは自動的に設定されます。
ComboBox	ComboBox コントロールは、**DrawMode = OwnerDraw** の場合にのみ高さの設定を可能にします。
ListBox	ListBox コントロールは、**IntegralHeight = False** の場合にのみ高さ（正確な）の設定を可能にします。

主な特長

役立ちそうな **C1Sizer** の主な特長として、次の事項が挙げられます。

- **グラデーション背景の作成**
  "グラデーションの編集"スマートタグからアクセス可能な、使いやすい C1Sizer グラデーションエディタを使用して、サイザーパネルにすばやくグラデーションを追加したり、グラデーション設定をカスタマイズしたりできます。
- **角に丸みがあるフォームの更新**
  新しい Border プロパティにより、丸みのある角をサイザーに追加したり、色、太さなどを変更したりする柔軟性が得られます。
- **背景に画像を追加する際の柔軟性**
  新しい画像プロパティにより、背景の一部に画像を使用する際に、より優れた機能と柔軟性が得られます。これで、画像の配置とスケールを制御できるようになりました。
- **フォーム上での簡単なコントロールの整列**
  C1Sizer コントロールは、各コントロールがフォーム上で1つまたは複数のグリッドボックスを占有する一枚のグラフ用紙のように機能します。グラフを使用して、フォーム上でコントロールをきれいにレイアウトできます。子コントロールを作成する前または作成した後にグリッドを自由に設定できます。
- **含まれているコントロールを比例的にリサイズ**
  C1SizerLight コントロールは、表示しないコンポーネントであり、フォームのサイズと位置を制御します。フォームをリサイズすると、C1SizerLight が含まれているすべてのコントロールを比例的にリサイズし、どの解像度でもフォームの外観が保持されます。また、C1SizerLight で含まれているいくつかのコントロールのフォントもリサイズできます。
- **使いやすいグリッドエディタ**
  グリッドエディタを使用して、フォームを効率的に設定します。1つまたは複数の C1Sizer コントロールを追加し、フォームを埋めるように Dock プロパティを設定するだけです。次に、グリッドエディタを使用してグリッドのレイアウトを設定し、グリッドに配置するコントロールを追加します。グリッドのバンド（行と列）の数とサイズを設定したり、スプリッタとして機能するバンドや固定サイズのバンドを指定したりすることもできます。

設計時サポート

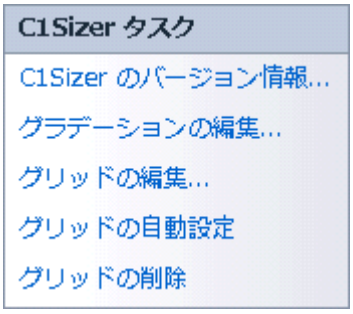
C1Sizer ('C1Sizer クラス' in the on-line documentation) には、カスタム コンテキスト メニュー、スマートタグ、および豊富な設計時サポートを提供するデザイナが用意されていることにより、オブジェクトモデルとの作業を簡素化します。

次のセクションでは、設計時に **C1Sizer** のコントロール/コンポーネントを構成する方法を説明します。

C1Sizer のスマートタグ

C1Sizer ('C1Sizer クラス' in the on-line documentation) コントロールには、Visual Studio の.NET コントロールに提供しているスマートタグが用意されています。スマートタグは、各コンポーネント/コントロールのよく使用されるプロパティを提供するタスクメニューへのショートカットです。**C1Sizer** コントロールのスマートタグを使用すると、迅速かつ容易に **C1Sizer グリッド エディタ**および他のよく使用されるプロパティにアクセスできます。

[C1Sizer タスク] メニューにアクセスするには、C1Sizer コントロールの右上隅にあるスマートタグをクリックします。**[C1Sizer タスク]** メニューが開きます。



[C1Sizer タスク] メニューは、以下のように操作します。

[C1Sizer のバージョン情報]

[C1Sizer のバージョン情報]項目をクリックすると、**[C1Sizer のバージョン情報]**ダイアログボックスが表示され、C1Sizer のバージョン番号、オンラインリソースなどを見ることができます。

[グラデーションの編集]

[グラデーションの編集]項目をクリックすると、**[C1Sizer のグラデーションエディタ]**ダイアログボックスが表示されます。C1Sizer のグラデーションエディタの要素の詳細については、「C1Sizer グラデーションエディタ」トピックを参照してください。

[グリッドの編集]

[グリッドの編集]項目をクリックすると、C1Sizer グリッドエディタが開きます。グリッドの要素の詳細は、「C1Sizer グリッドエディタ」を参照してください。

[グリッドの自動設定]

[グリッドの自動設定]項目をクリックすると、グリッド中のコントロールを持たない未使用のバンドをすべてクリアします。

[グリッドの削除]

[グリッドの削除] 項目をクリックすると、グリッドからバンドを削除します。

[親コンテナでドッキングを解除する]

[親コンテナでドッキングを解除する]をクリックすると、C1Sizer コントロールはその親コンテナ内に入ります。

C1SizerLight のスマートタグ

C1SizerLight ('C1SizerLight クラス' in the on-line documentation) コンポーネントのスマートタグを使用すると、迅速かつ容易に **C1Sizer のバージョン情報** のダイアログボックスにアクセスできます。

[C1SizerLight タスク] メニューにアクセスするには、**C1SizerLight** コンポーネントの右上隅にあるスマートタグをクリックします。**[C1SizerLight タスク]** メニューが開きます。

[C1SizerLight タスク] メニューは、以下のように操作します。

[C1Sizer のバージョン情報]

[C1Sizer のバージョン情報]項目をクリックすると、**[C1Sizer のバージョン情報]**ダイアログボックスが表示され、C1Sizer のバージョン番号、オンラインリソースなどを見ることができます。

コンテキストメニュー

C1Sizer ('C1Sizer クラス' in the on-line documentation) の各コンテキストメニューに、Visual Studio がすべての .NET コントロールに対して提供する追加のコマンドが用意されています。

C1Sizer コンテキストメニュー

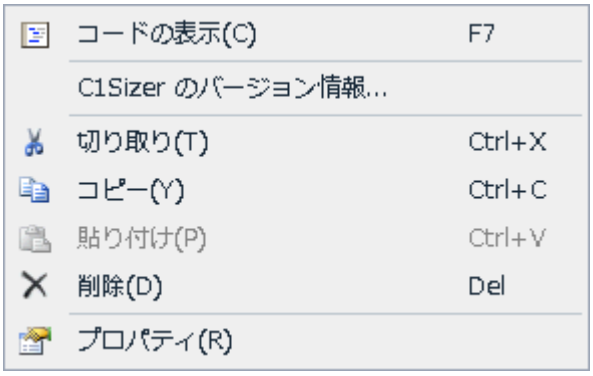
C1Sizer コントロールを右クリックして、そのコンテキストメニューを開きます。

下表に、**C1Sizer** に追加されたカスタム項目を簡単に説明します。

カスタム項目	説明
C1Sizer のバージョン情報	C1Sizer のバージョン情報やオンラインリソースを検索するのに便利な **C1Sizer のバージョン情報**ダイアログボックスを表示します。
グラデーションの編集	**C1Sizer グラデーションエディタ**を開きます。
グリッドの編集	**C1Sizer グリッドエディタ**を開きます。
グリッドの自動設定	グリッド中のコントロールを持たない未使用のバンドをすべてクリアします。
グリッドの削除	グリッドからバンドを削除します。

C1SizerLight コンテキストメニュー

C1SizerLight ('C1SizerLight クラス' in the on-line documentation) コンポーネント上で右クリックすると、**C1SizerLight** コンテキストメニューが表示されます。



下表に、**C1SizerLight** に追加されたカスタム項目を簡単に説明します。

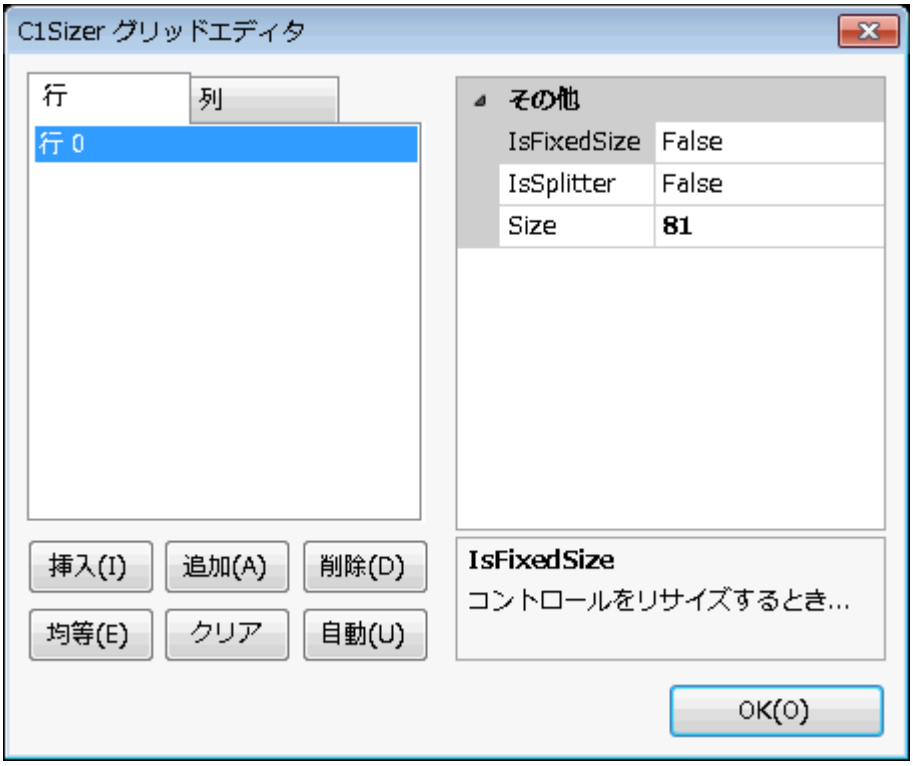
コマンド	説明
C1Sizer のバージョン情報	C1Sizer のバージョン情報やオンラインリソースを検索するのに便利な **C1Sizer のバージョン情報** のダイアログボックスを表示します。

C1Sizer グリッドエディタ

C1Sizer には、**グリッドエディタ**が用意されており、設計時にグリッドの列や行にバンドを追加したり、編集したりすることができます。

C1Sizer グリッドエディタへのアクセス

C1Sizer タスクメニューを開いて、グリッドの編集アイテムをクリックする、または **C1Sizer** コントロールを右クリックしてコンテキストメニューからグリッドの編集を選択します。



タブ

C1Sizer グリッドエディタは、重要なタブを二つ提供します。「**行**」と「**列**」二つのタブは、同じコマンドボタンとプロパティを共有しています。「**列**」タブをクリックして、グリッドの列のバンドを作成または修正します。「**行**」タブをクリックして、グリッドの行のバン

ドを作成または修正します。

コマンドボタン

二つのタブは、以下のコマンドボタンを共有しています。

コマンド	説明
挿入	新しいバンドをコレクション内の指定された位置に挿入します。
追加	新しいバンドをコレクションに追加します。
削除	選択されたバンドを削除します。
均等	すべてのバンドを同じサイズに設定します。
クリア	すべてのバンドをクリアします。
自動	子コントロールに基づいてバンドを作成します。
OK	変更を更新して **C1Sizer グリッドエディタ**を閉じます。

プロパティ

二つのタブは、以下のプロパティを共有しています。

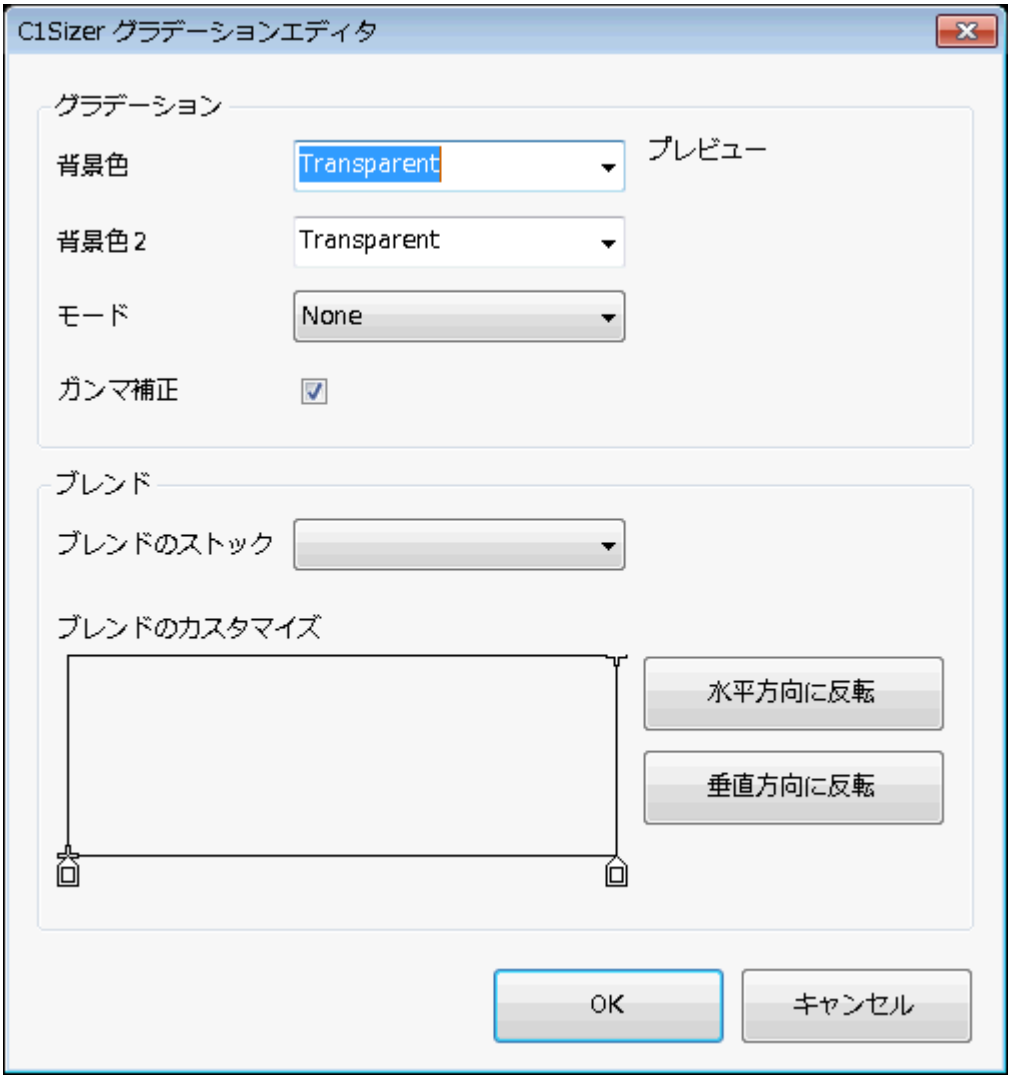
プロパティ	説明
IsFixedSize	コントロールをリサイズするときバンドのサイズを固定するかどうかを指定します。デフォルトの値は、False です。
IsSplitter	バンドはスプリッタのように動作するかどうか（実行時にマウスでリサイズできるかどうか）を指定します。デフォルトの値は、False です。
Size	行の高さまたは列の幅をピクセル単位で取得または設定します。

C1Sizer グラデーションエディタ

C1Sizer ('C1Sizer クラス' in the on-line documentation) には、デザイン時に C1Sizer グリッドにグラデーションを追加するグラデーションエディタが用意されています。

C1Sizer グラデーションエディタにアクセスする:

[**C1Sizer のタスク**]メニューを開き、グラデーションの編集項目をクリックするか、**C1Sizer** コントロールを右クリックして、そのコンテキストメニューから[**グラデーションの編集**]を選択します。



グラデーショングループ

グラデーションエディタは、[**グラデーション**]と[**ブレンド**]という2つのグループで構成されます。[**グラデーション**]グループは BackColor、Gradient の各プロパティ、および **C1Sizer** コントロールに対して行うグラデーションの更新をプレビューできるプレビューウィンドウで構成されます。

「**グラデーション**」グループは、以下のプロパティで構成されます。

プロパティ	説明
C1Sizer.BackColor ('BackColor プロパティ' in the on-line documentation) プロパティ	背景をペイントするために使用される Drawing.Color を設定します。
Gradient.BackColor2 ('BackColor2 プロパティ' in the on-line documentation) プロパティ	背景のグラデーションを作成するために使用される二次色を取得または設定します。
GradientMode ('GradientMode 列挙体' in the on-line documentation) 列挙体	背景のグラデーションモードを指定します。
Gradient.GammaCorrection ('GammaCorrection プロパティ' in the on-line documentation) プロパティ	背景のグラデーションにガンマ補正を適用するかどうかを取得または設定します。

ブレンドグループ

[**ブレンド**]グループでは、ストックされたブレンドから選択したり、ブレンドカラーを水平方向または垂直方向に反転してブレンド

をカスタマイズしたりできます。

C1SizerLight コンポーネント

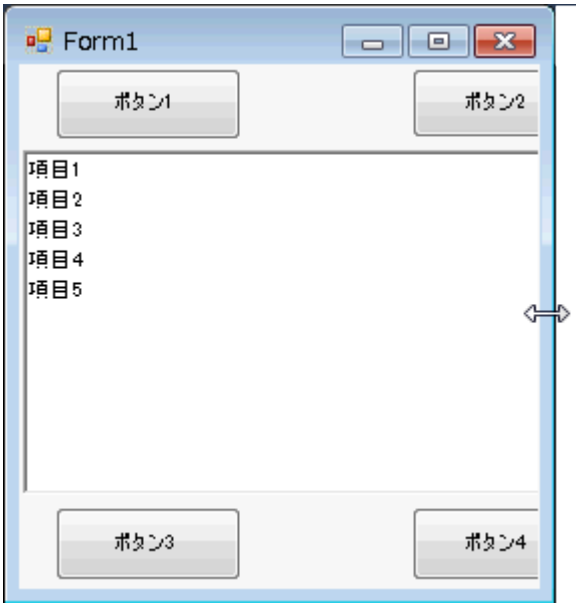
C1SizerLight ('C1SizerLight クラス' in the on-line documentation) は非ビジュアルコンポーネントです。これをフォームに追加すると、フォームのサイズと位置が記録されます。フォームがサイズ変更された場合、**C1SizerLight** は、フォーム上のすべてのコントロールを同じ割合でサイズ変更するため、どの解像度でもフォームの外観が維持されます。また、**C1SizerLight** は、フォーム上の全部または一部のコントロールのフォントをサイズ変更することもできます。

C1SizerLight コンポーネントはたいへん簡単に使用できます。まず Windows フォームを通常どおりに作成し（または、既存の Windows フォームを開き）、それに **C1SizerLight** コントロールを追加します。次にプロジェクトを実行し、フォームをサイズ変更します。フォーム上のすべてのコントロールは、フォームと一緒に自動的にサイズ変更されます（フォントも含む）。

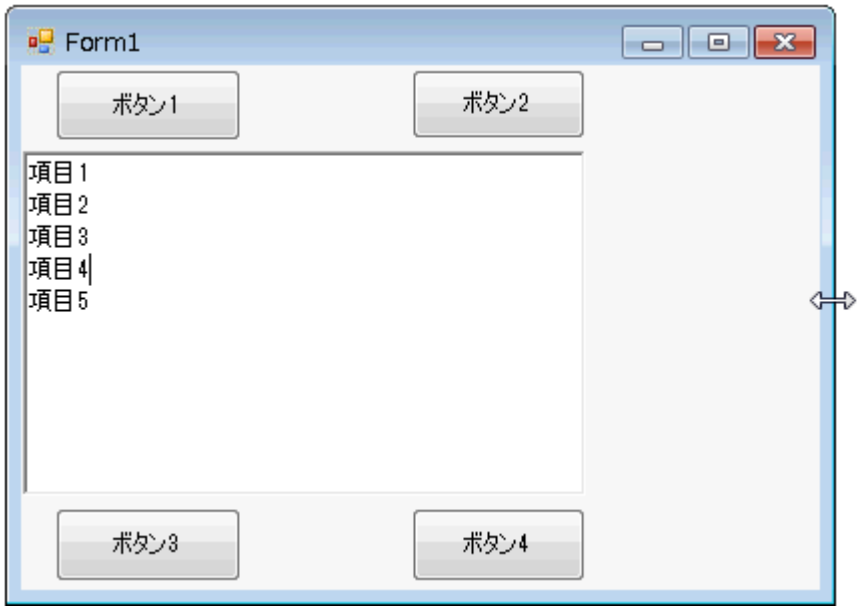
クイックチュートリアル

クイックチュートリアルでは、**C1SizerLight ('C1SizerLight クラス' in the on-line documentation)** コントロールの動作について説明します。また、**C1SizerLight.ResizeFont ('ResizeFonts プロパティ' in the on-line documentation)** プロパティと**C1SizerLight.ResizingFont ('ResizingFont イベント' in the on-line documentation)** イベントの処理についても記載されています。
これをテストするには、Visual Studio を開いて新しいプロジェクトを作成した後、次のような外観になるようにフォームを設計します。

次にプロジェクトを実行し、フォームをサイズ変更します。フォームを縮小すると、コントロールの一部がクリップされます。

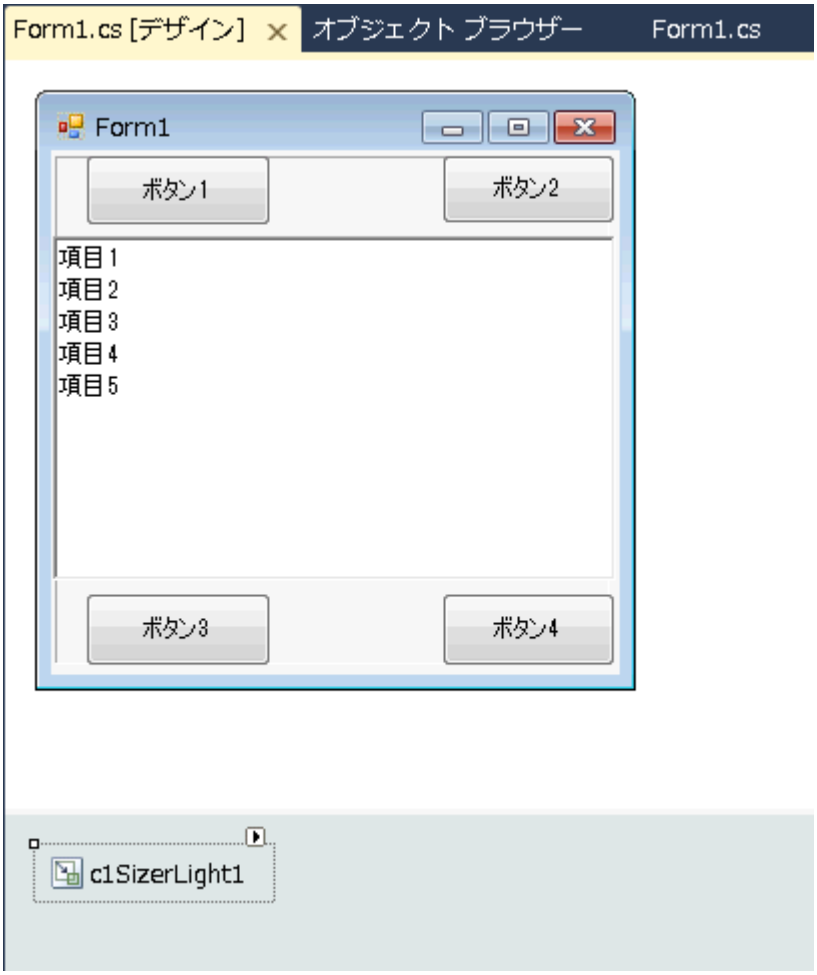


拡大すると、空の灰色の領域が表示されます。

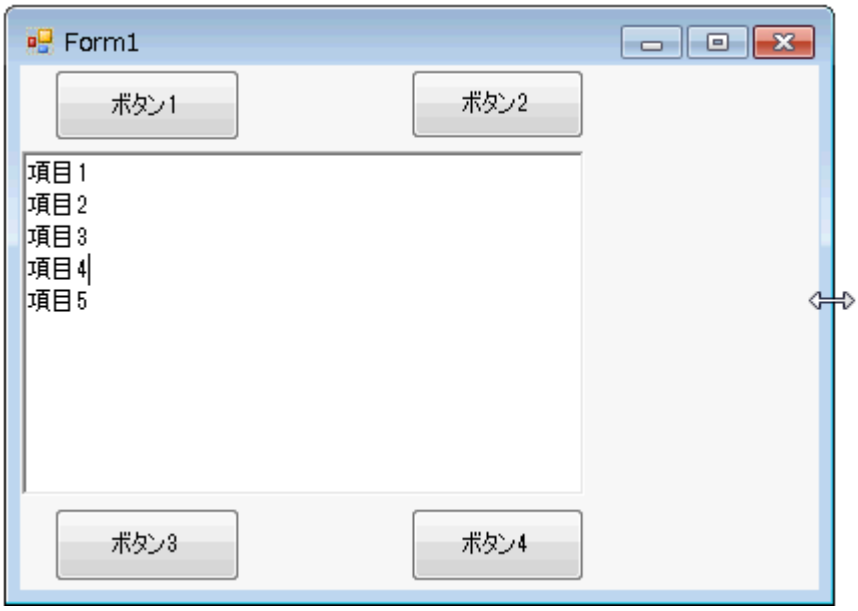


この動作は、コントロールの **Dock** プロパティや **Align** プロパティを使用すると改善できます。この2つのプロパティは、コントロールが互いに重ならないようにしたり、フォーム上に空の領域が残らないようにしますが、同時には使用できません。また、モニタの解像度によっては、フォームを拡大したときにフォントが非常に小さく表示される場合があります。

フォームのサイズを固定してサイズ変更機能をあきらめるのではなく、フォームに **C1SizerLight** コンポーネントを追加してみます。これは非ビジュアルコンポーネントなので、フォームの下部にあるトレイ領域に表示されます。



プロパティを設定する必要も、コードを記述する必要もなく、プロジェクトを再度実行するだけです。フォームをサイズ変更してみます。含まれているすべてのコントロールが自動的にサイズ変更され、フォームの形が維持されます。



フォントのサイズ変更

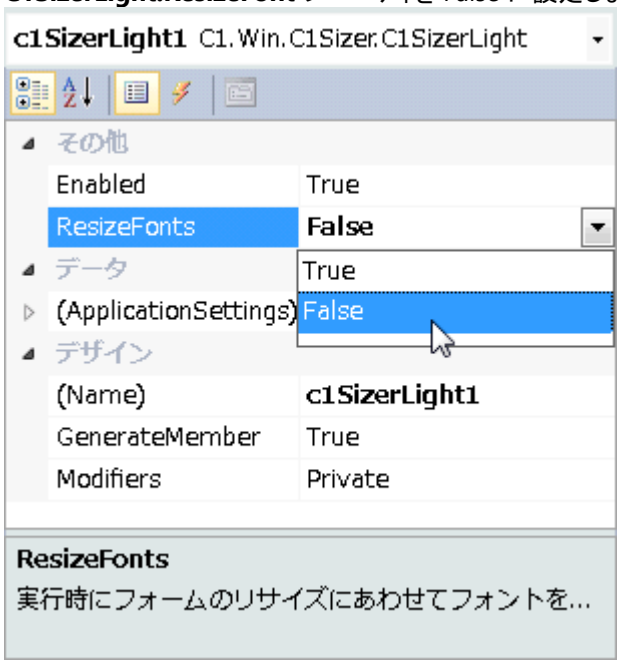
フォームをサイズ変更すると、ボタンコントロールのフォントは新しいサイズに合わせてサイズ変更されます。これは、**C1SizerLight** のデフォルトの動作です。

この動作を抑止して、元のフォントサイズをすべて維持することもできます。それには、**C1SizerLight.ResizeFont** プロパティ

を使用します。**C1SizerLight.ResizeFont** プロパティを False に設定すると、フォントはサイズ変更されません。

C1SizerLight.ResizeFont プロパティを設計時に False に設定するには、以下の手順に従います。

1. プロパティウィンドウで **C1SizerLight** を選択します。
2. **C1SizerLight.ResizeFont** プロパティを False に設定します。



また、特定のコントロールでフォントがサイズ変更されないようにすることもできます。たとえば、サンプルフォームの場合、リストコントロールのフォントサイズは固定し、ボタンコントロールのフォントだけを更新できると便利です。一般に、内容をスクロールできるコントロールにフォントのサイズ変更機能は不要だからです。それには、**C1SizerLight.ResizingFont** イベントを処理して、このイベントを一部のコントロールでキャンセルする必要があります。たとえば、次のイベントハンドラは、コンポーネントがボタンコントロールのフォントだけをサイズ変更するようにします。

▶ Visual Basic コードの書き方

Visual Basic

```
Private Sub C1SizerLight1_ResizingFont(ByVal sender As Object, _
    ByVal e As C1.Win.C1Sizer.C1SizerLightEventArgs) Handles _
    C1SizerLight1.ResizingFont
    If Not (e.Control.GetType() Is GetType(Button)) Then
    e.Cancel = True
    End If
End Sub
```

▶ C# コードの書き方

C#

```
private void c1SizerLight1_ResizingFont(object sender,
    C1.Win.C1Sizer.C1SizerLightEventArgs e)
    {
    if (!(e.Control is Button))
    e.Cancel = true;
}
```

プロジェクトを再度実行し、フォームをサイズ変更すると、**ListBox** コントロールは元のフォントを維持します。

ドッキングおよびネストされたコントロール

C1SizerLight には強力で使いやすいさまざまな機能がありますが、.NET Framework のネイティブのレイアウト機能をそのまま使用することもできます。たとえば、フォームに ToolBar コントロールや StatusBar コントロールを追加する場合は、これらのコントロールをフォームの最上部または最下部にドッキングし、**C1SizerLight** がこれらのコントロールをサイズ変更しないようにすることが普通です。

このため、**C1SizerLight** は、ドッキングされたコントロールのサイズ変更は行いません。コントロールがドッキングされている点が考慮され、それに応じてフォームのクライアント領域が狭められるため、ドッキングされていないコントロールはすべて正しくサイズ変更されます。

これにより、**C1SizerLight** を簡単に使用できると共に、.NET Framework に組み込まれたレイアウト機能も活用できます。

MDI フォーム

C1SizerLight は、MDI 子フォームでも動作します。**C1SizerLight** コンポーネントを子フォームに追加するだけで、通常のフォームと同様に均等にサイズ変更されるようになります。

C1Sizer コントロール

C1Sizer ('C1Sizer クラス' in the on-line documentation) は、.NET Framework から提供される基本レイアウト機能（**Dock** プロパティと **Anchor** プロパティ）を拡張する強力なグリッドレイアウトマネージャを持つコンテナコントロールです。C1Sizer コントロールを使用すると、いくつかのテーブル領域で構成されるグリッドを定義し、それらのテーブル領域にスナップするコントロールを追加できます。C1Sizer コントロールがサイズ変更されると、テーブル領域が自動的に再計算され、内部のコントロールは自動的に新しい位置に移動します。設計時にテーブル領域を設定して、スプリッタとして動作したり、コントロールがサイズ変更されてもサイズが固定されるように調整できます。

C1Sizer を1枚の方眼紙と考えてください。各コンポーネントは、方眼紙の何個かのます目に収まります。これは、隣接するセルをマージすることができる Microsoft Word™ などのワードプロセッサの表や、同様の機能を持つ Java のグリッドバッグレイアウトに似ています。

プロジェクトで **C1Sizer** コントロールを使用する方法は、基本的に2つあります。子コントロールの作成前にグリッドを設定する方法については、「チュートリアル 1: グリッドを設定してからコントロールを追加する」と、作成後にグリッドを設定する方法については、「チュートリアル 2: コントロールを追加してからグリッドを設定する」を参照してください。

Bounds ('Bounds プロパティ' in the on-line documentation) プロパティ以外のプロパティが必要となるコントロールは以下の通りです。

コントロール	プロパティ設定
TextBox	TextBox コントロールは、MultiLine プロパティが True に設定されている場合のみ高さを設定できます。これ以外の場合は、フォントに応じて自動的に高さが設定されます。
ComboBox	ComboBox コントロールは、DrawMode プロパティが OwnerDraw に設定されている場合のみ高さを設定できます。
ListBox	ListBox コントロールは、IntegralHeight プロパティが False に設定されている場合のみ高さを（正確に）設定できます。

次のセクションでは、C1Sizer コントロールの主な特長を示すチュートリアルが提供されています。このチュートリアルでは、いくつかの簡単なプロジェクトを作成して各手順の詳細を説明しています。

下記は、そのチュートリアルの概要を記載しています。

名前	説明
チュートリアル 1	このチュートリアルでは、**C1Sizer** コントロールの主な機能を紹介します。行と列の挿入、グリッドにラベルやデータ入力コントロールの追加、利用されていないバンドのクリーンアップ、スプリッターの設定、固定サイズのバンドの作成などを実行してグリッドを設定する方法について説明します。
チュートリアル 2	このチュートリアルでは、**C1Sizer** コントロールの通常の使い方について説明します。コントロールを追加してグリッドを設定する方法について節召します。

チュートリアル 1: グリッドを設定してからコントロールを追加する

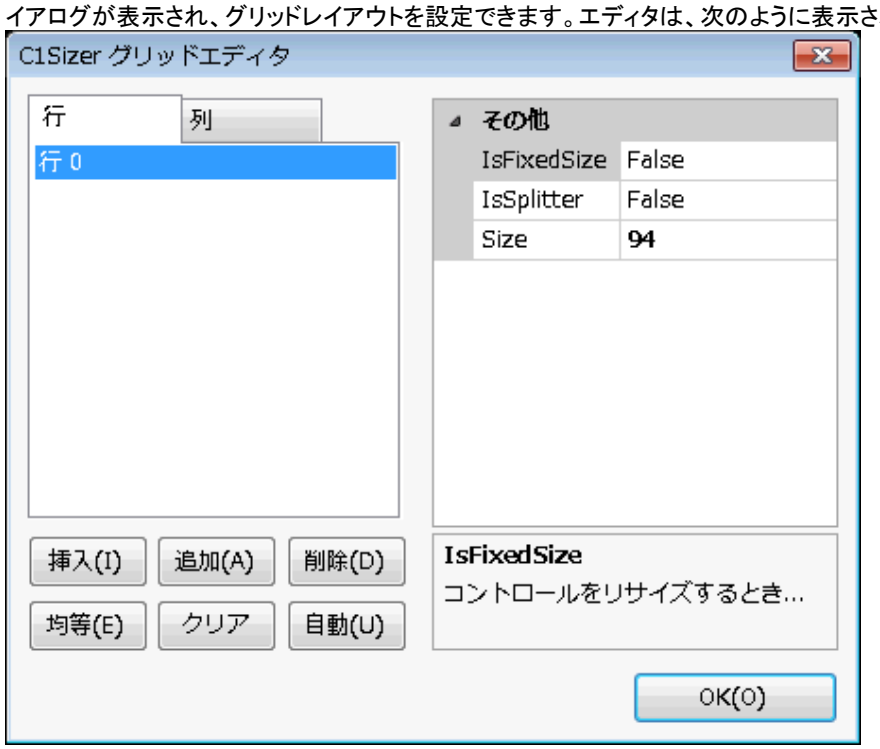
このチュートリアルでは、**C1Sizer グリッドエディタ**の要素とプロパティを使用してグリッドを設定する方法について学びます。このチュートリアルは以下の **C1Sizer** 機能を例示します。

- 行と列を挿入／削除する
- グリッド上のコントロールを整列する
- **C1Sizer.SplitterWidth ('SplitterWidth プロパティ' in the on-line documentation)** と **C1Sizer.BorderWidth ('BorderWidth プロパティ' in the on-line documentation)** プロパティを設定する
- AutoGrid を適用して使用されていないバンドをクリーンアップする
- **Band.IsFixedSize ('IsFixedSize プロパティ' in the on-line documentation)** と **Band.IsSplitter ('IsSplitter プロパティ'**

in the on-line documentation) プロパティを設定する

C1Sizer のためにグリッドを設定するには、以下の手順に従います。

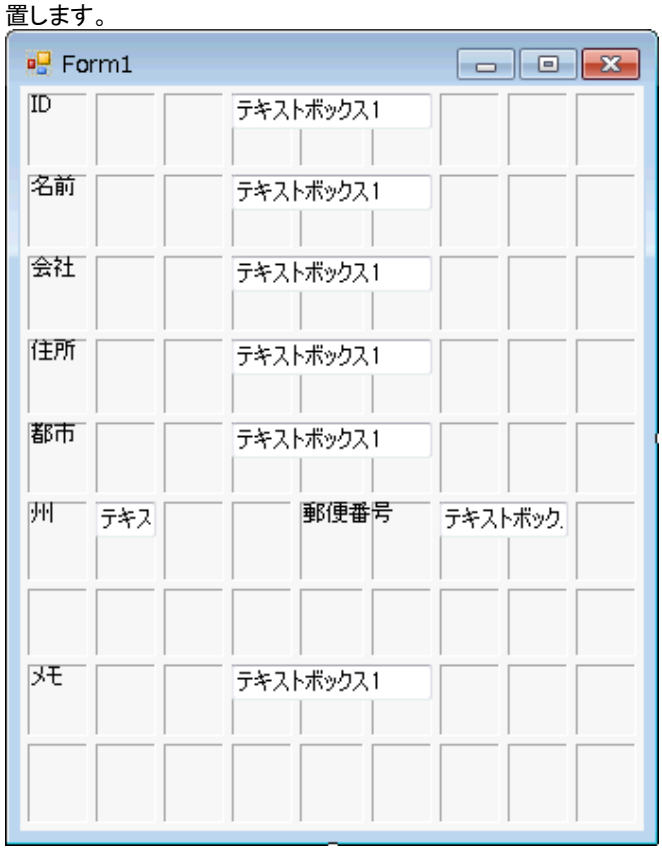
1. Visual Studio を開いて新しいプロジェクトを作成した後、**C1Sizer** コントロールをフォームに追加し、**Dock** プロパティを **DockStyle.Fill** に設定してフォーム全体を使用するようにします。この時点では、**C1Sizer** には、グリッドレイアウトも子コントロールも設定されていません。そのため、コントロールの使い方に関する簡単な指示メッセージが表示されますが、この段階では無視してかまいません。
2. 次に、**C1Sizer** コントロールを右クリックし、メニューの[グリッドの編集]オプションを選択します。[**C1Sizer グリッドエディタ**]ダイアログが表示され、グリッドレイアウトを設定できます。エディタは、次のように表示されます。



3. エディタをグリッドの隣に置き、コントロールに対して行った変更の効果がわかるようにします。次に、グリッドに行が8つできるまで〈**挿入**〉ボタンをクリックします。次に、「**列**」タブをクリックし、グリッドに列が8つできるまで再度〈**挿入**〉をクリックします。設計時にコントロールは次のように表示されます。

設計時のグリッドは、へこんだ四角形の集まりとして表示されます。〈**OK**〉をクリックしてエディタを閉じ、フォームにコントロールをいくつか追加します。コントロールがグリッドレイアウトにどのようにスナップするかに注目してください。どのコントロールもサイズ変更できます。また、コントロールを複数の行と列にまたがるように配置することもできます。

4. チュートリアルに従い、フォームの外観が次のようになるように、いくつかのテキストボックスとラベルコントロールを追加して配置します。



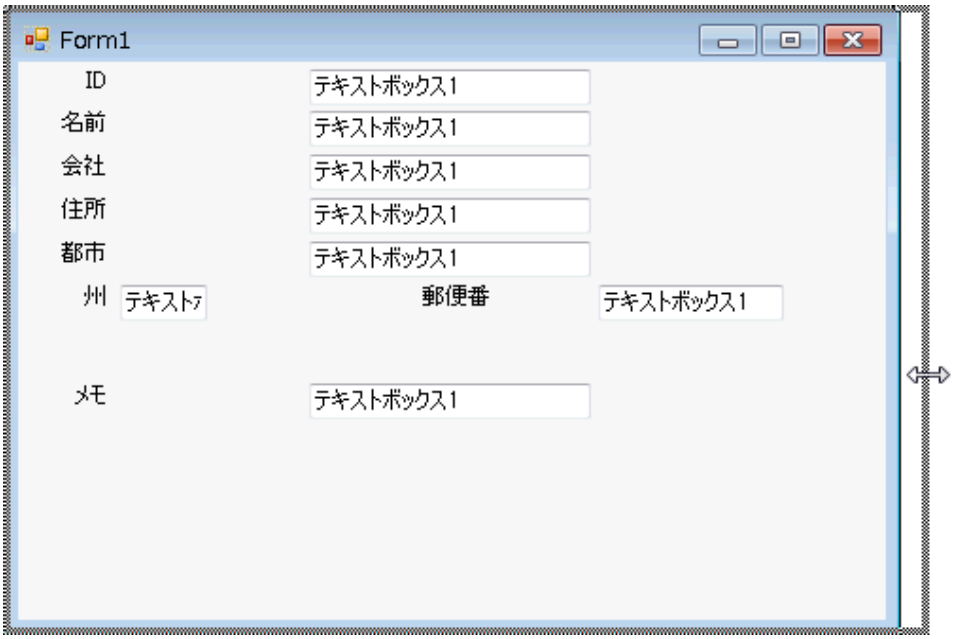
メモ: テキストボックスコントロールの **MultiLine** プロパティを必ず True に設定してください。そうしないと、垂直方向にサイズ変更できなくなります。また、フォーム上にテキストボックスを配置すると、自動的に3列に拡張します。

5. 2番目の列にあるカーソルを選択して、左側にドラッグしてラベルを配置するためのスペースを作成します。そして、各行上のカーソルをドラッグして上側にドラッグし、各行の高さを減ります。以下の画像のようになります:

6. 各ラベルの **AutoSize** プロパティを **False**に設定して、**TextAlign** プロパティを **TopRight** に設定します。
ラベルは、以下のように表示されます。

コントロールどうしが近すぎるまたは離れすぎている場合は、**C1Sizer.SplitterWidth ('SplitterWidth プロパティ' in the on-line documentation)** プロパティと **C1Sizer.BorderWidth ('BorderWidth プロパティ' in the on-line documentation)** プロパティを変更して目的のレイアウトにします。

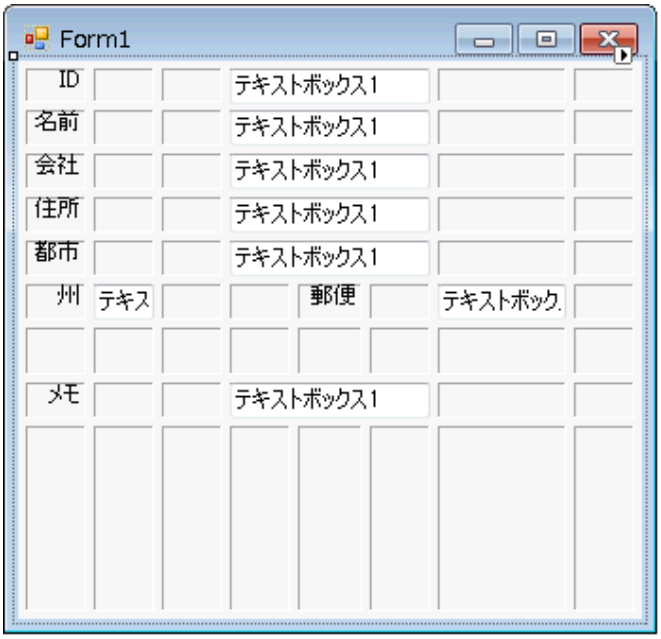
この段階でプロジェクトを実行すると、フォームをサイズ変更したときにテキストボックスとラベルが自動的にサイズ変更されます。



グリッドのクリーンアップ

現在、グリッドには8行、8列があります。これらのテーブル領域のほとんどは使用されて(コントロールが結び付けられて)いますが、すべてが使用されているわけではありません。レイアウトをクリーンアップして未使用のテーブル領域を取り除くには、**C1Sizer** コントロールを右クリックし、[**グリッドの自動設定**]オプションを選択します。これで、コントロールが結び付けられていないテーブル領域はすべて消去されます。

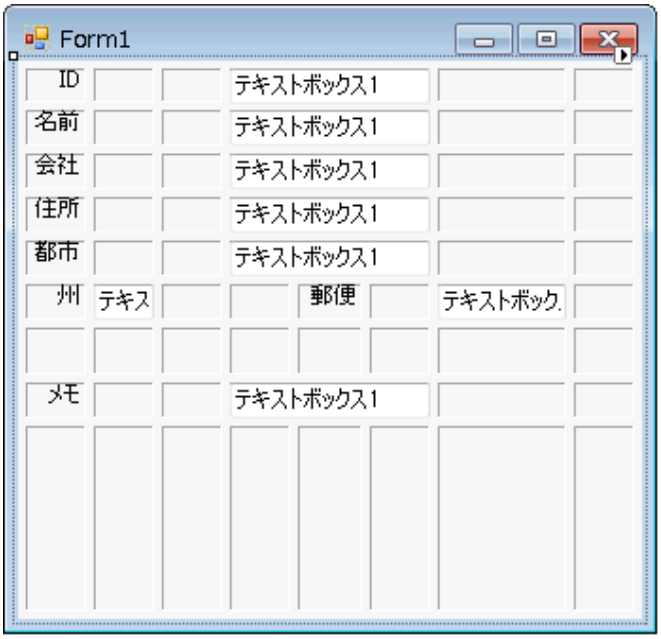
C1Sizer が以下のように表示されます。



行と列の削除

C1Sizer グリッドエディタの最初列（**Column 0**）を削除するには、**C1Sizer** コントロールを右クリックし、メニューから［**グリッドの編集**］オプションを選択します。**C1Sizer** グリッドエディタが表示されます。列 タブにて、［**削除**］ボタンをクリックし、**OK** ボタンを削除します。

C1Sizer グリッドが設計時に以下のように表示されます。



スプリッタの設定

グリッド内の各テーブル領域（行または列）は、スプリッタとして動作するように設定できます。たとえば、ユーザーが左側のラベル領域のサイズを制御できるようにするには、グリッドエディタを表示し、次に示すように、最初の列を選択して **Band.IsSplitter ('IsSplitter プロパティ' in the on-line documentation)** プロパティを **True** に設定します。

メモ:テーブル領域を右クリックし、コンテキストメニューの[**スプリッタ**]を選択して、スプリッタを追加することもできます。

最初の列の右側にある濃い灰色のバーに注目してください。これは、このスプリッタを実行時にユーザーがドラッグできることを示しています。ドラッグすると、グリッドの最初の2列のラベルとテキストボックスが自動的にサイズ変更されます。

固定サイズテーブル領域の設定

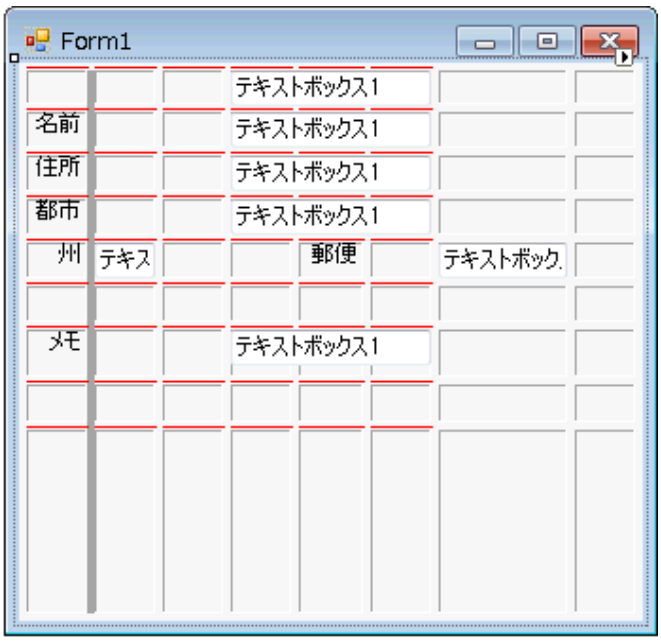
コントロールがサイズ変更されると、テーブル領域の比率を維持するようにグリッドレイアウトが調整されます。特定のテーブル領域を固定サイズとして指定し、サイズ変更されないようにすることができます。

たとえば、下端のテキストボックスだけを垂直方向にサイズ変更できるようにし、他のテキストボックスは一定の高さに維持できます。

それには、グリッドエディタを表示し、次に示すように、最初の6行を選択して **Band.IsFixedSize ('IsFixedSize プロパティ' in the on-line documentation)** プロパティを **True** に設定します。

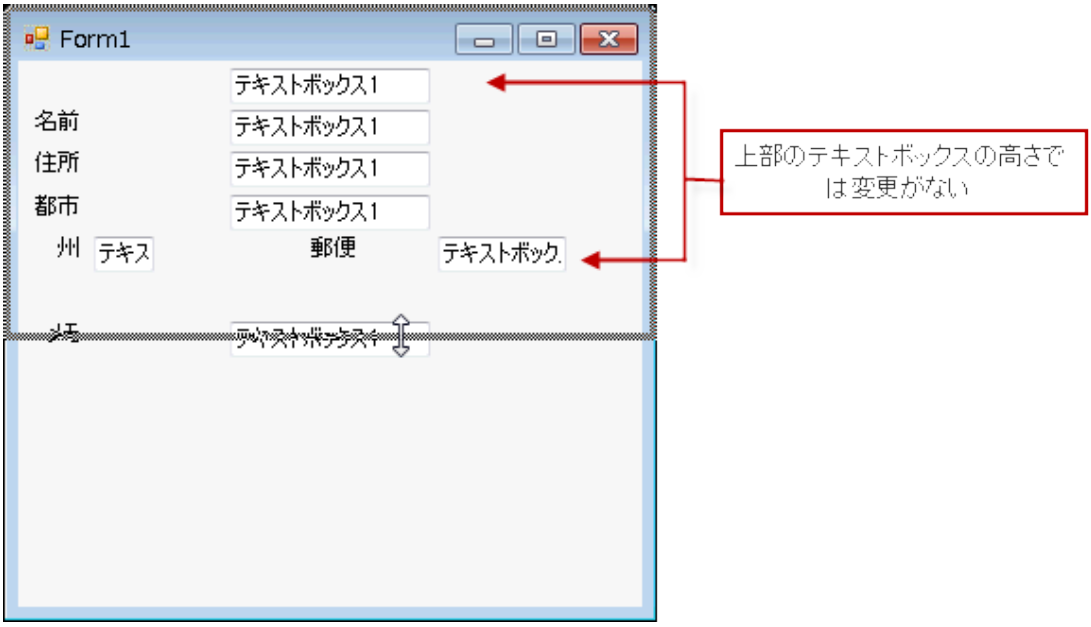
メモ:テーブル領域を右クリックし、コンテキストメニューの[**サイズを固定する**]を選択して、固定サイズのテーブル領域を設定することもできます。

OK ボタンをクリックします。最初の6行に赤色のマーカーが表示され、これらの行のサイズがフォームとともに変更されない(最後の2行のサイズが変更する)ことを示します。



プログラムを実行します。

ラベルとテキストボックスのサイズを調整するには、スプリッタを使用します。フォームをリサイズして上部のテキストボックスが高さを保持することを確認します。

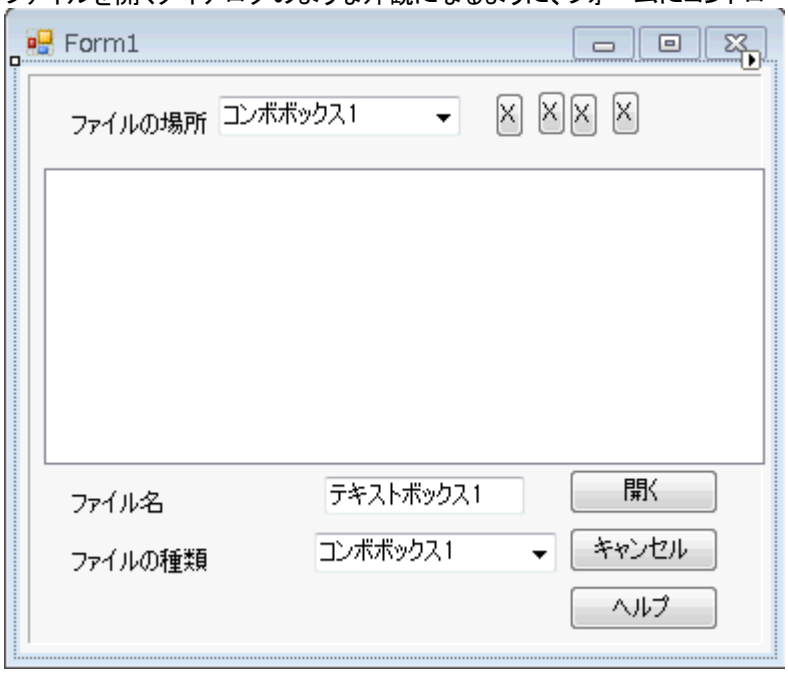


これで、チュートリアル1が終わります。次のチュートリアルでは、フォームにコントロールを追加して、C1Sizer グリッドを設定する方法について説明します。

チュートリアル 2: コントロールを追加してからグリッドを設定する

前のチュートリアルでは、グリッドを設定してからコントロールを追加しました。逆の順序で作業することもできます。最初はグリッドのことは考えず、フォームを通常の方法で設計します。フォームを作成したら、**C1Sizer** によってグリッドを自動作成します。もちろん、グリッドの作成後も、以前に使用した設計時コマンドを使ってグリッドを変更し、グリッドの外観や動作を調整できます。

1. Visual Studio を開いて新しいプロジェクトを作成した後、**C1Sizer** コントロールをフォームに追加し、**Dock** プロパティを **DockStyle.Fill** に設定してフォーム全体を使用するようにします。
2. ファイルを開くダイアログのような外観になるように、フォームにコントロールをいくつか追加します。

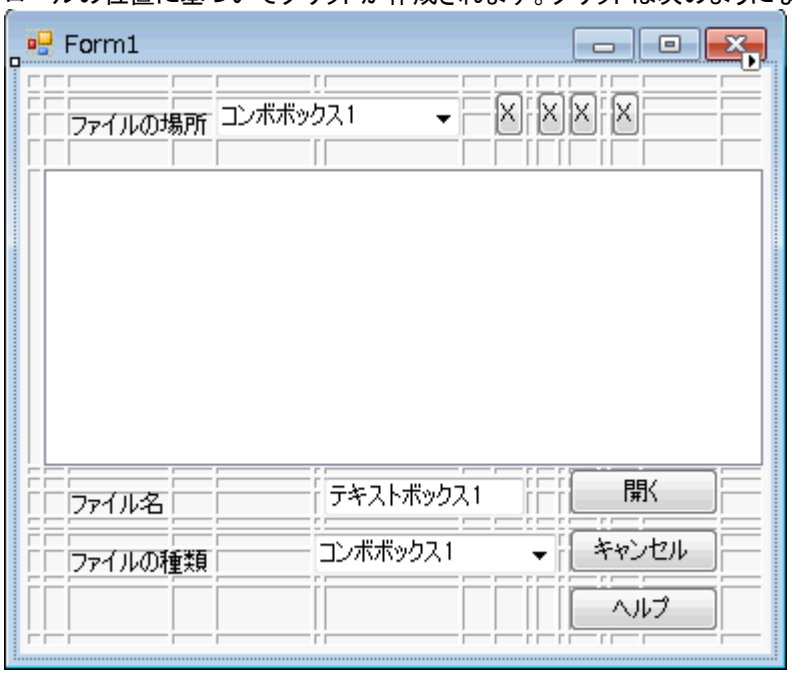


この新しいフォームの設定では、Visual Studio のすべてのレイアウトコマンドを使用できます。ここまで特に新しいことはありません。

3. 次に、C1Sizer コントロールの **SplitterWidth ('SplitterWidth プロパティ' in the on-line documentation)** プロパティを小さい値（1または2ピクセル）に変更します。フォーム上のいくつかのボタンがきわめて近接しており、グリッドの

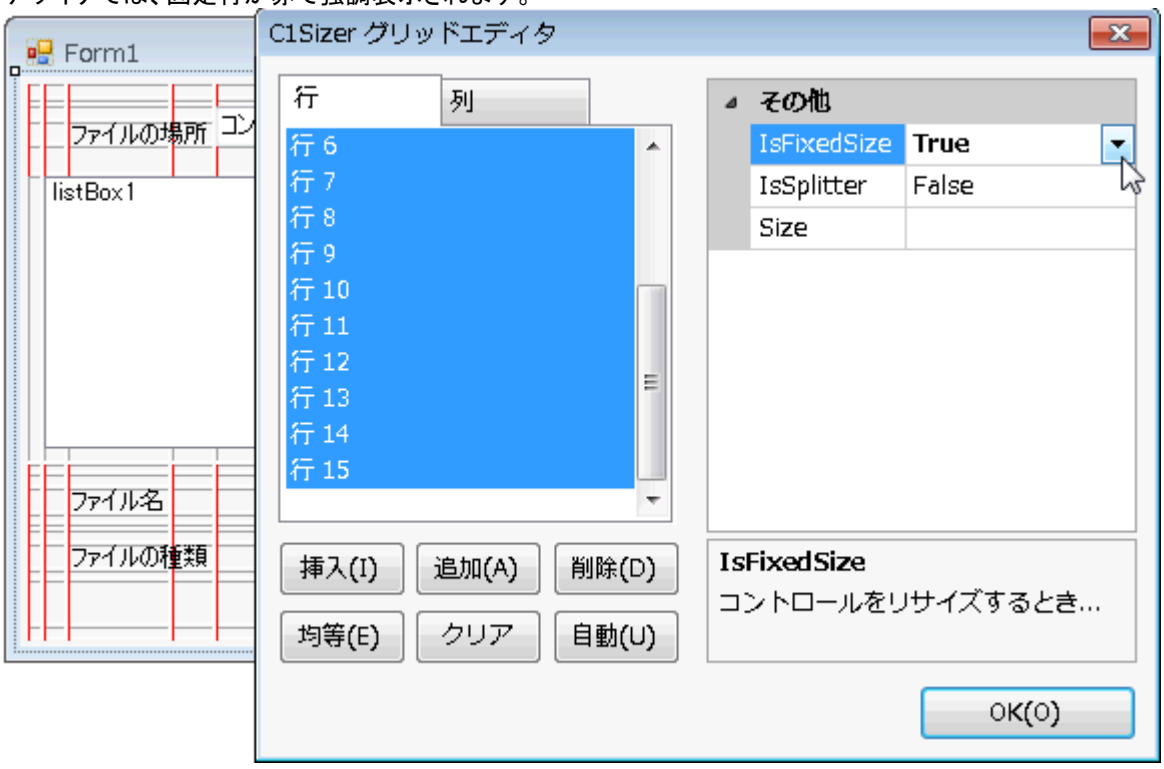
作成時にもその配置を維持するため、このような設定をお勧めします。

4. グリッドを作成するには、C1Sizer コントロールを右クリックし、[グリッドの自動設定]オプションを選択します。子コントロールの位置に基づいてグリッドが作成されます。グリッドは次のようになります:



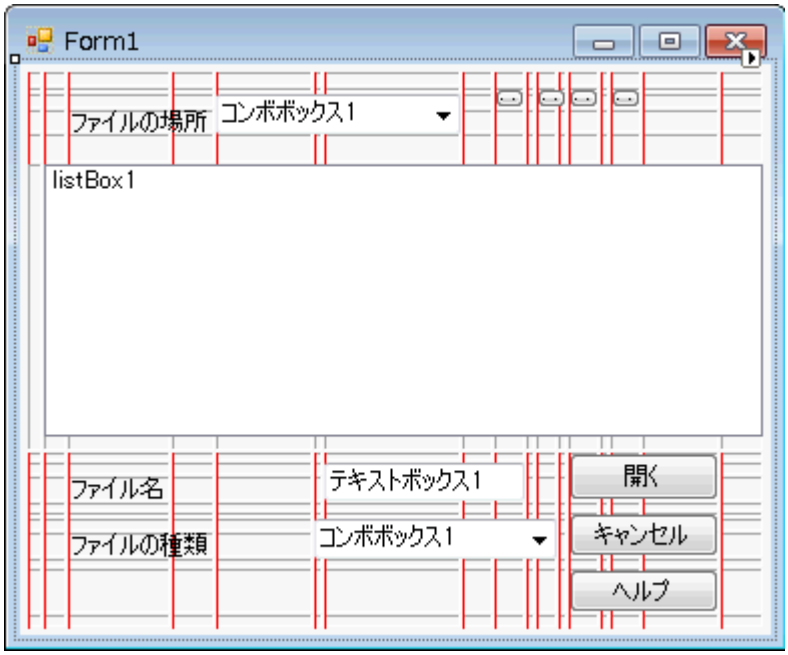
この段階でプロジェクトを実行すると、フォームをサイズ変更したときに内部のコントロールが均等にサイズ変更されます。

5. ただし、この例では、ボタンとテキストボックスは垂直方向に拡大縮小せず、リストボックスだけ拡大縮小されるようにします。その場合は、グリッドエディタを表示し、次のように ListBox が含まれている行を除くすべての行を選択して、**Band.IsFixedSize ('IsFixedSize プロパティ' in the on-line documentation)** プロパティを True に設定します。デザイナでは、固定行が赤で強調表示されます。



6. **OK** を選択します。

デザイナでは、固定行が赤で強調表示されます。



アプリケーションを実行してフォームをサイズ変更してみます。コントロールがグリッドにスナップし、レイアウトが自動的に調整されることを確認してください。

タスク別ヘルプ

タスク別ヘルプは、Visual Studio のプログラミングに習熟し、**C1Sizer ('C1Sizer クラス' in the on-line documentation)** と **C1SizerLight ('C1SizerLight クラス' in the on-line documentation)** コントロール/コンポーネントの一般的な使い方を知っていることを前提にしています。はじめて **C1Sizer for WinForms** を使用する場合、まず、**C1Sizer** コントロールトピックのチュートリアルを参照してください。

各トピックでは、**C1Sizer for WinForms** を使用する個々のタスクのソリューションを提供しています。また、各タスク別ヘルプトピックでは、新規 .NET プロジェクトが作成されていることを前提にしています。

プログラムでフォームに C1SizerLight コンポーネントを追加する

実行時に **C1SizerLight ('C1SizerLight クラス' in the on-line documentation)** コンポーネントをフォームに追加するには、以下のコードを使用します。

▶ Visual Basic コードの書き方

Visual Basic

```
If Me.components Is Nothing Then
Me.components = New System.ComponentModel.Container
End If
Dim szl As C1SizerLight = New C1SizerLight(components)
szl.SetAutoResize(Me, True)
```

▶ C# コードの書き方

C#

```
if (this.components == null)
this.components = new System.ComponentModel.Container();
C1SizerLight szl = new C1SizerLight(components);
szl.SetAutoResize(this, true);
```

実行時に、複数のフォームにより呼び出されたジェネリックフォームスタートアップ関数を作成するには、以下のコードを使用します。

▶ Visual Basic コードの書き方

Visual Basic

```
Dim c As Container = New System.ComponentModel.Container
     For Each f As Form In myFormList
     Dim szl As C1SizerLight = New C1SizerLight(c)
     szl.SetAutoResize(f, True)
Next
```

▶ C# コードの書き方

C#

```
Container c = new System.ComponentModel.Container();
    foreach (Form f in myFormList)
    {
        C1SizerLight szl = new C1SizerLight(c);
        szl.SetAutoResize(f, true);
```

```
    }
```

実行時に C1Sizer のグリッド上にコントロールを配置する

実行時にコントロールを **C1Sizer ('C1Sizer クラス' in the on-line documentation)** グリッドに配置するには、指定したコントロールを希望の場所に移動します。**C1Sizer** は、グリッドの一番近い位置にコントロールをスナップして、フォームのサイズが変更されたときにコントロールを自動的にリサイズします。コントロールは、単一セルまたは複数のセル上に配置することができます。

1. **C1Sizer** コントロールをフォームに追加します。
2. C1Sizer コントロールを右クリックして、[**グリッドの編集**]を選択します。
3. **C1Sizer グリッドエディタ** に3行と3列追加して、[**OK**] ボタンをクリックします。

Band.Bounds プロパティを使用してグリッドセルの位置を決定する、または **Control.Bounds** プロパティを使用してコントロールを移動するには、以下のコードを使用します。

▶ Visual Basic コードの書き方

Visual Basic

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs)
 ' ボタンを作成して、セル(0,0)に配置します。
 Dim b1 As Button = New Button
 c1Sizer1.Controls.Add(b1)
 b1.Visible = True
 b1.Bounds = GetCellRectangle(0, 0)
 ' セル範囲(1,1)～(2,2)の位置を計算します。
 Dim rc As Rectangle = GetCellRectangle(1, 1)
 rc = Rectangle.Union(rc, GetCellRectangle(2, 2))
 ' ボタンを作成して、セル範囲(1,1)-(2,2)に配置します。
 Dim b2 As Button = New Button
 c1Sizer1.Controls.Add(b2)
 b2.Visible = True
 b2.Bounds = rc
End Sub

Private Function GetCellRectangle(ByVal row As Integer, ByVal col As Integer) As
Rectangle
  Return Rectangle.Intersect(c1Sizer1.Grid.Rows(row).Bounds,
c1Sizer1.Grid.Columns(col).Bounds)
End Function
```

▶ C# コードの書き方

C#

```
private void Form1_Load(object sender, System.EventArgs e)
{
    //ボタンを作成して、セル(0,0)に配置します。
    Button b1 = new Button();
    c1Sizer1.Controls.Add(b1);
    b1.Visible = true;
    b1.Bounds = GetCellRectangle(0,0);
   //セル範囲(1,1)～(2,2)の位置を計算します。
    Rectangle rc = GetCellRectangle(1,1);
```

```
    rc = Rectangle.Union(rc, GetCellRectangle(2,2));
   //ボタンを作成して、セル範囲(1,1)-(2,2)に配置します。
    Button b2 = new Button();
    c1Sizer1.Controls.Add(b2);
    b2.Visible = true;
    b2.Bounds = rc;
}
private Rectangle GetCellRectangle(int row, int col)
{
        return Rectangle.Intersect(
        c1Sizer1.Grid.Rows[row].Bounds,
        c1Sizer1.Grid.Columns[col].Bounds);
}
```

行および列の 3D 枠線を作成する

C1Sizer ('C1Sizer クラス' in the on-line documentation) コントロールの行と列の境界線のスタイルを作成するには、次の手順に従います。**C1Sizer.Paint** イベントにて、**DrawBorder3D** メソッドを使用して、**C1Sizer**コントロールの列と行のエッチングされた3D境界線を描画して、本コントロールの行と列を再描画します

1. **C1Sizer** コントロールをフォームに追加して、**Dock** プロパティを **DockStyle.Fill** に設定すると、コントロールはフォーム全体に表示されます。
2. **C1Sizer** パネルにコントロールを追加します。例えば、2つのボタンコントロールを追加します。
3. パネルを右クリックして[**グリッドの自動設定**]を選択し、グリッドのレイアウトを作成または有効にします。
4. 下記のコードを **C1Sizer1_Paint** イベントに記載して、行と列の境界線を3Dスタイルに変更します。

▶ Visual Basic コードの書き方

Visual Basic

```
Private Sub C1Sizer1_Paint(ByVal sender As Object, ByVal e As PaintEventArgs)
' C1Sizerグリッドを描画します。
 For Each row As Band In Me.c1Sizer1.Grid.Rows
   Dim rcrow As Rectangle = row.Bounds
   For Each col As Band In Me.c1Sizer1.Grid.Columns
     Dim rccol As Rectangle = col.Bounds
     Dim rccel As Rectangle = Rectangle.Intersect(rcrow, rccol)
     ControlPaint.DrawBorder3D(e.Graphics, rccel, Border3DStyle.Etched)
   Next
 Next
End Sub
```

▶ C# コードの書き方

C#

```
        private void c1Sizer1_Paint(object sender, PaintEventArgs e)
        {
            // C1Sizerグリッドを描画します。
            foreach (Band row in this.c1Sizer1.Grid.Rows)
            {
                Rectangle rcrow = row.Bounds;
                foreach (Band col in this.c1Sizer1.Grid.Columns)
                {
```

```
                    Rectangle rccol = col.Bounds;
                    Rectangle rccel = Rectangle.Intersect(rcrow, rccol);
                    ControlPaint.DrawBorder3D(e.Graphics, rccel,
                        Border3DStyle.Etched);
                }
            }
        }
```

コントロールのレイアウト情報を保存する

C1Sizer ('C1Sizer クラス' in the on-line documentation) コントロールのレイアウト情報を保存するには、**GridDefinition ('GridDefinition プロパティ' in the on-line documentation)** プロパティを使用して、グリッド情報を含む文字列を取得または設定します。このプロパティは、ブラウズできませんが、ほかのプロパティとして使うことができます。

以下のコードサンプルは、C1.Win.C1Sizer と System.Collections の名前空間をソースコードにインポートしたことを前提にしています。グリッド定義とコントロール位置を保存または復元する二つのボタンを実行するには、以下のコードを使用します。

▶ Visual Basic コードの書き方

Visual Basic

```
Private _gridDefinition As String
Private _ctlPositions As ArrayList
 ;
Private Sub btnSaveGrid_Click(ByVal sender As Object, ByVal e As System.EventArgs)
'グリッドの定義を設定します。
_gridDefinition = C1Sizer1.GridDefinition
'コントロールの位置を保存します。
_ctlPositions = New ArrayList
For Each ctl As Control In C1Sizer1.Controls
_ctlPositions.Add(ctl.Bounds)
Next
EndSub
Private Sub btnRestoreGrid_Click(ByVal sender As Object, ByVal e As System.EventArgs)
If Not(_gridDefinition Is Nothing) Then
'グリッドの定義設定を元に戻します。
c1Sizer1.GridDefinition = _gridDefinition
Dim i AsInteger = 0
'コントロールの位置を元に戻します。
  While i< c1Sizer1.Controls.Count
c1Sizer1.Controls(i).Bounds = CType(_ctlPositions(i), Rectangle)
   System.Math.Min(System.Threading.Interlocked.Increment(i),i-1)
;  End While
EndIf
End Sub
```

▶ C# コードの書き方

C#

```
        string _gridDefinition;
        ArrayList _ctlPositions;
        private void btnSaveGrid_Click(object sender, System.EventArgs e)
```

```
{
    // グリッドの定義を設定します。
    _gridDefinition = c1Sizer1.GridDefinition;
    // コントロールの位置を保存します。
    _ctlPositions = new ArrayList();
    foreach (Control ctl in c1Sizer1.Controls)
    {
        _ctlPositions.Add(ctl.Bounds);
    }
}
private void btnRestoreGrid_Click(object sender, System.EventArgs e)
{
    if (_gridDefinition != null)
    {
        // グリッドの定義設定を元に戻します。
        c1Sizer1.GridDefinition = _gridDefinition;
        // コントロールの位置を元に戻します。
        for (int i = 0; i < c1Sizer1.Controls.Count; i++)
        {
            c1Sizer1.Controls[i].Bounds = (Rectangle)_ctlPositions[i];
        }
    }
}
```