

# SharpShooter Reports Documentation

---

Last modified on: November 15, 2012

※本ドキュメント内のスクリーンショットは英語表記ですが SharpShooter Reports JP(日本語版)では日本語で表示されます。



## 目次

SharpShooter Reports の基本的な使い方 .....	5
主な機能 .....	5
エディション .....	5
運用 .....	6
基本的な使い方 .....	6
基礎知識 .....	17
基本用語 .....	17
レポート作成に使われる主なコンポーネント .....	18
データソースの使用 .....	18
スクリプトの使用 .....	19
オブジェクトをデータにバインドする .....	19
ドキュメントの生成 .....	19
最終ドキュメントのビジュアル化と編集 .....	20
エクスポート、印刷、保存 .....	20
レポート デザイナ インターフェイス .....	21
タブ .....	21
コンテキスト タブ .....	28
アプリケーション メニュー .....	39
バインド バー .....	40
クイック アクセス ツールバー .....	40
コンテキスト メニュー .....	41
ドッキング .....	46
エディタ .....	48
ウィザード .....	59
ステータス バー .....	65
ホットキー .....	66
テンプレートのオブジェクト .....	68
分類 .....	68



バンド .....	68
ビジュアル オブジェクト .....	69
ユーザーガイド .....	71
テンプレート .....	71
ページ .....	73
オブジェクトの操作 .....	78
オブジェクトのプロパティ .....	78
テキスト出力 .....	94
リスト .....	99
グループ化.....	107
水平方向のリストと表.....	110
<b>SideBySide</b> バンド .....	111
ピボットテーブル .....	112
イメージ .....	118
シェープの使用 .....	121
バーコード.....	123
<b>Parameters</b> .....	125
ハイパーリンクとレポートのナビゲーション .....	126
サブレポートの使用 .....	130
<b>Zip</b> コード.....	134
マスターレポート .....	135
ウィザードを使ったレポートの作成 .....	137
本製品の特徴を使用する .....	148
スクリプト.....	148
データソース .....	157
エクスポート .....	168
プロジェクト .....	174
本製品の高度な特徴を使用する .....	178
カスタム レポート デザイナ .....	178



カスタム レポート ビューア .....	179
カスタム ウィザード .....	180
カスタム レポート コントロール.....	180
カスタム エクスポート フィルタ .....	181
カスタム テンプレートの保存 .....	181
ライブ レポート .....	181
レポート生成の操作 .....	182
付録 1 ドキュメント オブジェクト モデル .....	184



## SharpShooter Reports の基本的な使い方

SharpShooter Reports はさまざまな種類のソースを基にして複雑なレポートを作成したり、ドキュメントを色々なファイル形式にエクスポートすることができるフレキシブルな .NET 対応のレポートジェネレータです。本製品のコンポーネントは Windows フォーム アプリケーション、ASP.NET ページや Silverlight アプリケーションで使用できます。パッケージには、既存のレポートテンプレートやドキュメントの編集が行えるデザイナーが含まれています。レポートにさまざまな要素を含めることができます。内蔵されたピボットテーブルコンポーネントを使用して動的なデータ分析が行えます。レポートで使用するデータをチャートや図形で表すこともできます。

### 主な機能

- .NET4.0/4.5 完全対応。
- 完全にマネージドコード。
- .NET の統合開発環境 Visual Studio2005/2008/2010/2012 に対応。
- Win フォーム、ASP.NET の Web フォーム、Silverlight のサポート。
- ADO.NET 完全対応。
- 任意の .NET オブジェクトをデータソースとしてサポート。
- レポート生成ウィザード。
- 任意の .NET 言語を使った式やスクリプトの作成。
- データ バインディング モデル。
- XML ベースのレポートファイルにより、インターネットを使ったレポート転送処理が簡単。
- Win フォーム、ASP.NET の Web フォーム、Silverlight のレポートビューアコンポーネント。
- エンドユーザー レポートデザイナー
- 一つのレポートにデータソースを無制限に使用可能。
- レポートにバンドを無制限に組み込んだり連続での使用が可能。
- ピボットテーブルコンポーネントを内蔵。
- パラメータ化されたレポートの作成および使用。
- パラメータ化されたサブレポートの作成および使用。
- 段組みレポートを作成するための SideBySide の高度な概念。
- 短時間で統一されたレポートの生成が可能。
- 生成したレポートをオブジェクトグラフとして保存。
- 完全なドキュメントオブジェクトモデルは使用が簡単で、レポートテンプレートでも既存のドキュメントでも使用可能。
- バウンド/アンバウンドレポートのサポート。
- ほとんどのファイル形式へのエクスポートをサポート。
- マニュアルセクションの出力管理（任意）。
- Crystal Report のインポートオプション。
- Zip コードやバーコードの作成をサポート。
- さまざまなビジュアルオブジェクトを内蔵。
- GDI+ を使用するための機能が向上。
- さまざまなページサイズやレイアウトのページを無制限にレポートに組み込むことが可能。
- ヘッダーやフッタの有効利用。
- 背景に透かしを挿入可能。
- スタイルを使用可能。
- 複数の計測システムをサポート。
- 詳細な説明や豊富なサンプルを同梱。
- アプリケーションのデプロイメントが簡単。
- アプリケーションへの SharpShooter Reports のランタイムライセンスは、Win フォーム、ASP.NET の Web フォーム、Silverlight などの形態にかかわらず無償。

### エディション

SharpShooter Reports には次のエディションがあります。



**SharpShooter Reports.Win** – Windows フォームでレポートを作成/表示できる機能。レポートビューアとレポートデザイナー（エンドユーザーレポートデザイナーでレポートテンプレートや最終ドキュメントの編集が可能）を含んでいます。

**SharpShooter Reports.Web** – Web アプリケーションでレポートを作成/表示できる機能。本製品は、レポートを Windows、Linux、Mac OS、あるいは iOS や Android タブレット/フォンにまで送信する機能を提供できる、真のクロスプラットフォームのレポートビューアです。

**SharpShooter Reports.WPF** – 何度かマウスをクリックするだけで、複数のデータソースから複雑なレポートも作成でき、Windows Presentation Foundation に送ることができる強力な帳票コンポーネントです。

**SharpShooter Reports.Silverlight** –Silverlight アプリケーションでレポートを作成/表示する機能を提供します。このエディションにはレポートビューア（Silverlight アプリケーションでレポートを表示するためのコンポーネント）が含まれています。

**SharpShooter Reports.Enterprise** – Windows フォーム、Web、Silverlight アプリケーションでレポートを作成/表示できます。パッケージには必要なすべてのコンポーネント（SharpShooter Reports.Win、SharpShooter Reports.Web、SharpShooter Reports.Silverlight）が含まれています。

**SharpShooter Reports.Professional** – Windows フォーム、Web、Silverlight アプリケーションとしてレポートを作成/表示できるエディションで、製品コンポーネントのソースコード一式が含まれています。

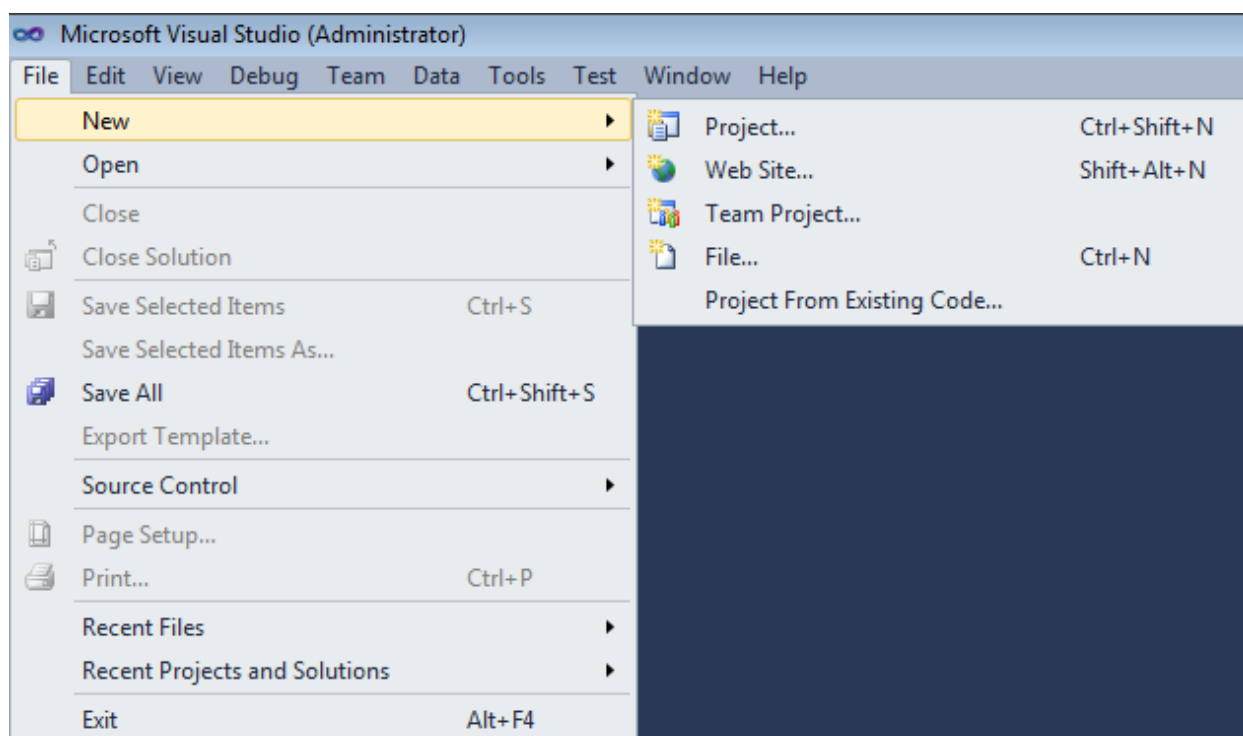
## 運用

本製品のコンポーネントを使用したプロジェクトを運用するために特別なことは全く必要ありません。コンピュータのファイルシステムまたはウェブサイトには、コンパイルしたファイルをコピーするだけです。本製品のインストールもクライアントコンピュータへのライセンス認証も必要ありません。開発者のコンピュータで本製品のライセンス認証を行うだけです。ライセンスはコンパイル時にアプリケーションに組み込まれます。

## 基本的な使い方

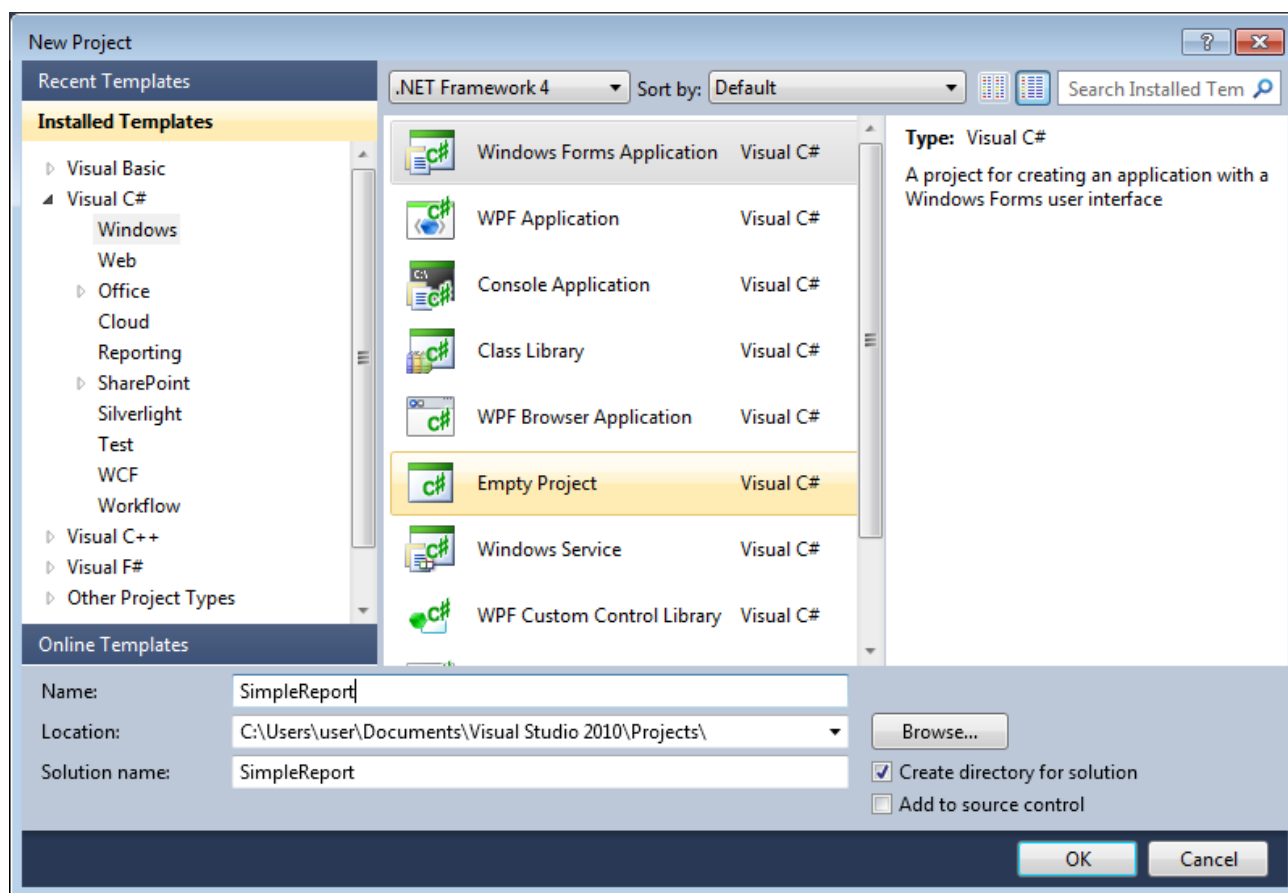
現在日付、テキスト、ロゴを表示するレポートテンプレート。

1. Microsoft Visual Studio で新規プロジェクトを作成します。それには、メインメニューから[ファイル/新規作成/プロジェクト]を選択します。

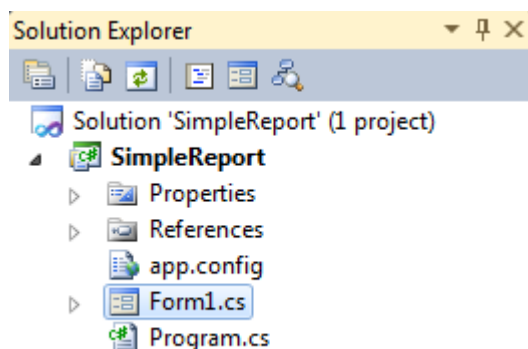




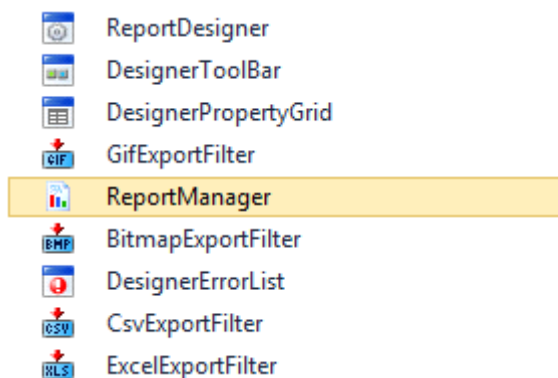
Windows アプリケーション プロジェクトのテンプレートを選択し、プロジェクト名を「SimpleReport」に設定し、プロジェクトを保存するフォルダを指定します。



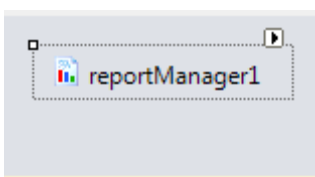
2. ソリューションエクスプローラの「Form1.cs」をダブルクリックしてエディタのウィンドウにアプリケーションのメインフォームを開きます。



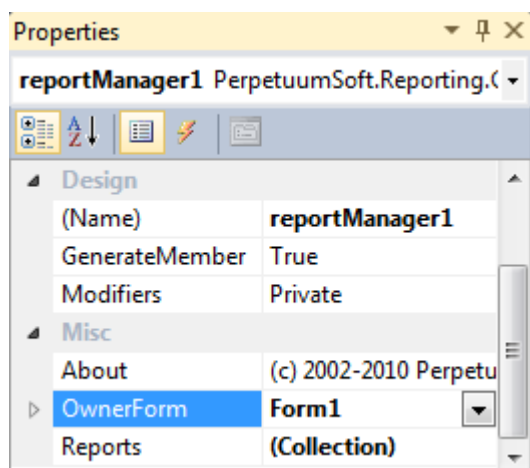
ツールボックスから「ReportManager」コンポーネントをドラッグ&ドロップします。このコンポーネントはレポートテンプレートやデータソースのコレクションを格納します。



このコンポーネントはウィンドウ下部に表示されます。



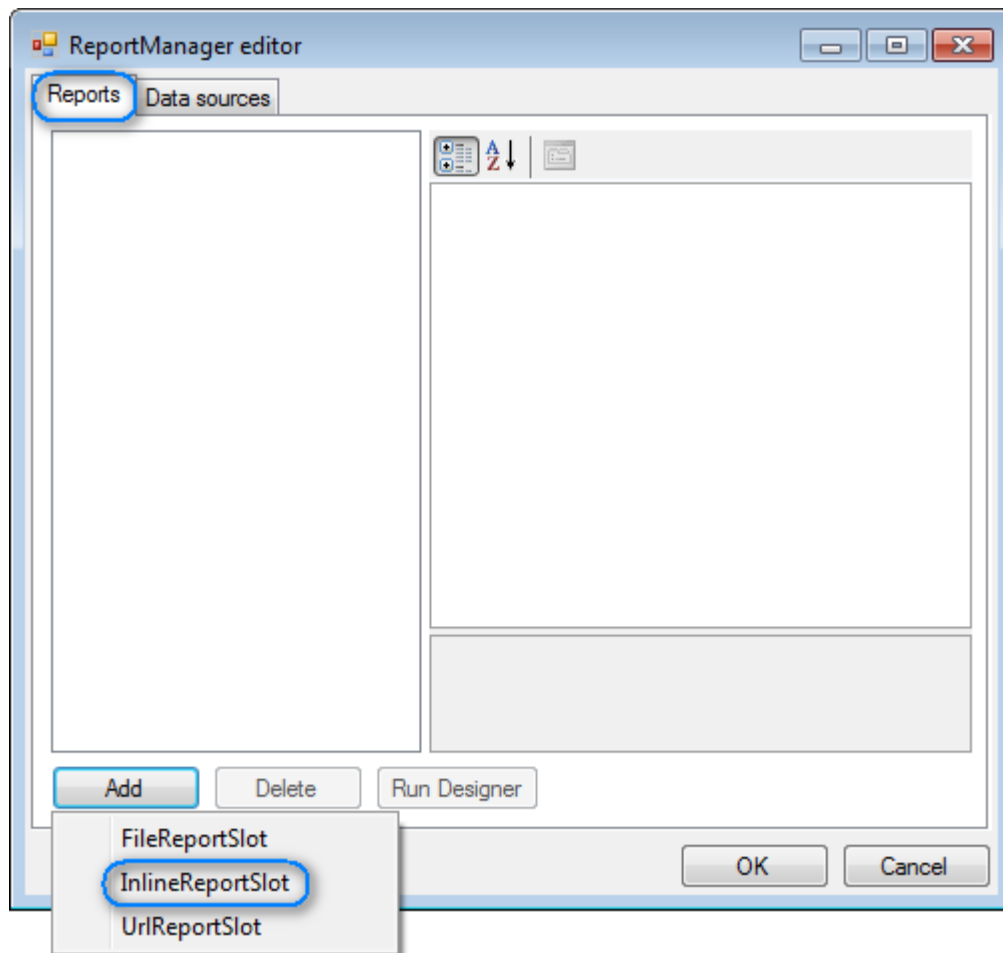
3. ReportManager コンポーネントを貼り付けたフォームを選択し、プロパティグリッドの ReportManager の OwnerForm プロパティを初期化します。





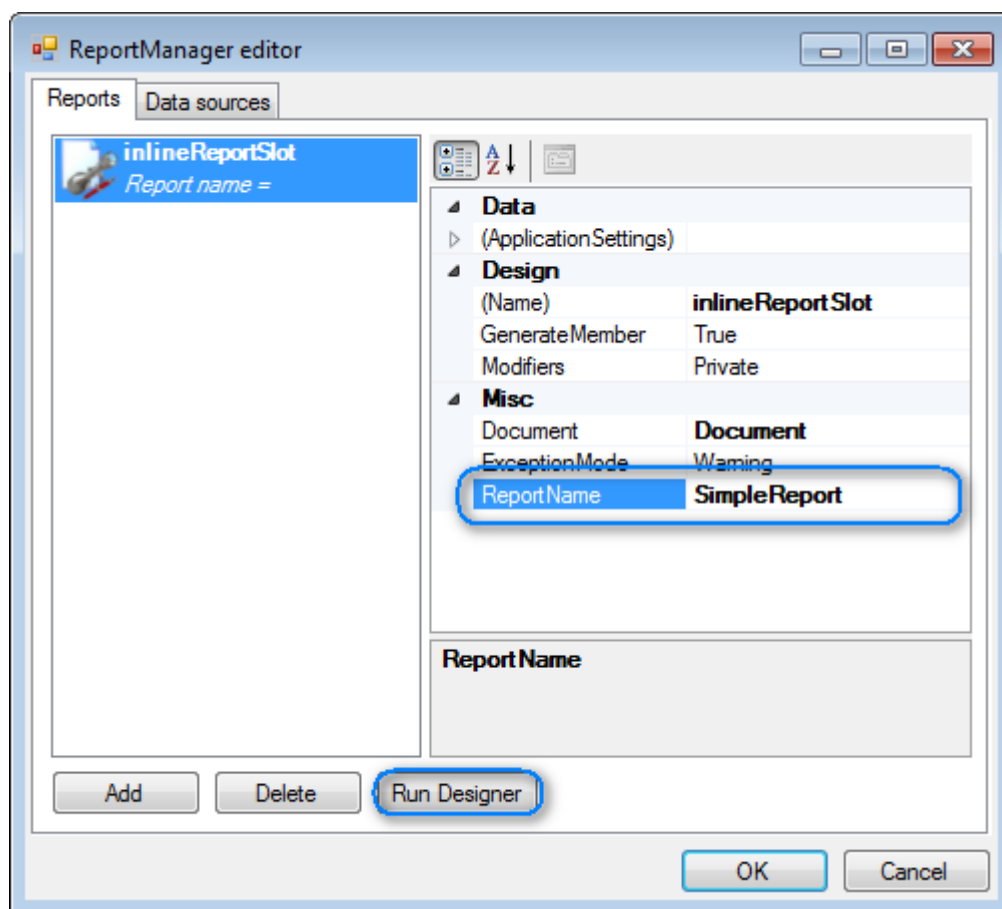


4. ReportManager コンポーネントをダブルクリックして ReportManager のエディタを開きます。

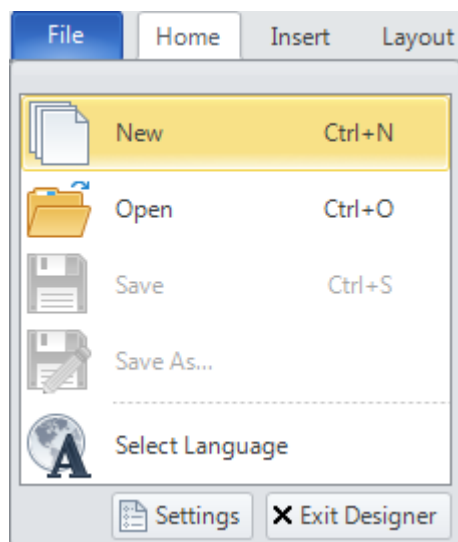


「レポート」タブの「追加」ボタンを押して「InlineReportSlot」を選択します。

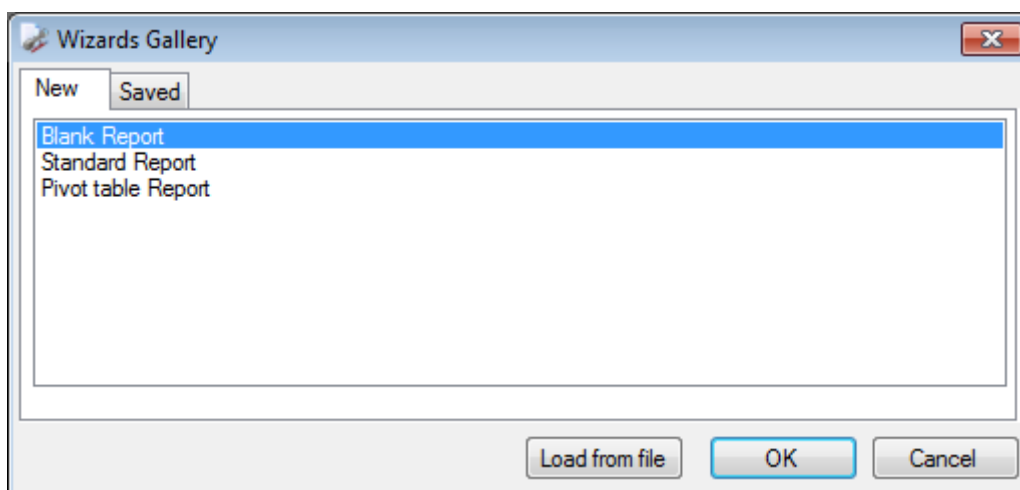
5. ReportName プロパティにレポート名 (SimpleReport) を設定し、「デザイナーの起動」ボタンを押してレポートデザイナーを開きます。



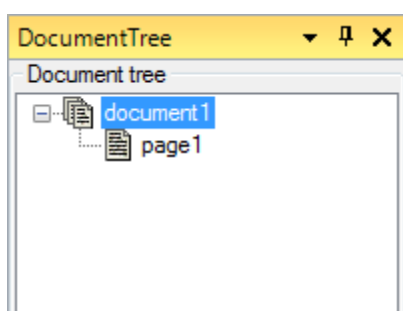
6. アプリケーションメニューの [ファイル/新規] を選択して、新しい白紙のテンプレートを作成します。



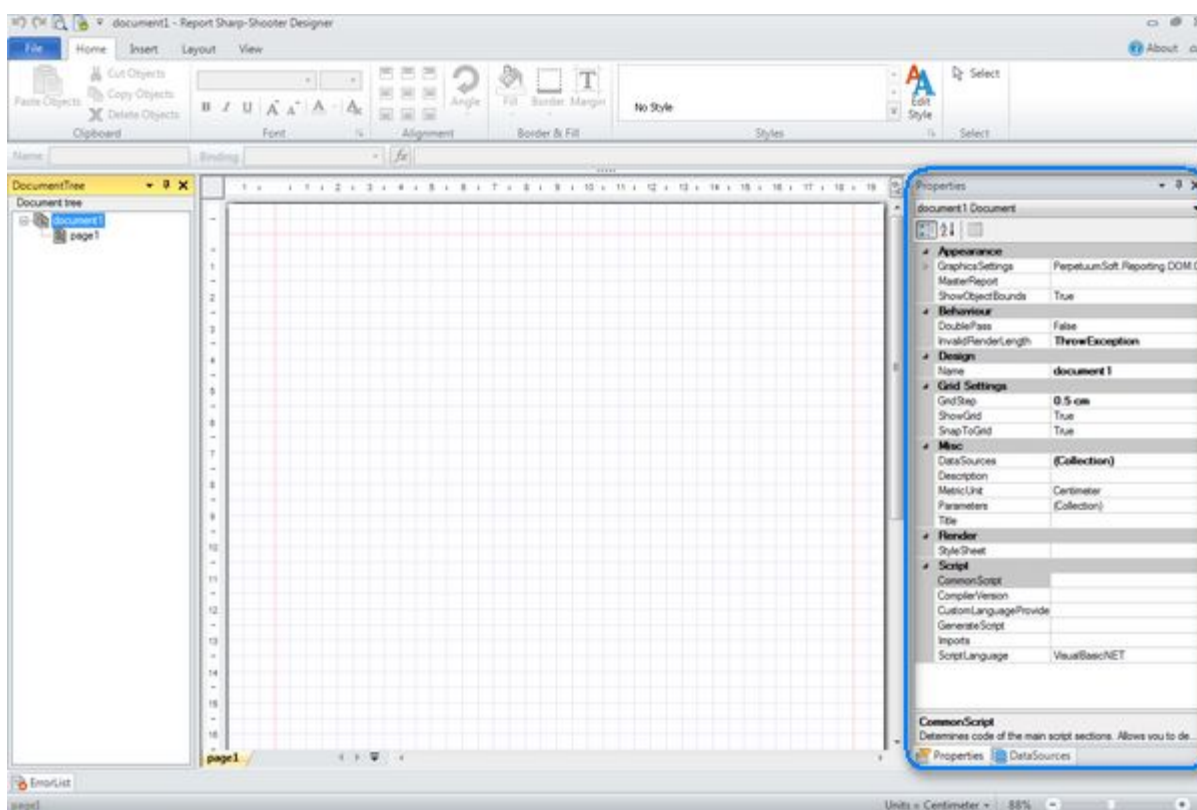
ウィザードギャラリーから「白紙」を選択し、「OK」ボタンを押します。



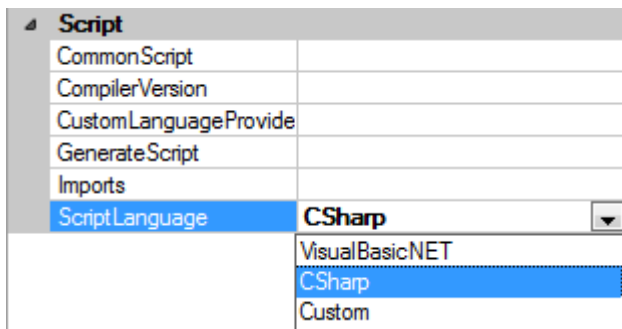
7. ドキュメントツリーの「document」を選択します。



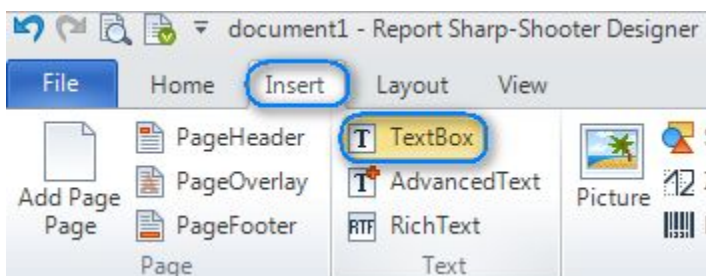
「プロパティ」ウィンドウに編集集中のテンプレートのプロパティが表示されます。



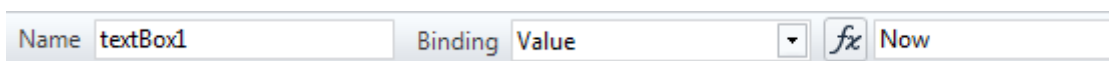
ScriptLanguage プロパティを「CSharp」に設定します。



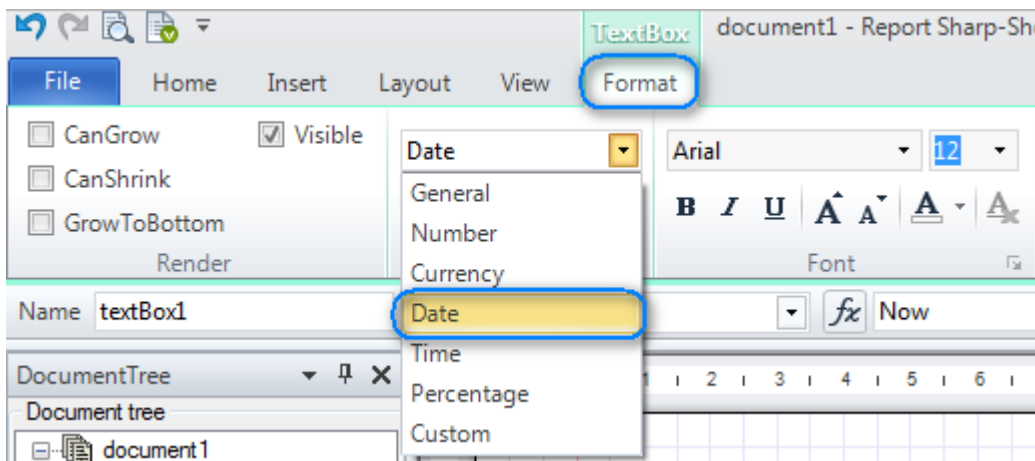
8. 「挿入」タブに行き、「TextBox」ボタンをクリックします。



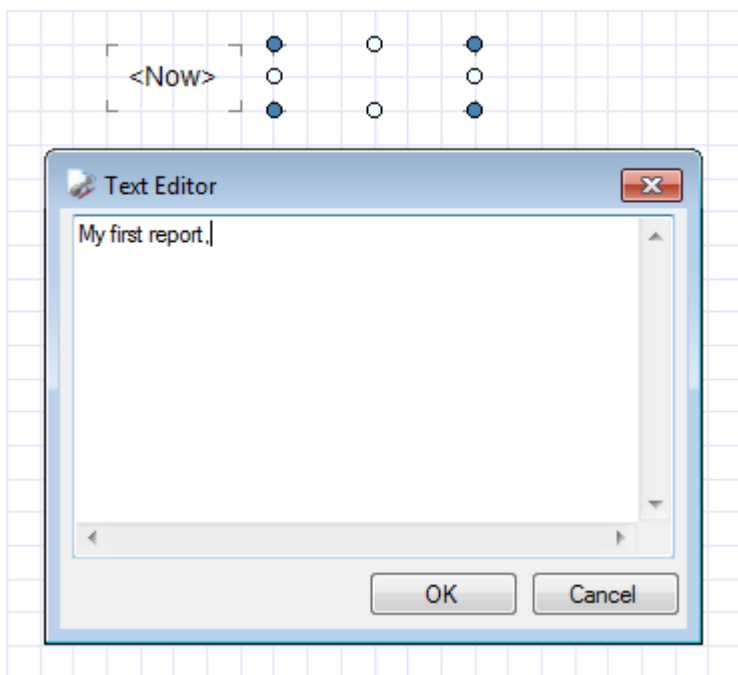
TextBox を左クリックしてテンプレートに追加し、Value プロパティを「Now」に設定します。



TextBox タブセットの「形式」タブに行き、TextFormat の一覧から「Date」を選択します。



9. テンプレートに **TextBox** コンポーネントをもう 1 つ追加します。(TextBox をダブルクリックするか、Enter キーを押して) テキストエディタを開き、「My first report」と入力します。



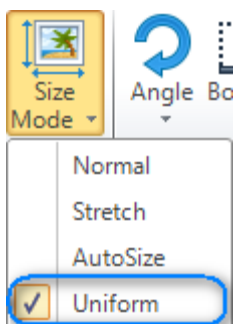
10. 「挿入」タブに行き、「Picture」ボタンをクリックします。



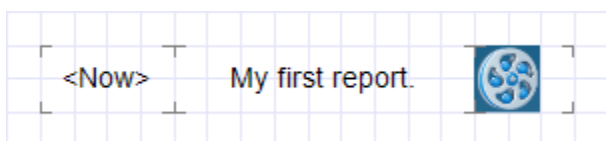
テンプレートを左クリックして **Picture** コントロールを追加します。

**Picture** オブジェクトをダブルクリックして、イメージのパスを設定する「開く」ダイアログウィンドウを開いて、ロゴを選択し、「開く」ボタンを押します。

**Picture** タブセットの「形式」タブに行き、「サイズモード」ボタンをクリックして、プロパティ値に「Uniform」を選択します。

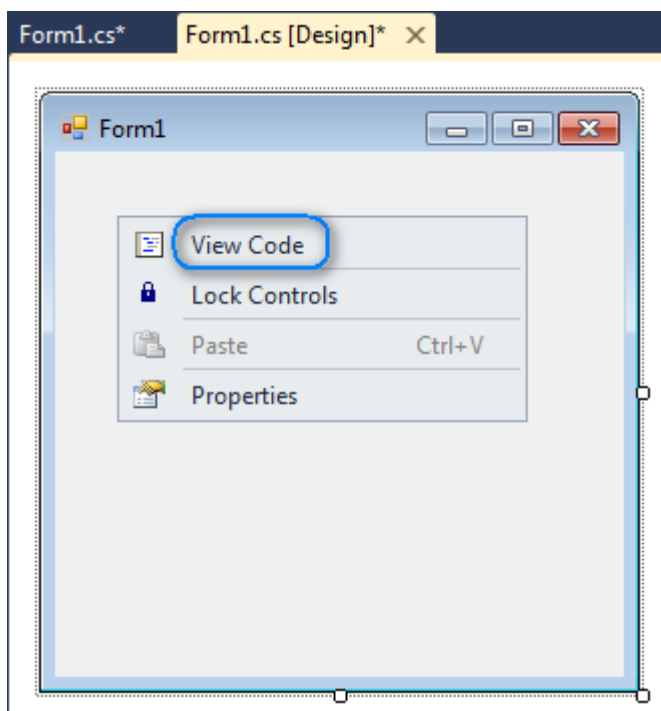


レポートテンプレート :



11. テンプレートを保存して、レポートデザイナーを閉じます。

12. フォームを右クリックして、コンテキストメニューから[コードの表示]を選択してコードに切り替えます。

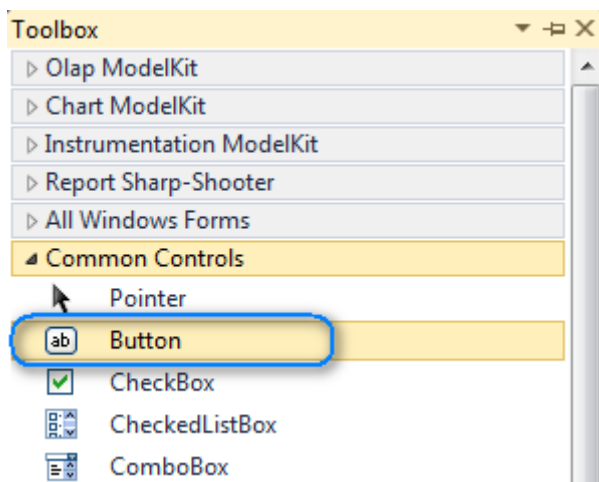


Form クラスのコンストラクタにレポートを表示するコードを追加します。InlineReportSlot オブジェクトの RenderCompleted イベントハンドラを作成します。

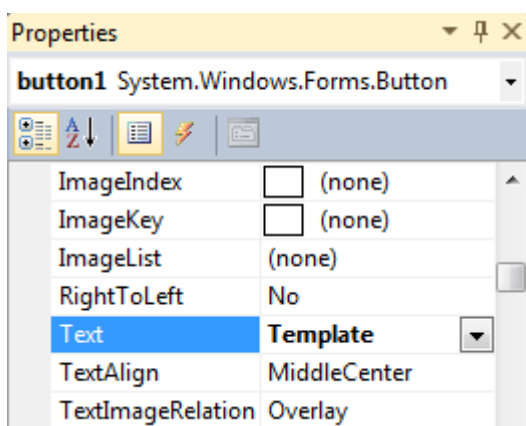
```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        inlineReportSlot.RenderCompleted += new EventHandler(reportSlot_RenderCompleted);
        inlineReportSlot.Prepare();
    }
    private void reportSlot_RenderCompleted(object sender, EventArgs e)
    {
        using (PerpetuumSoft.Reporting.View.PreviewForm previewForm = new
PerpetuumSoft.Reporting.View.PreviewForm(inlineReportSlot))
        {
            previewForm.WindowState = FormWindowState.Maximized;
            previewForm.ShowDialog(this);
        }
    }
}
```

12. Form1.cs[Design] タブをクリックしてアプリケーションのフォームに戻ります。

フォームにボタンを 2 つ追加します（ツールボックスの **Button** コンポーネントをフォームにドラッグ&ドロップします）。



フォームの **Button** コントロールを選択し、**Text** プロパティを変更します。最初のボタンの **Text** プロパティには「テンプレート」を、2つ目のボタンの **Text** プロパティには「レポート」を設定します。



フォームの **Button** コントロールをダブルクリックして、ボタンの **Click** イベントハンドラを作成します。このハンドラにレポートの生成を開始するコードを追加します。コード例：

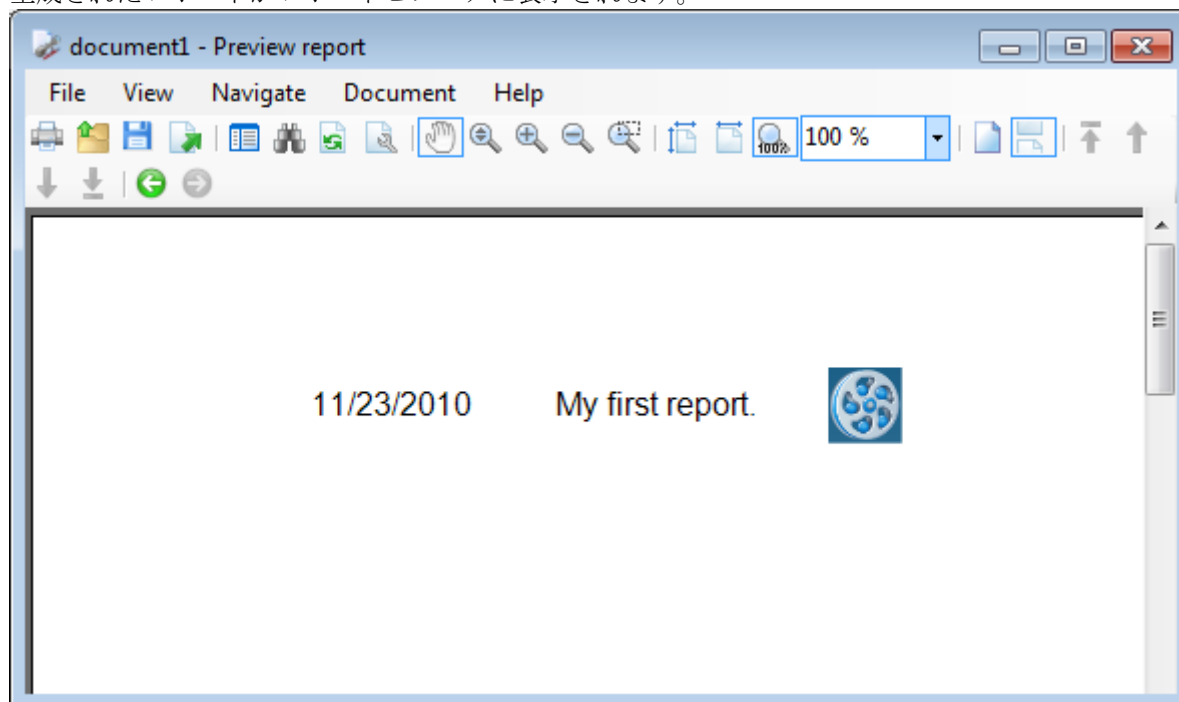
```
private void button1_Click(object sender, EventArgs e)
{
    inlineReportSlot.DesignTemplate();
}

private void button2_Click(object sender, EventArgs e)
{
    inlineReportSlot.Prepare();
}
```

13. Visual Studio のツールバーにある「デバッグ開始」ボタンを押してアプリケーションを実行します。



生成されたレポートがレポートビューアに表示されます。



「テンプレート」ボタンは、レポートを編集するためにテンプレートを開きます。「レポート」ボタンはレポートを生成し、レポートビューアに表示します。







## 基礎知識

### 基本用語

SharpShooter Reports で使用する基本用語について説明します。

#### レポートテンプレート

レポートテンプレートは、**IsTemplate** プロパティが「True」に設定されている、**PerpetuumSoft.Reporting.DOM.Document** クラスのオブジェクトです。レポートテンプレートには、複数のオブジェクトを格納する 1 つまたは複数のページが格納されます。最終レポートの見た目は、格納されているオブジェクトのレイアウトやプロパティの初期化によって設定されます。例えば、レポートのデータを取得するためのレポートのデータソースとデータの表示方法をを設定します。テンプレートには、レポート生成時に実行されるスクリプトが格納されます。テンプレートファイルは **xml** 形式のファイルで、拡張子は **.rst** です (**.rst** ファイルはテンプレートをレポートデザイナーから保存すると作成されます)。

**PerpetuumSoft.Reporting.Components.ReportSlot** クラスのオブジェクトは、レポートテンプレートを保存するために使用されます。**ReportSlot** オブジェクトの **LoadReport** や **SaveReport** メソッドはレポートテンプレートを取得したり保存したりするために使用されます。

#### レポートデザイナー

**ReportDesigner** コンポーネントは、テンプレートの作成や編集に使用します。テンプレートのプロパティやテンプレートのページのプロパティ設定、オブジェクトの追加、オブジェクトのレイアウトおよびプロパティの設定、レポートのデータソース設定、レポートの外観やレポート内容を定義する機能を提供します。

**ReportDesigner** コンポーネントを使用してテンプレートを保存したり、保存したテンプレートを開いてレポートの編集/編成を行うことができます。レポートデザイナーからレポートビューアを呼び出して、作成したレポートを表示できます。

#### レポートマネージャ

**ReportManager** コンポーネントは、レポートやデータソースのコレクションの保存に使用します。レポートテンプレートを保存するオブジェクトのコレクションには **Reports** プロパティからアクセスできます。データソースのコレクションには **DataSource** プロパティからアクセスできます。

#### 最終ドキュメント

最終ドキュメントは、**IsTemplate** プロパティが「False」に設定されている、**PerpetuumSoft.Reporting.DOM.Document** クラスのオブジェクトです。最終ドキュメントには、テキスト、チャート、ゲージ、またはイメージなど目に見える情報を含んだオブジェクトだけが含まれます。最終ドキュメントのファイルは **xml** 形式で、拡張子は **.rsd** です。

#### レポートビューア

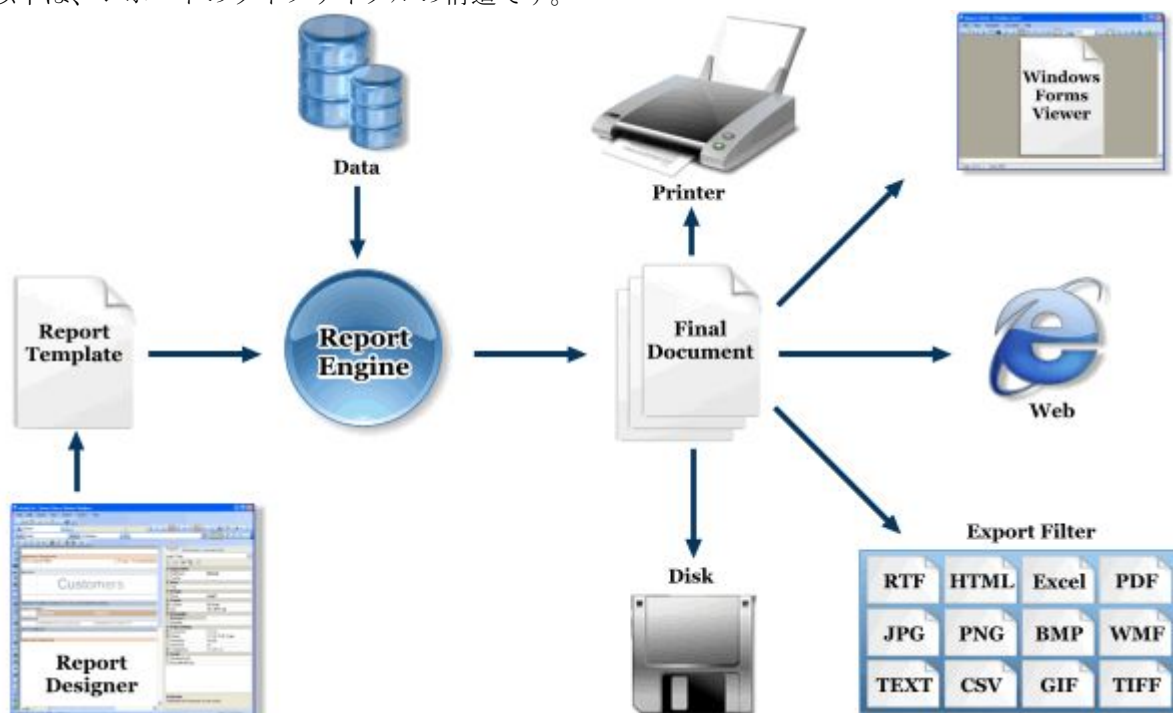
**ReportViewer** コンポーネントは、最終ドキュメントを表示するために使用します。最終ドキュメントでは、ズーム、検索、ブックマークのツリー表示、様々な表示パラメータ操作が可能です。レポートのエクスポート、印刷、ファイルへの保存が可能です。最終ドキュメントを編集するには、レポートデザイナーを起動してください。

#### エクスポート フィルタ

SharpShooter Reports は、最終ドキュメントを GIF、PNG、JPG、BMP、EMF、PDF、HTML、CSV、TXT、Excel、XML、XML for Excel、RTF などの一般的な形式にエクスポートする機能を提供します。

## インタラクションの構造

以下は、レポートのライフサイクルの構造です。



## レポート作成に使われる主なコンポーネント

最終ドキュメントの見た目は、レポートテンプレートのオブジェクトで決まります。どのオブジェクトもプロパティのセットがあります。プロパティの設定時にレポートの見た目が決まります。これらのプロパティに一定の値またはレポートの生成時に計算される式を設定でき、スクリプトをバインドしています。

レポートを作成するために使用されるオブジェクトはすべて、バンドとビジュアルオブジェクトの2種類に分けられます。

バンドは最終ドキュメントには表示されません。バンドは他のオブジェクトのコンテナで、バンドに格納されているオブジェクトの表示方法を設定するためのものです。それには、リストや表の作成、データソースの設定、グループ化や並べ替え条件の設定などがあります。一部のバンドは他のバンドのコンテナになります。

レポートのビジュアルオブジェクトは、テキスト、イメージ、Zip コード、バーコード、シェープ、Windows フォーム コントロールのイメージといったさまざまなデータを表示するために使用されます。ビジュアルオブジェクトのプロパティは、データの形式やスタイルを設定します。

## データソースの使用

ADO.NET オブジェクト、System.Collection.IEnumerable インターフェイスの System.ComponentModel.IListSource を実装するクラス、その他のデータソース。その他のデータソースの場合、渡されたオブジェクトのプロパティは使用されているレポートデータになります。

レポートマネージャを使用してデータソースを設定できます。データソースはレポートマネージャに格納されるすべてのレポートテンプレートに使用できます。加えて、データソースを特定のテンプレートに組み込むことができます。

組み込み式のデータソースは SqlDataSource、XmlDataSource、OdbcDataSource の3つがあります。



## スクリプトの使用

スクリプトは、レポート生成時に計算される式を設定するために使用されます。式の計算結果はプロパティ値として使用できます。スクリプトは、最終ドキュメントに要素を生成したり、レポート要素のレイアウトを手動で行うために使用できます。

スクリプトは一部のプログラミング言語では式になり、.NET の任意のプログラミング言語を使用できます。

レポート生成時に **SharpShooter Reports** を使用して、アプリケーションで使用可能な任意のクラスのインスタンスを作成したり使用したりすることができます。

## オブジェクトをデータにバインドする

データバインドは、データソースのデータを表示するレポートで使用され、パスが 2 つ実行されます。まず、データソースとして使用するオブジェクトが設定されます。次に、特定のオブジェクトにフィールドが設定されます。レポートのデータソースがコレクションのオブジェクトの場合、コレクションのオブジェクトのフィールドが設定され、データバンドの入れ子のオブジェクトがリストの表示に使用されます。オブジェクトがコレクションではない場合、レポートはオブジェクトのプロパティを表示できます。式をバインドすることで、最終的なオブジェクトがデータソースにバインドされます。

## ドキュメントの生成

レポートの生成には、**ReportSlot** オブジェクトの **Prepare** メソッドや **RenderDocument** メソッドが使用されます。これらの違いは、**Prepare** メソッドは個別のスレッドでレポート生成プロセスを行います。レポートマネージャには、レポート生成イベントを実行するフォームのスレッドを設定するための **OwnerForm** プロパティがあります。イベントハンドラがフォームに貼り付けられているオブジェクトを参照する場合、このプロパティの初期化が必要な場合があります。このようなハンドラを個別のスレッドで実行すると、**System.InvalidOperationException** 例外が発生する場合があります。

レポートの生成プロセスの制御や設定する機能を持たせるために、**Prepare** メソッドや **RenderDocument** メソッドのオーバーロードを使用することができます。これらのオーバーロードはオブジェクトをパラメータとして取得し、**IRenderSite** インターフェイスを実装します。このインターフェイスを実装するオブジェクトは、レポート生成プロセスの情報を取得するために使用できます。ただ、このインターフェイスで記述されたオブジェクトのメソッドを設定する必要があります。

テンプレートがデータソースからデータを取得するオブジェクトを使用している場合、そのオブジェクト名に設定されたデータソースがレポート生成時に検索されます。検索は、レポートマネージャに追加されたソース（現在のレポートスロットに格納されたソース）と、組み込み式のデータソースで検索されます。加えて、データソースの取得時にイベントが呼び出されます。そのイベントを使って動的にデータソースに接続できます。

レポート生成時にレポートパラメータが必要な場合があります。**GetReportParameter** イベントは、パラメータを取得するために使用します。このハンドラで指定したパラメータのコレクションを初期化する必要があります。このコレクションは、**PerpetuumSoft.Reporting.Components.GetReportParameterEventArgs** オブジェクトの **Parameters** プロパティから取得できます。このクラスのインスタンスはそのハンドラにパラメータとしてデリゲートされます。注意しなければならないことは、コントロールのプロパティからパラメータの値を取得する場合、対象のレポートマネージャの **OwnerForm** プロパティを初期化しなければならないということです。

レポートにサブレポートを使用する場合、対象のレポートマネージャに、**SubReport.TemplateName** プロパティと一致する **ReportName** プロパティを持ったレポートスロットを格納してください。入れ子のレポートを動的に作成する場合は、**ReportManager** の **ResolveSubReport** イベントを使用して初期化できます。**PerpetuumSoft.Reporting.Components.ResolveSubReportEventArgs** オブジェクトは、このハンドラにパラメータとして渡されます。このオブジェクトには **TemplateName** プロパティがあり、入れ子のレポートが必要とする物を定義し、**Template** プロパティ経由で渡します。



## 最終ドキュメントのビジュアル化と編集

最終ドキュメントを表示するには、Windows アプリケーションの場合はレポートビューアを、ASP.NET アプリケーションの場合は SharpShooterWebView を使用できます。

最終ドキュメントはレポートテンプレートに従って、レポートで作成および貼り付けられたビジュアルオブジェクトだけで構成されています。レポートデザイナーで最終ドキュメントを開いて、オブジェクトの静的なプロパティ（サイズ、レイアウト、スタイル）をすべて編集できます。また、ドキュメントに新しいビジュアルオブジェクトを追加することもできます。バンドを追加すると、レポートデザイナーで表示された状態で最終ドキュメントに表示されます。

## エクスポート、印刷、保存

最終ドキュメントを作成してしまえば、さまざまな形式でドキュメントを提供することができます。

SharpShooter Reports はレポートを PDF、GIF、HTML、XML for Excel、RTF などのさまざまな形式にエクスポートできます。エクスポートパラメータを設定するためのオプションがあります。例えば、イメージ品質の設定、ドキュメントの特定ページの選択などです。

最終ドキュメントをハードディスクに保存できます。ドキュメントは、.rsd 拡張子の付いた xml 形式で保存されます。.rsd ファイルはレポートビューアで使われます。

レポートビューアからドキュメントを印刷できます。印刷設定では、印刷するページ数やドキュメントのコピーの数を指定します。

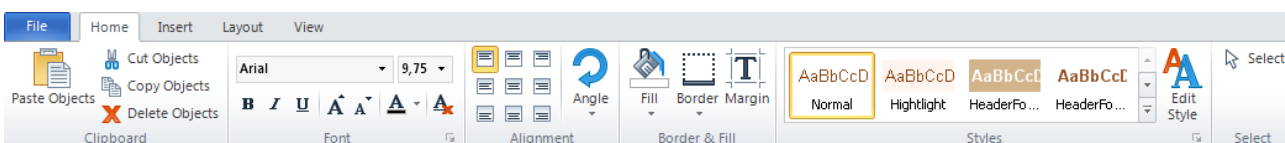
## レポート デザイナ インターフェイス

レポートデザイナはレポートテンプレートの作成や編集を行います。デザイナを使って、テンプレートにページやオブジェクトを追加したり、静的/動的なプロパティの設定やデータソースに接続したりできます。

### タブ

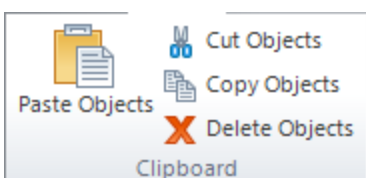
#### ホーム タブ

このタブには、一番よく使われるコマンドが含まれています。



#### クリップボード

クリップボードで使われるボタンが含まれています。



[オブジェクトの貼り付け] ボタン：クリップボードの内容を選択領域に貼り付けます。

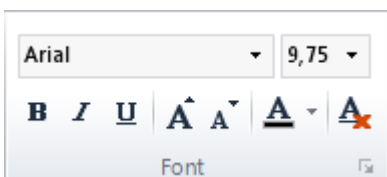
[オブジェクトの切り取り] ボタン：選択したオブジェクトを切り取って、クリップボードに貼り付けます。

[オブジェクトのコピー] ボタン：選択したオブジェクトをコピーし、クリップボードに貼り付けます。

[オブジェクトの削除] ボタン：選択したオブジェクトを削除します。

#### フォント

Font プロパティで使われる制御が含まれています。



[フォント名] 一覧：利用可能なスタイルが格納されています (Font.FamilyName プロパティを設定します)。

[フォントサイズ] 一覧：フォントサイズを格納し、任意のサイズを設定できます (Font.Size プロパティを設定します)。

[太字] ボタン：テキストに太字を設定します (Font.Bold プロパティを設定します)。

[斜体] ボタン：テキストに斜体を設定します (Font.Italic プロパティを設定します)。

[下線] ボタン：テキストに下線を設定します (Font.Underline プロパティを設定します)。

[フォントを大きくする] ボタン：フォントサイズを大きくします (Font.Size プロパティを設定します)。

[フォントを小さくする] ボタン：フォントサイズを小さくします (Font.Size プロパティを設定します)。



[塗りつぶし] ボタン：テキストの色と塗りつぶしを設定します（選択したオブジェクトの **TextFill** プロパティを設定します）。



[主な塗りつぶし] と [標準の塗りつぶし]：カラーパレットが格納されています。

[最近使った塗りつぶし]：最近使った塗りつぶしが格納されます。

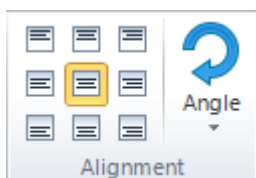
[塗りつぶしなし]：すべての塗りつぶし設定をキャンセルし、**TextFill** プロパティを [None] に設定します。

[その他の塗りつぶし...]：カスタムな塗りつぶしを作成するために、「塗りつぶしエディタ」を開きます。

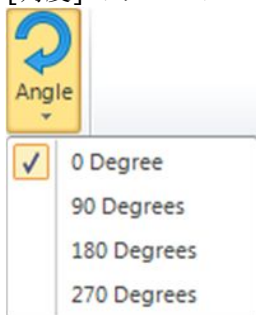
[スタイルのみ使用] ボタン：スタイル設定を行います。これらの設定は選択したスタイルと一致し、変更をすべて削除します。

### 位置合わせ

《位置合わせ》にある表を使って、**TextAlign** プロパティや **ImageAlign** プロパティを設定できます。



[角度] ボタン：プロパティの値リストを開いて **Angle** プロパティを設定します。



### 枠と塗りつぶし



[塗りつぶし] ボタン：背景の色と塗りつぶしを設定します（**Fill** プロパティを設定します）。

[主な塗りつぶし] と [標準の塗りつぶし]：カラーパレットを格納しています。

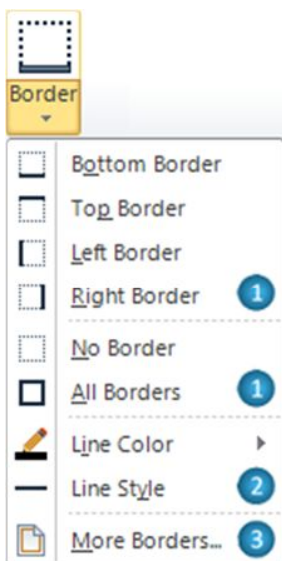
[最近使った塗りつぶし]：最近使った塗りつぶしが格納されます。



[塗りつぶしなし] : すべての塗りつぶし設定をキャンセルし、Fill プロパティを [None] に設定します。

[その他の塗りつぶし...] : カスタムな塗りつぶしを作成するために、「塗りつぶしエディタ」を開きます。

[枠] ボタン : 枠を設定します (Border プロパティを設定します)。



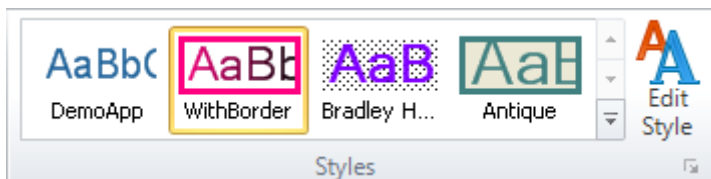
1 - 枠線の位置を設定します。

2 - 枠線のスタイルや色を設定します。[線の色] のサブメニューにはカラーパレットが格納されています。[線スタイル] のサブメニューには、線の種類が格納されています。

3 - [その他の枠...] : 「枠エディタ」を開きます。

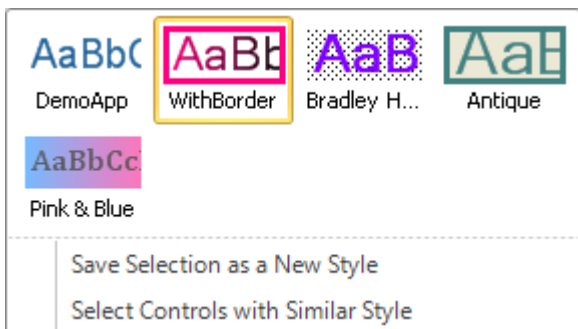
[余白] ボタン : Margins プロパティを設定する「余白エディタ」を開きます。

### スタイル



「スタイルギャラリー」に、テンプレートのスタイルがプレビューされます。スタイルのプレビューをクリックすると、選択したオブジェクトにそのスタイルを適用します (StyleName プロパティを設定します)。

[選択したスタイルを新規スタイルとして保存] : 新しいスタイル名を設定するために「書式から新しいスタイルを作成」ダイアログを開き、選択したオブジェクトのスタイルを保存します。

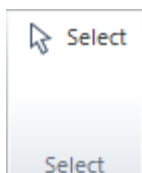


[スタイルの編集] ボタン : 「スタイルシートエディタ」を開きます。

[似たようなスタイルのコントロールを選択] : 現在ページのオブジェクトの StyleName プロパティが、選択したスタイルに設定されているものはすべて選択します。

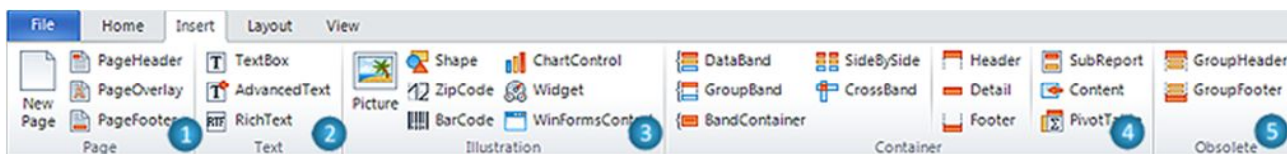
### 選択

[選択] ボタン : 選択モードを切り替えます。



## 挿入タブ

このタブを使って、テンプレートにオブジェクトを追加できます。ボタンはグループごとに分けられています。



**1** - 《ページ》には、新規ページ、PageHeader、Page Overlay、PageFooter バンドを追加するボタンが格納されています。

**2** - 《テキスト》には、テキスト出力オブジェクト : TextBox、AdvancedText、RichText の追加ボタンが格納されています。

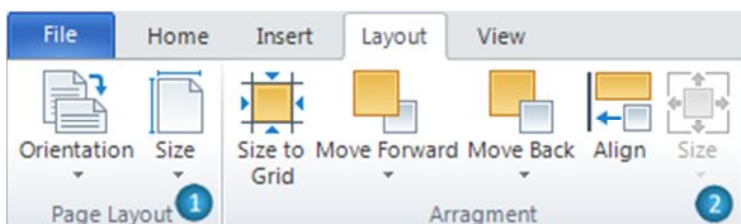
**3** - 《図表》には、さまざまなビジュアルオブジェクト : Picture、Shape、ZipCode、BarCode、ChartControl、Widget、WinFormsControl を追加するボタンが格納されています。

**4** - 《コンテナ》には、さまざまなバンド : DataBand、GroupBand、BandContainer、SideBySide、CrossBand、Header、Detail、Footer、SubReport、Content、PivotTable を追加するボタンが格納されています。

**5** - 《旧式》には、旧式で旧バージョンとの互換性に必要なバンド : GroupHeader、GroupFooter を追加するボタンが格納されています。これらのコンポーネントを使用して新規レポートを作成することはお勧めしておりません。GroupBand に Header や Footers を使用してください。

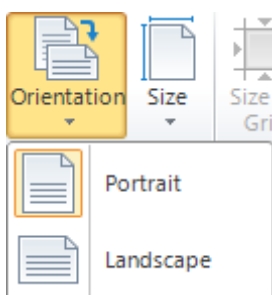
## レイアウト タブ

このタブには、ページレイアウト、オブジェクトの間隔や位置に作用する制御が含まれています。



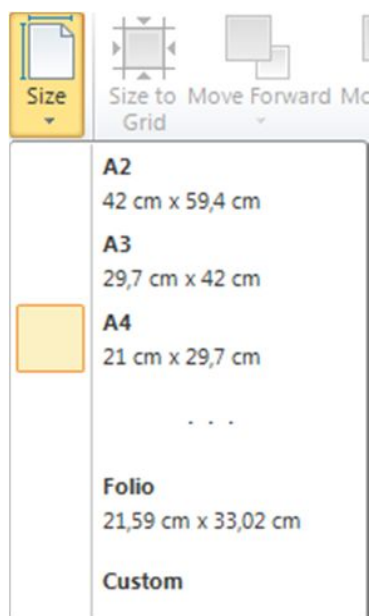
**1** - 《ページレイアウト》には、ページのプロパティを設定するボタンが格納されています。

[向き] ボタン : Page.Orientation プロパティの値を設定するために値リストを開きます。



[サイズ] ボタン : Page.PaperKind プロパティの値を設定するために値リストを開きます。[ユーザー定義] はユーザー定義の用紙サイズを設定するための「用紙サイズエディタ」を開きます。

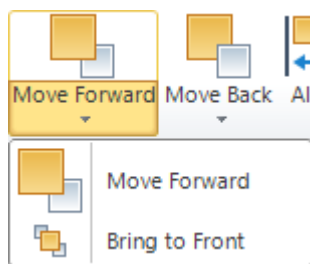




2 - 《順序》を使用して、オブジェクトのサイズや位置を操作できます。

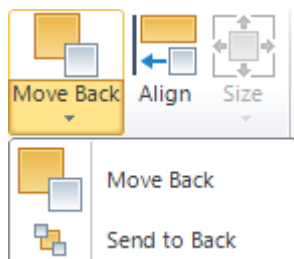
[グリッドのサイズに合わせる] ボタン：選択したオブジェクトのサイズをグリッドに合わせます。

[整列] ボタン：オブジェクトの整列を設定するための「整列」エディタを開きます。



[前面に移動]：現在のコンテナの他のオブジェクトの背面に隠れているオブジェクトを前面に移動します。

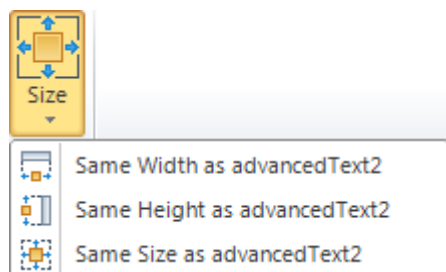
[先頭に移動]：現在のコンテナの他のオブジェクトの背面に隠れているオブジェクトを先頭に移動します。



[背面に移動]：現在のコンテナで選択したオブジェクトを他のオブジェクトの背面に移動します。

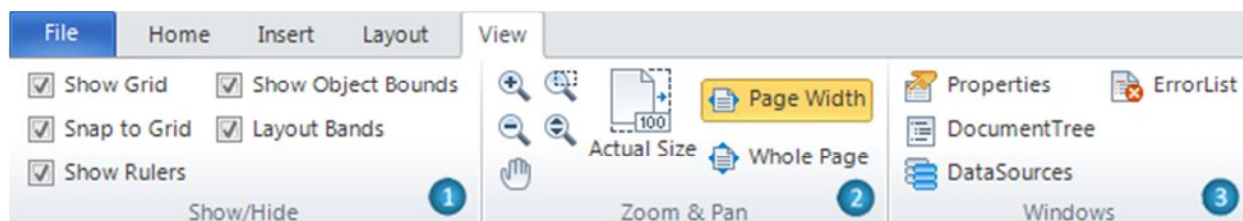
[末尾に移動]：現在のコンテナで選択したオブジェクトを末尾に移動します。

[サイズ] ボタン：選択したオブジェクトのサイズを変更します。選択した主要オブジェクトのように、パラメータを設定します。



## 表示タブ

このタブには、テンプレートの表示（表示モード、表示/非表示オプション、ズーム、ウィンドウ操作）に作用する制御が含まれています。



**1** - 《表示/非表示》では、デザイナの表示を設定できます。

[グリッドを表示]：グリッドの表示/非表示を定義します（`Document.ShowGrid` プロパティを設定します）。

[グリッドに合わせる]：グリッドに合わせるか定義します（`Document.SnapToGrid` プロパティを設定します）。

[ルーラの表示]：ルーラの表示/非表示を定義します。

[オブジェクトの境界を表示]：オブジェクトの境界を表示/非表示にします（`Document.ShowObjectBounds` プロパティを設定します）。

[バンドのレイアウト]：バンドの自動レイアウトオプションを切り替えます。

**2** - 《ズーム & パン》では、ズームやパンモードを設定できます。

**3** - 《ウィンドウ》には、「プロパティ」ウィンドウ、「ドキュメントツリー」ウィンドウ、「データソース」ウィンドウ、「エラー一覧」ウィンドウを開くボタンが格納されています。

## コンテキストタブ

コンテキストタブは、特定のオブジェクトを選択した時にだけ関係のあるコマンドのコレクションを表示するために使用します。

### TextBox タブセット

TextBox タブセットには「形式」タブが含まれています。

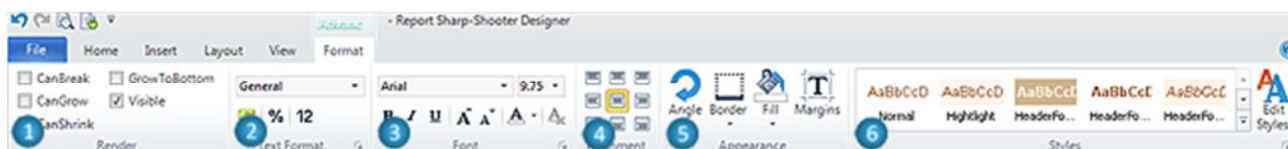


TextBox の「形式」タブには次のグループが含まれています。

- 1 - «生成»には、生成に関するプロパティ：CanGrow、CanShrink、GrowToBottom、Visible を設定するためのチェックボックスが格納されています。
- 2 - «テキスト形式»は TextFormat プロパティを設定します。このリストには、TextFormat プロパティの利用可能な値が格納されています。ボタンで瞬時に「通貨」、「パーセント」、「数値」を設定できます。
- 3 - «フォント»では、Font プロパティを設定できます。
- 4 - «位置合わせ»では、TextAlign プロパティを設定できます。
- 5 - «外観»には、見た目に関するプロパティ：Angle、Border、Fill、Margins の設定ボタンが含まれています。
- 6 - «スタイル»では、StyleName プロパティを設定できます。

### AdvancedText タブセット

AdvancedText タブセットには「形式」タブが含まれています。

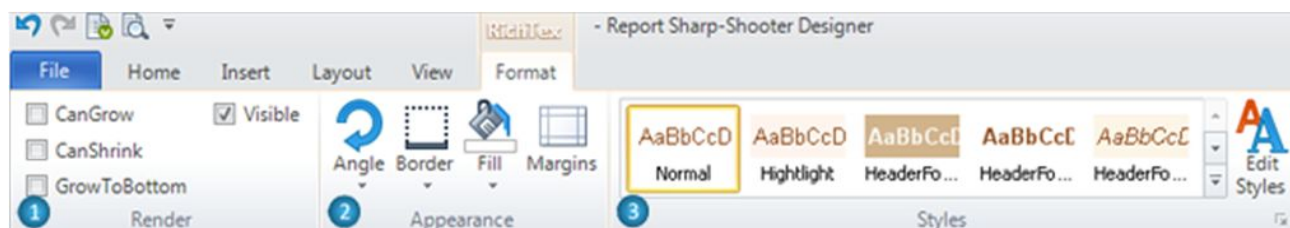


AdvancedText の「形式」タブには、次のグループが含まれています。

- 1 - «生成»には、生成に関するプロパティ：CanBreak、CanGrow、CanShrink、GrowToBottom、Visible を設定するためのチェックボックスが格納されています。
- 2 - «テキスト形式»は TextFormat プロパティを設定します。このリストには、TextFormat プロパティの利用可能な値が格納されています。ボタンで瞬時に「通貨」、「パーセント」、「数値」を設定できます。
- 3 - «フォント»では、Font プロパティを設定できます。
- 4 - «位置合わせ»では、TextAlign プロパティを設定できます。
- 5 - «外観»には、見た目に関するプロパティ：Angle、Border、Fill、Margins を設定するためのボタンが含まれています。
- 6 - «スタイル»では、StyleName プロパティを設定できます。

## RichText タブセット

RichText タブセットには「形式」タブが含まれています。

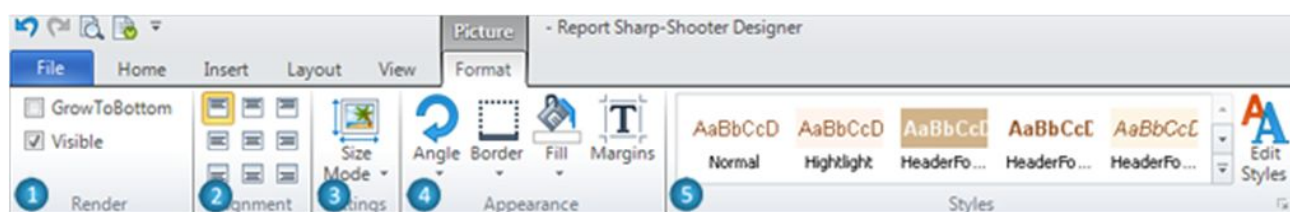


RichText の「形式」タブには、次のグループが含まれています。

- 1 - 《生成》には、生成に関するプロパティ：CanGrow、CanShrink、GrowToBottom、Visible を設定するためのチェックボックスが格納されています。
- 2 - 《外観》には、見た目に関するプロパティ：Angle、Border、Fill、Margins を設定するためのボタンが含まれています。
- 3 - 《スタイル》では、StyleName プロパティを設定できます。

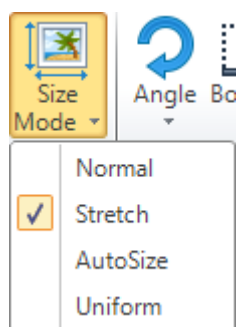
## Picture タブセット

Picture タブセットには「形式」タブが含まれています。



Picture の「形式」タブには、次のグループが含まれています。

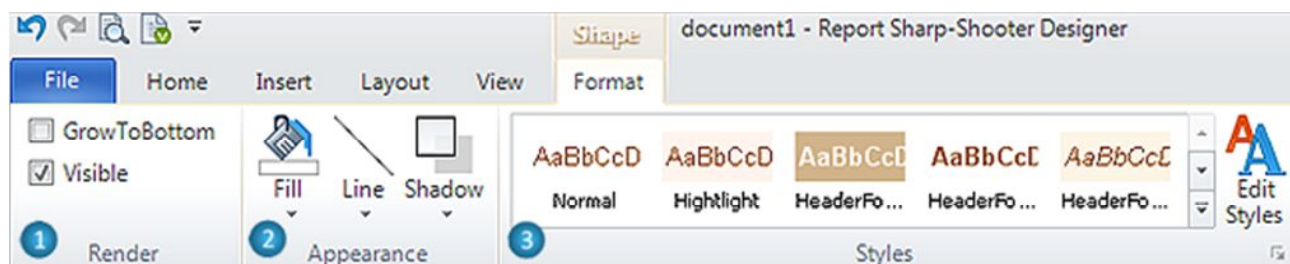
- 1 - 《生成》には、生成に関するプロパティ：GrowToBottom、Visible を設定するためのチェックボックスが格納されています。
- 2 - 《位置合わせ》では、ImageAlign プロパティを設定できます。
- 3 - 《設定》には、SizeMode プロパティの値リストを開いて設定するための [サイズモード] ボタンが格納されています。



- 4 - 《外観》には、見た目に関するプロパティ：Angle、Border、Fill、Margins を設定するボタンが含まれています。
- 5 - 《スタイル》では、StyleName プロパティを設定できます。

## Shape タブセット

Shape タブセットには「形式」タブが含まれています。

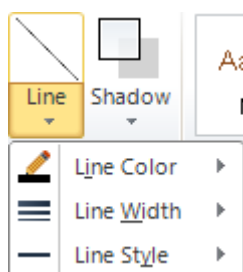


Shape の「形式」タブには、次のグループが含まれています。

**1** - «生成»には、生成に関するプロパティ：**GrowToBottom**、**Visible** を設定するためのチェックボックスが格納されています。

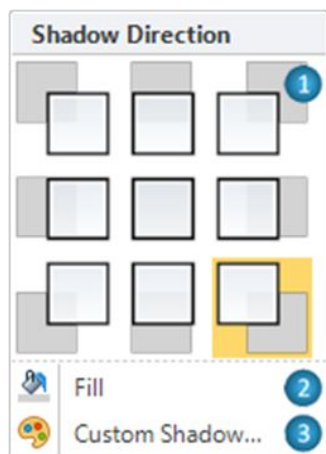
**2** - «外観»には、見た目に関するプロパティ：**Fill**、**Line**、**Shadow** を設定するボタンが含まれています。

[線] ボタン：**Shape** の線（線の色、線幅、線スタイル）を設定できます。



[線の色] のサブメニューにはカラーパレットが格納されています。[線スタイル] のサブメニューには線の種類が格納されています。

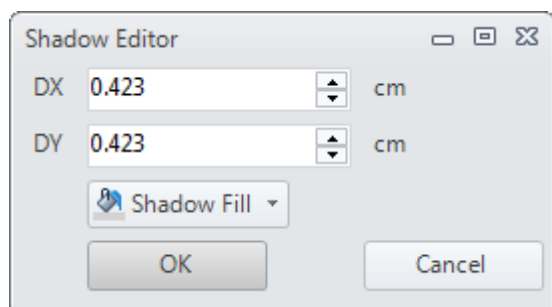
[影] ボタン：**Shape** の影を設定します。



**2.1** - 影の位置を設定する領域

**2.2** - [塗りつぶし] のサブメニュー：影の塗りつぶしを設定するカラーパレットを開きます。

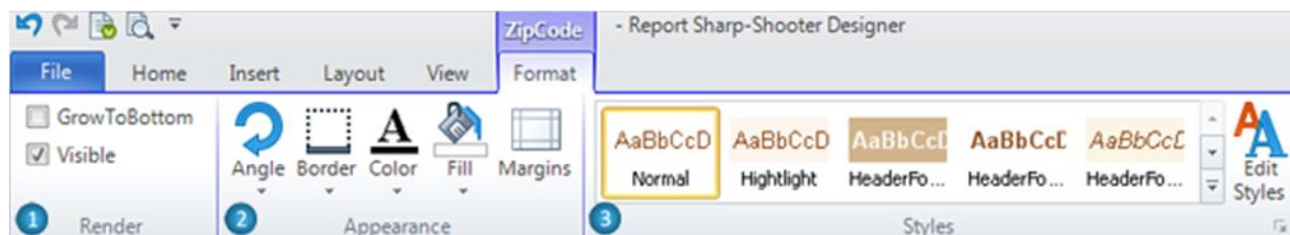
**2.3** - [カスタムな影]：影のプロパティを設定する「影エディタ」を開きます。



3 - 《スタイル》では、StyleName プロパティを設定できます。

### ZipCode タブセット

ZipCode タブセットには「形式」タブが含まれています。



ZipCode の「形式」タブには、次のグループが含まれています。

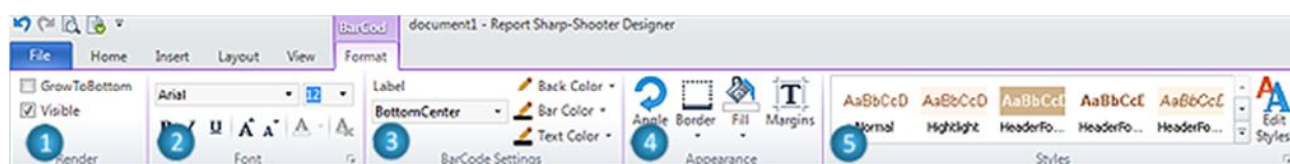
1 - 《生成》には、生成に関するプロパティ：GrowToBottom、Visible を設定するためのチェックボックスが格納されています。

2 - 《外観》には、見た目に関するプロパティ：Angle、Border、Fill、Margins と Color（ZipCode.Color プロパティを設定します）を設定するボタンが含まれています。

3 - 《スタイル》では、StyleName プロパティを設定できます。

### BarCode タブセット

BarCode タブセットには「形式」タブが含まれています。



BarCode の「形式」タブには、次のグループが含まれています。

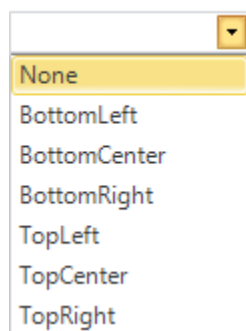
1 - 《生成》には、生成に関するプロパティ：GrowToBottom、Visible を設定するためのチェックボックスが格納されています。

2 - 《フォント》では、Font のプロパティを設定できます。

3 - 《バーコード設定》は、バーコード特有のプロパティを設定します。

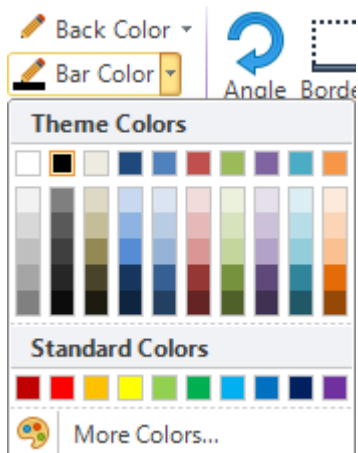
[ラベルの位置]：リストには LabelPosition プロパティの値が格納されています。[なし]を選択すると ShowLabel プロパティが False に設定されます。

Label



[背景色]、[Bar 色]、[テキストの色] ボタンは、それぞれ BackColor、BarColor、TextColor プロパティを設定します。



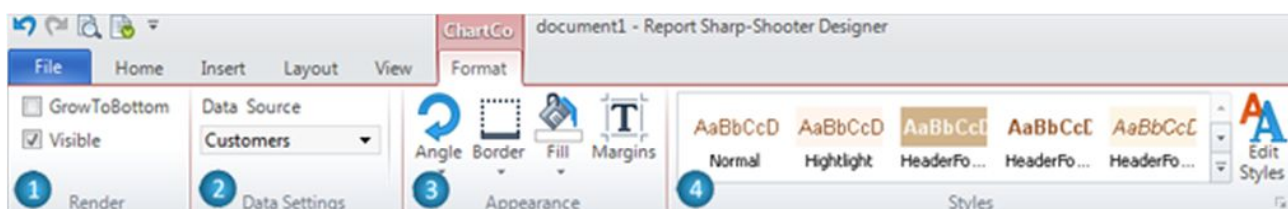


4 - 《外観》には、見た目に関するプロパティ : **Angle**、**Border**、**Fill**、**Margins** を設定するボタンが含まれています。

5 - 《スタイル》では、**StyleName** プロパティを設定できます。

### ChartControl タブセット

ChartControl タブセットには「形式」タブが含まれています。



ChartControl の「形式」タブには、次のグループが含まれています。

1 - 《生成》には、生成に関するプロパティ : **GrowToBottom**、**Visible** を設定するためのチェックボックスが格納されています。

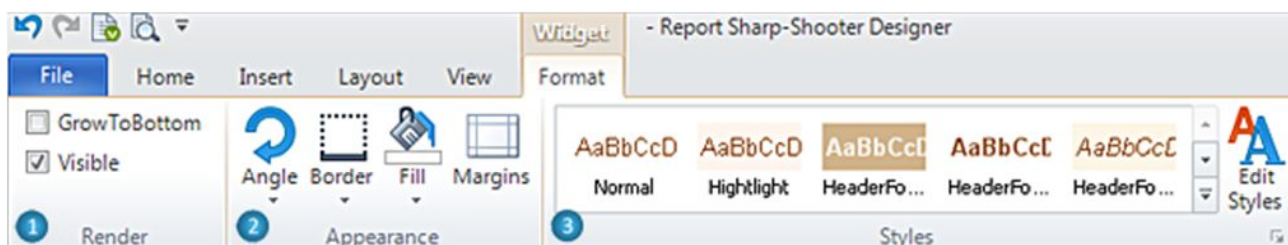
2 - 《データ設定》には、**DataSource** プロパティを設定するための制御が含まれています。

3 - 《外観》のグループには、見た目に関するプロパティ : **Angle**、**Border**、**Fill**、**Margins** を設定するボタンが含まれています。

5 - 《スタイル》では、**StyleName** プロパティを設定できます。

### Widget タブセット

Widget タブセットには「形式」タブが含まれています。



Widget の「形式」タブには、次のグループが含まれています。

1 - 《生成》には、生成に関するプロパティ : **GrowToBottom**、**Visible** を設定するためのチェックボックスが格納されています。

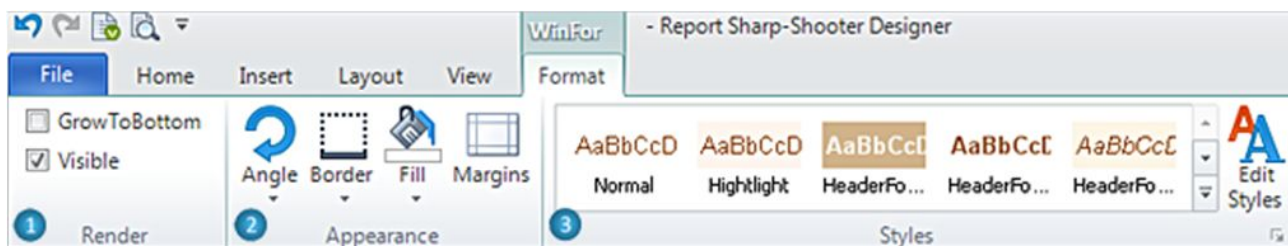


2 - 《外観》には、見た目に関するプロパティ : Angle、Border、Fill、Margins を設定するボタンが含まれています。

3 - 《スタイル》では、StyleName プロパティを設定できます。

### WinFormsControl タブセット

WinFormsControl タブセットには「形式」タブが含まれています。



WinFormsControl の「形式」タブには、次のグループが含まれています。

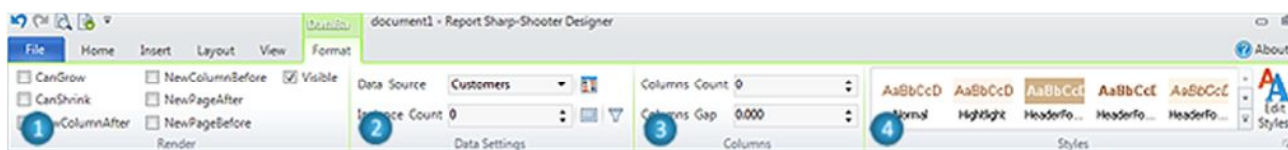
1 - 《生成》には、生成に関するプロパティ : GrowToBottom、Visible を設定するためのチェックボックスが格納されています。

2 - 《外観》には、見た目に関するプロパティ : Angle、Border、Fill、Margins を設定するボタンが含まれています。

3 - 《スタイル》では、StyleName プロパティを設定できます。

### DataBand タブセット

DataBand タブセットには「形式」タブが含まれています。



DataBand の「形式」タブには、次のグループが含まれています。

1 - 《生成》には、生成に関するプロパティ : CanGrow、CanShrink、NewColumnAfter、NewColumnBefore、NewPageAfter、NewPageBefore、Visible を設定するためのチェックボックスが格納されています。

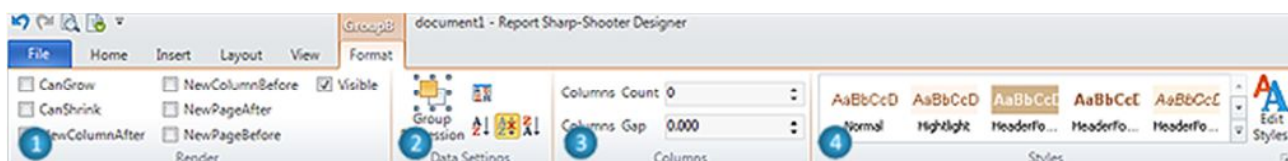
2 - 《データ設定》には、DataSource、InstanceCount、Sort、Totals、FilterExpression プロパティを設定するための制御が含まれています。

3 - 《列》には、ColumnsCount と ColumnsGap プロパティを設定するための制御が含まれています。

4 - 《スタイル》では、StyleName プロパティを設定できます。

### GroupBand タブセット

GroupBand タブセットには「形式」タブが含まれています。

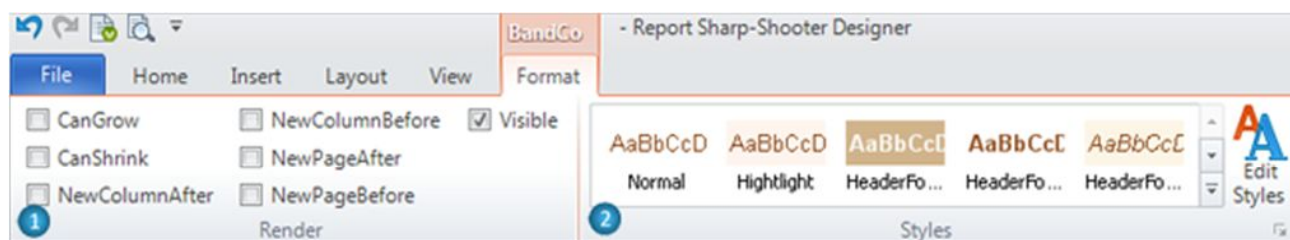


GroupBand の「形式」タブには、次のグループが含まれています。

- 1 - 《生成》には、生成に関するプロパティ：CanGrow、CanShrink、NewColumnAfter、NewColumnBefore、NewPageAfter、NewPageBefore、Visible を設定するためのチェックボックスが格納されています。
- 2 - 《データ設定》には、GroupExpression、SortExpression、Order プロパティを設定するための制御が含まれています。
- 3 - 《列》には、ColumnsCount と ColumnsGap プロパティを設定するための制御が含まれています。
- 4 - 《スタイル》では、StyleName プロパティを設定できます。

### BandContainer タブセット

BandContainer タブセットには「形式」タブが含まれています。

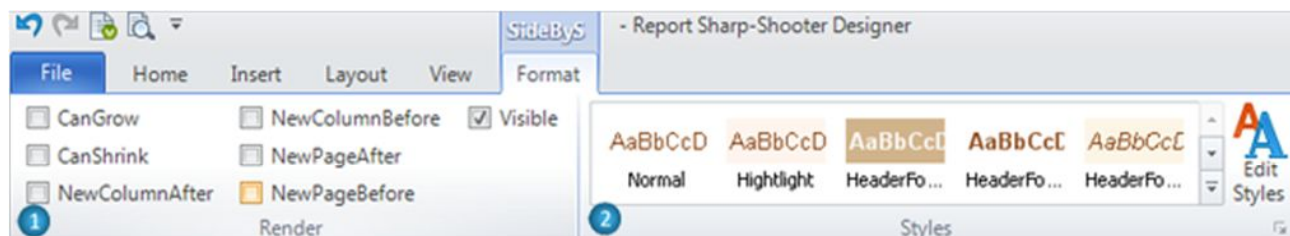


BandContainer の「形式」タブには、次のグループが含まれています。

- 1 - 《生成》には、生成に関するプロパティ：CanGrow、CanShrink、NewColumnAfter、NewColumnBefore、NewPageAfter、NewPageBefore、Visible を設定するためのチェックボックスが格納されています。
- 2 - 《スタイル》では、StyleName プロパティを設定できます。

### SideBySide タブセット

SideBySide タブセットには「形式」タブが含まれています。

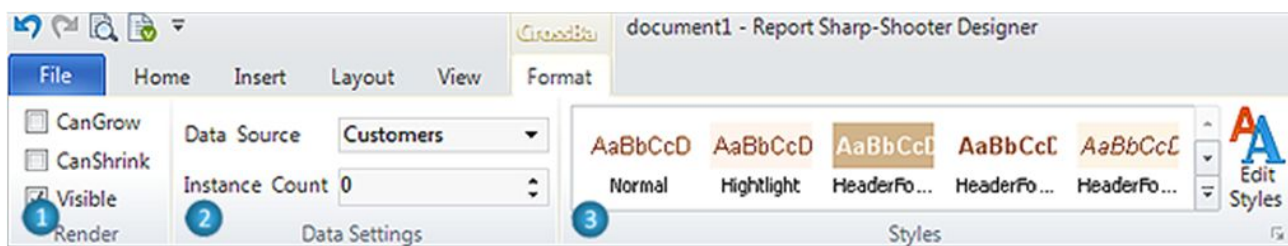


SideBySide の「形式」タブには、次のグループが含まれています。

- 1 - 《生成》には、生成に関するプロパティ：CanGrow、CanShrink、NewColumnAfter、NewColumnBefore、NewPageAfter、NewPageBefore、Visible を設定するためのチェックボックスが格納されています。
- 2 - 《スタイル》では、StyleName プロパティを設定できます。

### CrossBand タブセット

CrossBand タブセットには「形式」タブが含まれています。

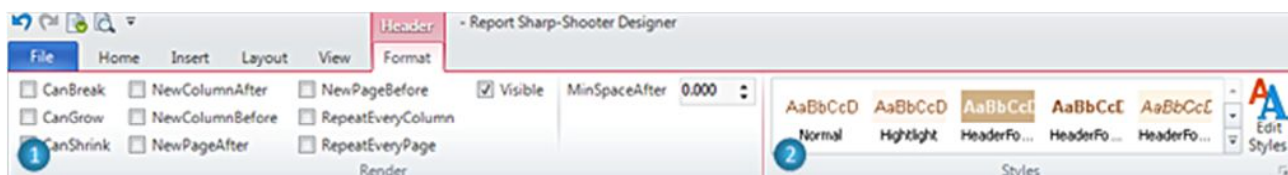


CrossBand の「形式」タブには、次のグループが含まれています。

- 1 - «生成»には、生成に関するプロパティ：CanGrow、CanShrink、Visible を設定するためのチェックボックスが格納されています。
- 2 - «データ設定»には、DataSource と InstanceCount プロパティを設定するための制御が含まれています。
- 3 - «スタイル»では、StyleName プロパティを設定できます。

### Header タブセット

Header タブセットには「形式」タブが含まれています。

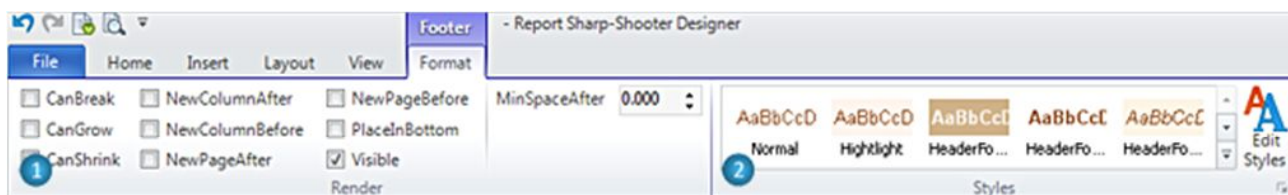


Header の「形式」タブには、次のグループが含まれています。

- 1 - «生成»には、生成に関するプロパティ：CanBreak、CanGrow、CanShrink、NewColumnAfter、NewColumnBefore、NewPageAfter、NewPageBefore、RepeatEveryColumn、RepeatEveryPage、Visible を設定するためのチェックボックスと、MinSpaceAfter プロパティを設定するためのカウンタ：MinSpaceAfter が格納されています。
- 2 - «スタイル»では、StyleName プロパティを設定できます。

### Footer タブセット

Footer タブセットには「形式」タブが含まれています。

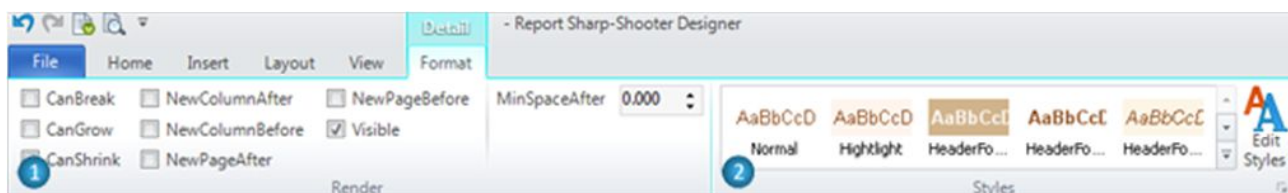


Footer の「形式」タブには、次のグループが含まれています。

- 1 - «生成»には、生成に関するプロパティ：CanBreak、CanGrow、CanShrink、NewColumnAfter、NewColumnBefore、NewPageAfter、NewPageBefore、PlaceInBottom、Visible を設定するためのチェックボックスと、MinSpaceAfter プロパティを設定するためのカウンタ：MinSpaceAfter が格納されています。
- 2 - «スタイル»では、StyleName プロパティを設定できます。

### Detail タブセット

Detail タブセットには「形式」タブが含まれています。



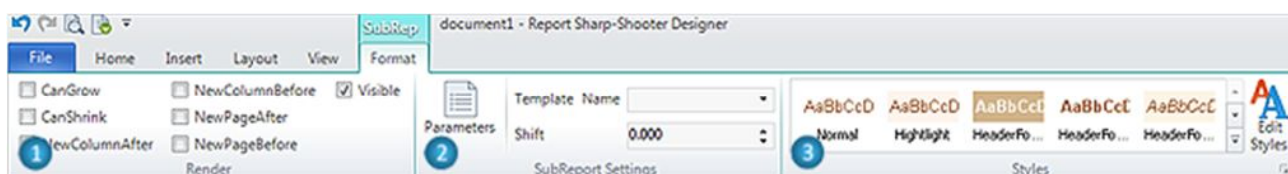
Detail の「形式」タブには、次のグループが含まれています。

1 - 《生成》には、生成に関するプロパティ：CanBreak、CanGrow、CanShrink、NewColumnAfter、NewColumnBefore、NewPageAfter、NewPageBefore、Visible を設定するためのチェックボックスと、MinSpaceAfter プロパティを設定するためのカウンタ：MinSpaceAfter が格納されています。

2 - 《スタイル》では、StyleName プロパティを設定できます。

### SubReport タブセット

SubReport タブセットには「形式」タブが含まれています。



SubReport の「形式」タブには、次のグループが含まれています。

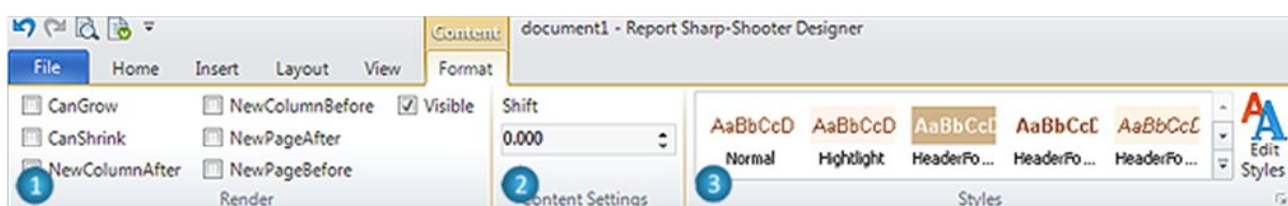
1 - 《生成》には、生成に関するプロパティ：CanGrow、CanShrink、NewColumnAfter、NewColumnBefore、NewPageAfter、NewPageBefore、Visible を設定するためのチェックボックスが格納されています。

2 - 《サブレポート設定》には、Parameters、TemplateName、Shift プロパティを設定するための制御が含まれています。

3 - 《スタイル》では、StyleName プロパティを設定できます。

### Content タブセット

Content タブセットには「形式」タブが含まれています。



Content の「形式」タブには、次のグループが含まれています。

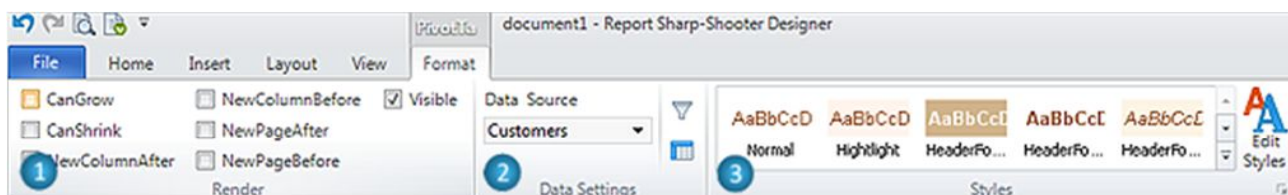
1 - 《生成》には、生成に関するプロパティ：CanGrow、CanShrink、NewColumnAfter、NewColumnBefore、NewPageAfter、NewPageBefore、Visible を設定するためのチェックボックスが格納されています。

2 - 《コンテンツ設定》には Shift プロパティの制御が含まれています。

3 - 《スタイル》では、StyleName プロパティを設定できます。

### PivotTable タブセット

PivotTable タブセットには「形式」タブが含まれています。



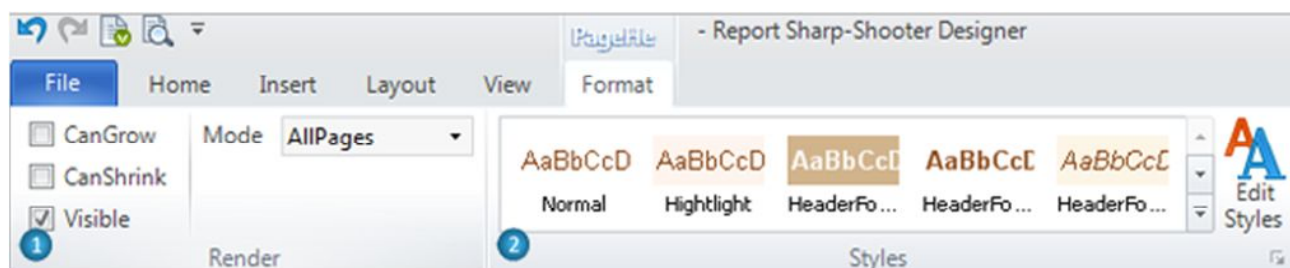
PivotTable の「形式」タブには、次のグループが含まれています。

- 1 - 《生成》には、生成に関するプロパティ : CanGrow、CanShrink、NewColumnAfter、NewColumnBefore、NewPageAfter、NewPageBefore、Visible を設定するためのチェックボックスが格納されています。
- 2 - 《データ設定》には、DataSource、FilterExpression、Layout プロパティを設定するための制御が含まれています。
- 3 - 《スタイルは》では、StyleName プロパティを設定できます。



## PageHeader タブセット

PageHeader タブセットには「形式」タブが含まれています。

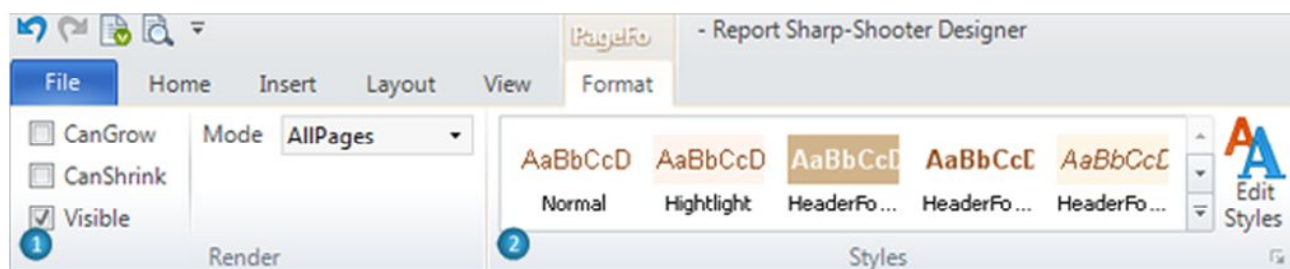


PageHeader の「形式」タブには、次のグループが含まれています。

- 1 - «生成»には、生成に関するプロパティ：CanGrow、CanShrink、Visible を設定するためのチェックボックスと、Mode プロパティの値を含んでいる Mode 一覧が格納されています。
- 2 - «スタイル»では、StyleName プロパティを設定できます。

## PageFooter タブセット

PageFooter タブセットには「形式」タブが含まれています。

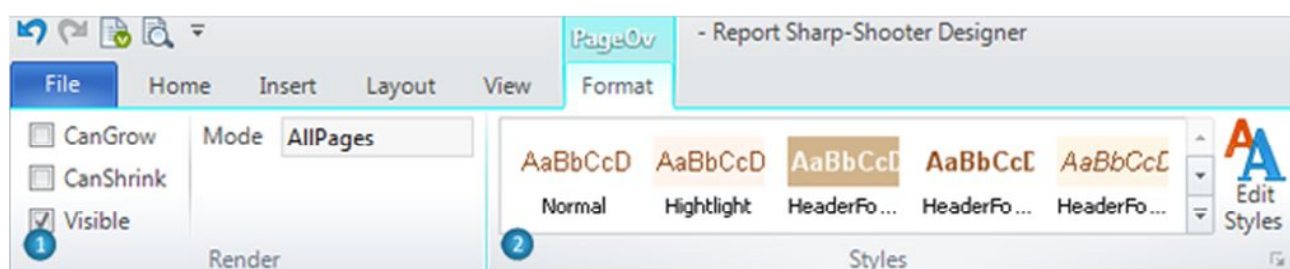


PageFooter の「形式」タブには、次のグループが含まれています。

- 1 - «生成»には、生成に関するプロパティ：CanGrow、CanShrink、Visible を設定するためのチェックボックスと、Mode プロパティの値を含んでいる Mode 一覧が格納されています。
- 2 - «スタイル»では、StyleName プロパティを設定できます。

## PageOverlay タブセット

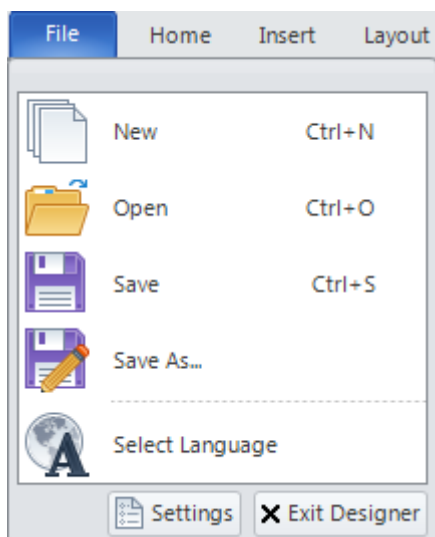
PageOverlay タブセットには「形式」タブが含まれています。



PageOverlay の「形式」タブには、次のグループが含まれています。

- 1 - «生成»には、生成に関するプロパティ：CanGrow、CanShrink、Visible を設定するためのチェックボックスと、Mode プロパティの値を含んでいる Mode 一覧が格納されています。
- 2 - «スタイル»では、StyleName プロパティを設定できます。

## アプリケーションメニュー



[新規] : 新しいテンプレートを作成するウィザードギャラリーを開きます。

[開く] : テンプレート/ドキュメントを開く「テンプレートを開く」ウィンドウを開きます。

[保存] : 編集したテンプレート/ドキュメントの変更を保存します。テンプレート/ドキュメントが保存されない場合は、保存先のフォルダやファイル名を選択する「テンプレートの保存」ウィンドウを開きます。

[名前を付けて保存...] : レポート/ドキュメントを保存するフォルダやファイル名を選択する「テンプレートの保存」ウィンドウを開きます。

[言語設定] : ローカライゼーション言語を選択する「言語設定」ウィンドウを開きます。

[設定] ボタン : デザイナのテーマや色スキーマを選択する「パレット設定」ウィンドウを開きます。

[デザイナを閉じる] ボタン : レポートデザイナを閉じます。

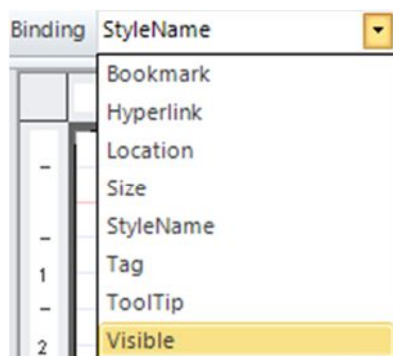
## バインドバー

バインドバーを使用すれば、瞬時にオブジェクトのプロパティをバインドできます。



**1** - [名称] - 選択したオブジェクトの名前を設定します。

**2** - [バインド] - 選択したオブジェクトの動的なプロパティ一覧で、値を設定するプロパティを選択できます。

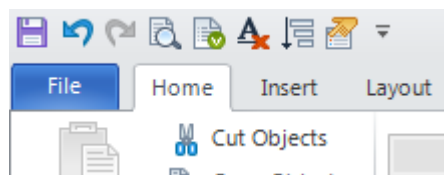


**3** - [スクリプトエディタ] ボタン- 選択したオブジェクトの式を設定するスクリプトエディタを開きます。

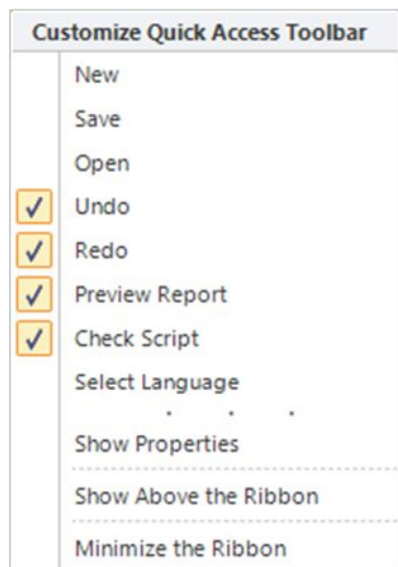
**4** - [F(x)] - 選択したオブジェクトに動的なプロパティを設定するための式を設定します。

## クイック アクセス ツールバー

クイック アクセス ツールバーは、よく使うコマンドにアクセスします。



カスタマイズすることで、必要なコマンドを追加できます。





クイック アクセス ツールバーに次のコマンドを追加できます。

ボタン	動作
新規ドキュメント	新しいテンプレートを作成します
保存	テンプレートの変更を保存します
開く	テンプレート/ドキュメントをファイルから読み込みます
言語設定	「言語設定」ダイアログを開きます
Font	「フォント」ダイアログを開きます
スタイルのみ使用	スタイル設定を設定します
Fill	Fill プロパティを設定する「塗りつぶしエディタ」を開きます
Text Fill	TextFill プロパティを設定する「塗りつぶしエディタ」を開きます
Border	「枠エディタ」を開きます
Margins	「余白エディタ」を開きます
Text Format	「フォーマットエディタ」を開きます
スタイルシートエディタ	「スタイルシートエディタ」を開きます
整列	「整列」ダイアログを開きます
実サイズ	100% ズームを設定します
ページの幅	ページの幅に合わせます
ページ全体	ページ全体に合わせます
プロパティ	「プロパティ」ウィンドウを開きます
ドキュメントツリー	「ドキュメントツリー」ウィンドウを開きます
データソース	「データソース」ウィンドウを開きます
エラー一覧	「エラー一覧」ウィンドウを開きます
グリッドを表示	グリッドを表示/非表示にします
グリッドに合わせる	「グリッドに合わせる」オプションをオン/オフにします
ルーラの表示	ルーラを表示/非表示にします
オブジェクトの境界を表示	オブジェクトの境界を表示/非表示にします
バンドのレイアウト	「バンドの自動レイアウト」オプションをオン/オフにします
テンプレートのプロパティ	テンプレートのプロパティを開きます

## コンテキストメニュー

コンテキストメニューには、一番よく使われるメニューが含まれています。メニュー一覧は、現在選択されているレポートオブジェクトによって決まります。

### 一般的な動作

	Cut Objects	Ctrl+X	1
	Copy Objects	Ctrl+C	2
	Paste Objects	Ctrl+V	3
	Delete Objects	Ctrl+D	4

**1** [オブジェクトの切り取り] - 選択したオブジェクトを切り取って、クリップボードに貼り付けます。

**2** [オブジェクトのコピー] - 選択したオブジェクトをコピーして、クリップボードに貼り付けます。

**3** [オブジェクトの貼り付け] - 選択した領域にクリップボードの内容を貼り付けます。

**4** [オブジェクトの削除] - 選択したオブジェクトを削除しま

す。

## 編集や順序



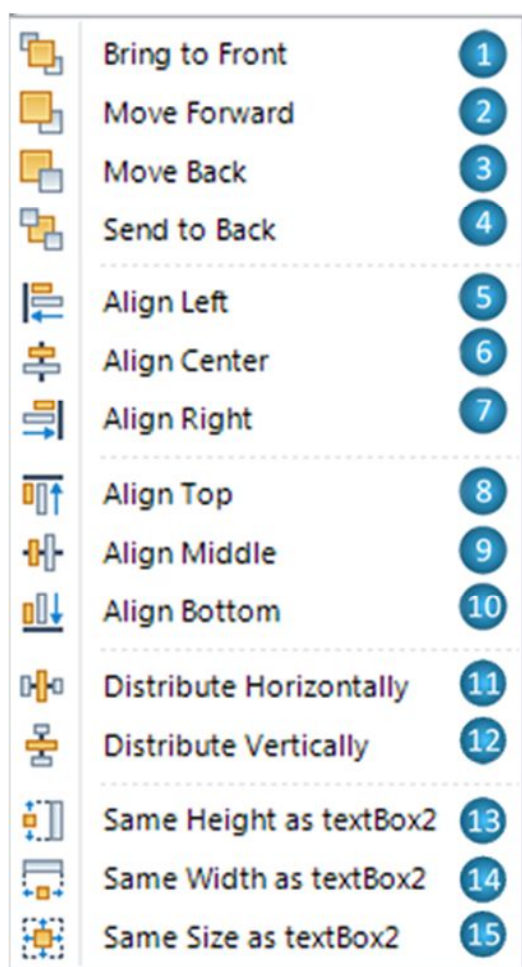
**1 [編集]** - このメニューはエディタを開きます。

- `TextBox.Text` プロパティを設定する、`TextBox` オブジェクトの「テキストエディタ」
- `AdvancedText element to set AdvancedText.Text` プロパティを設定する、`AdvancedText` オブジェクトの「書式付きテキストエディタ」
- `RichText.RTFtext` プロパティを設定する、`RichText` オブジェクトの「RichText エディタ」
- イメージを選択し、`Picture.Image` プロパティを設定する「ファイルを開く」ダイアログを開きます。
- `BarCode.Code` プロパティを設定する、`BarCode` オブジェクトの「テキストエディタ」
- `ChartControl.Chart` プロパティを設定する、`ChartControl` オブジェクトの `SharpShooter Charts` デザイナ
- `Widget.Instrument` プロパティを設定する、`Widget` オブジェクトの `Instrument` デザイナ
- `PivotTable.Layout` プロパティを設定する、`PivotTable` の「ソースフィールド」

**2 [クリーン]** - このメニューは内容をきれいにします。

- `TextBox` オブジェクトの `TextBox.Text` プロパティ
- `AdvancedText` オブジェクトの `AdvancedText.Text` プロパティ
- `RichText` オブジェクトの `RichText.RTFtext` プロパティ
- `Picture` オブジェクトの `Picture.Image` プロパティ
- `BarCode` オブジェクトの `BarCode.Code` プロパティ
- `ChartControl` の `ChartControl.Chart` プロパティ
- `Widget` オブジェクトの `Widget.Instrument` プロパティ

**3 [順序]** - このサブメニューには、オブジェクトの順序、位置、サイズを操作するためのメニューが含まれています。



**3.1 [先頭に移動]** - 現在のコンテナの他のオブジェクトで隠されないように、選択したオブジェクトを先頭に移動します。

**3.2 [前面に移動]** - 現在のコンテナの一部のオブジェクトで隠れるように、選択したオブジェクトを前面に移動します。

**3.3 [背面に移動]** - 現在のコンテナで他のいくつかのオブジェクトで隠れるように、選択したオブジェクトを背面に移動します。

**3.4 [末尾に移動]** - 選択したオブジェクトを現在のコンテナのオブジェクトの最後尾に移動します。

次のメニューは、複数のオブジェクトを選択した場合のみ利用できます。

**3.5 [左揃え]** - 選択したオブジェクトの左側を最左端のオブジェクトの左枠に揃えます。

**3.6 [中央揃え]** - 選択したオブジェクトの中心を最左端と最右端のオブジェクトの中心に揃えます。

**3.7 [右揃え]** - 選択したオブジェクトの右側を最右端のオブジェクトの右枠に揃えます。

**3.8 [上揃え]** - 選択したオブジェクトの上部を最上部のオブジェクトの上枠に合わせます。

**3.9 [中間揃え]** - 選択したオブジェクトの中央を最上部と最下部のオブジェクトの中間に揃えます。

**3.10 [下揃え]** - 選択したオブジェクトの下部を最下部のオブジェクトの下枠に揃えます。

**3.11 [左右に整列]** - 選択したオブジェクトを左右に等間隔で整列します。

**3.12 [上下に整列]** - 選択したオブジェクトを上下に等間隔で整列します。

**3.13 [<選択したメインのオブジェクト\*>と同じ高さ]** - 選択したすべてのオブジェクトの高さを選択したメインのオブジェクトの高さと同じにします。

**3.14 [<選択したメインのオブジェクト>と等幅]** - 選択したすべてのオブジェクトの幅を選択したメインのオブジェクトと等幅にします。

**3.15 [<選択したメインのオブジェクト>と同サイズ]** - 選択したすべてのオブジェクトを選択したメインのオブジェクトと同サイズにします。

\* 選択したメインのオブジェクト - 現在選択したオブジェクトで最後に選択されたオブジェクト。濃い青色で示されます。

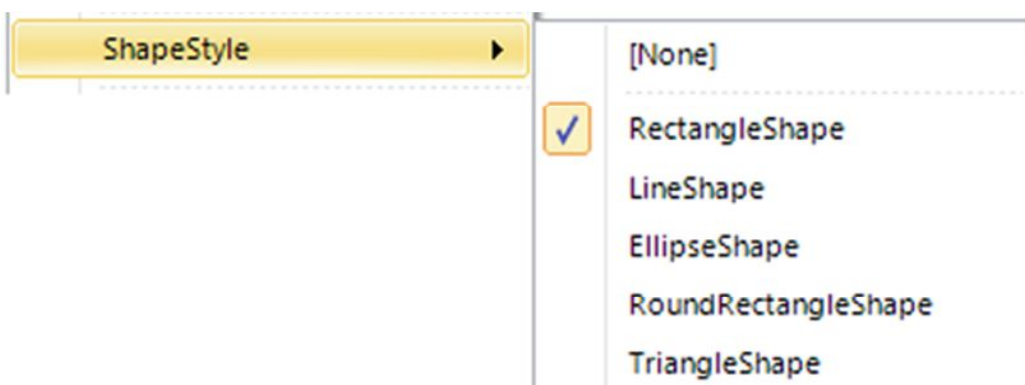
### オブジェクトに依存するメニュー

オブジェクト固有のプロパティを設定するメニューは以下の通りです。 .

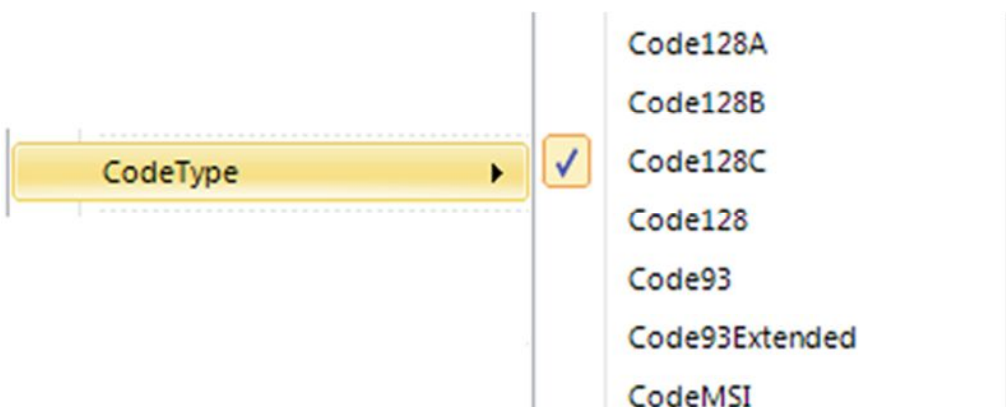
**SizeMode - Picture.SizeMode** プロパティを設定します。このサブメニューには、**SizeMode** プロパティの値リストが表示されます。



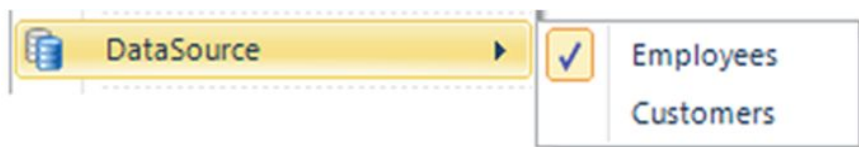
ShapeStyle – Shape.ShapeStyle プロパティを設定します。このサブメニューには、ShapeStyle プロパティの値リストが表示されます。



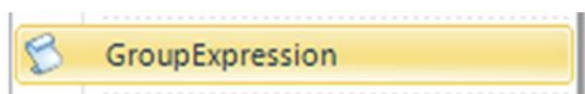
CodeType – BarCode.CodeType プロパティを設定します。このサブメニューには、CodeType プロパティの値リストが表示されます。



DataSource – DataSource property of DataBand、PivotTable、CrossBand の DataSource プロパティを設定します。このサブメニューには、テンプレートのデータソースの一覧が表示されます。



GroupExpression – GroupBand.GroupExpression プロパティを設定する「スクリプトエディタ」を開きます。



SubReport – SubReport.TemplateName プロパティを設定します。このサブメニューには、現在のレポートマネージャのレポート名の一覧が表示されます。

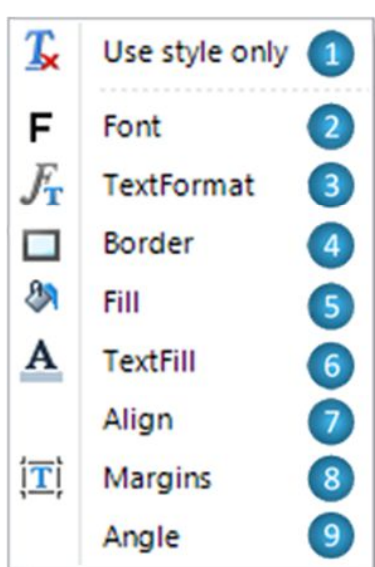


## オブジェクトのスタイル編集



**1 StyleName** – オブジェクトの **StyleName** プロパティを設定します。このサブメニューには、テンプレートで定義したスタイル一覧が表示されます。

**2 Style** – このサブメニューには、次のメニューが表示されます。



**2.1 [スタイルのみ使用]** – オブジェクトの書式設定（例：フォント、枠など）をすべてクリアします。オブジェクトの見た目は現在適用されるスタイルで示されます。

**2.2 Font** – Font プロパティを設定する「フォント」ダイアログを開きます。

**2.3 TextFormat** – TextFormat プロパティを設定する「フォーマット」ダイアログを開きます。

**2.4 Border** – Border プロパティを設定する「枠エディタ」を開きます。

**2.5 Fill** - Fill プロパティを設定する「塗りつぶしエディタ」を開きます。

**2.6 TextFill** - TextFill プロパティを設定する「塗りつぶしエディタ」を開きます。

**2.7 Align** – TextAlign または ImageAlign プロパティを設定します。このサブメニューには、位置合わせのオプションが表示されます。

**2.8 Margins** - Margins プロパティを設定する「余白エディタ」を開きます。

**2.9 Angle** – Angle プロパティを設定します。Angle プロパティの値リストが表示されます。

## 生成のサブメニュー

このサブメニューは、オブジェクトの生成に関するプロパティを設定する機能を提供しています。



**1 CanGrow** – CanGrow プロパティを True / False に設定します。

**2 CanShrink** – CanShrink プロパティを True / False に設定します。

**3 CanBreak** – CanBreak プロパティを True / False に設定します。

**4 GrowToBottom** – GrowToBottom プロパティを True / False に設定します。

**5 Visible** – Visible プロパティを True / False に設定します。

**6 Mode** – Mode プロパティを設定します。このサブメニューには、Mode プロパティの値リストが表示されます。



## その他のメニュー

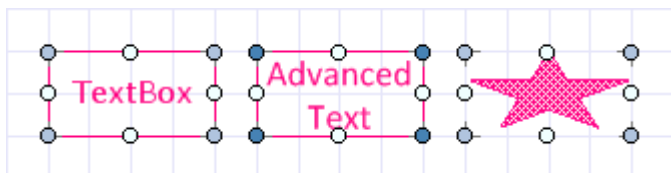


[プロパティを表示] - プロパティグリッドに選択したオブジェクトのプロパティを表示します。

## グループ選択

複数のオブジェクトを選択すると、コンテキストメニューには選択されたすべてのオブジェクトに共通するメニューが表示されます。コマンドは選択した各オブジェクトに適用されます。

下図では、選択されたメインオブジェクトは **AdvancedText** オブジェクトになっています。

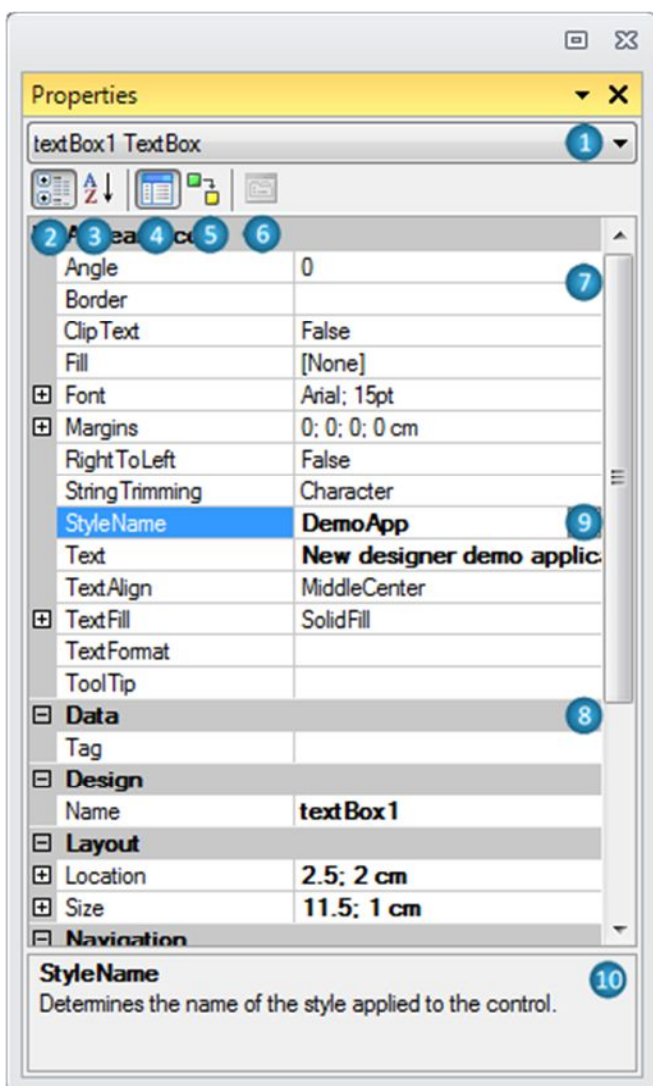


## ドッキング

リボンインターフェイスを持つ新しいレポートデザイナーは、ウィンドウのドッキングをサポートしています。プロパティ、データソース、ドキュメントツリー、エラー一覧のウィンドウをドッキングできます。つまり、デザイナーウィンドウの脇にウィンドウをドッキングさせるか、独自の小さなウィンドウにフローティングさせることができます。

## プロパティグリッド

プロパティグリッドは、選択したオブジェクトのプロパティを表示/編集するためにあります。



**1** - [オブジェクト一覧] - テンプレートのすべてのオブジェクトを表示する、テンプレートのオブジェクト一覧です。プロパティグリッドは、[オブジェクト一覧] で選択したオブジェクトのプロパティを表示します。最初の単語はオブジェクト名で、二番目の単語はオブジェクト型です。

**2** - [項目別] - このボタンはプロパティを項目別で表示します。

**3** - [アルファベット順] - このボタンはプロパティをアルファベット順で表示します。

**4** - [プロパティ] - このボタンは静的なプロパティの表示に切り替えます。

**5** - [バインド] - このボタンは動的なプロパティの表示に切り替えます。

**6** - [プロパティ ページ] - このボタンは使われておりません。

**7** - [プロパティ グリッド] - オブジェクトのプロパティグリッド

**8** - プロパティ群

**9** - プロパティ名と値

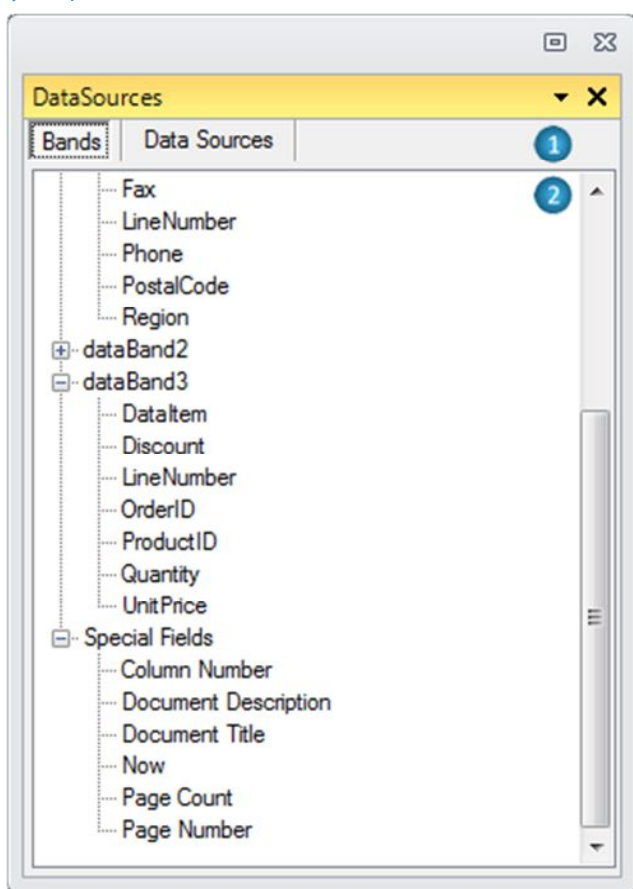
## 10 - [説明] - 選択したプロパティの説明

プロパティの設定にさまざまな方法が使用されます。

<b>Design</b>	
Name	dataBand1 <b>1</b>
ShowBottom	True
ShowCaption	True
<b>Data</b>	
DataSource	
FilterExpression	<b>2</b> ...
InstanceCount	0
<b>Data</b>	
DataSource	UserDayaSource <b>3</b> ▼
FilterExpression	
InstanceCount	0

- 1** - テキストや数値を入力するフィールド
- 2** - このボタンは、プロパティのエディタを開きます。
- 3** - 値リスト

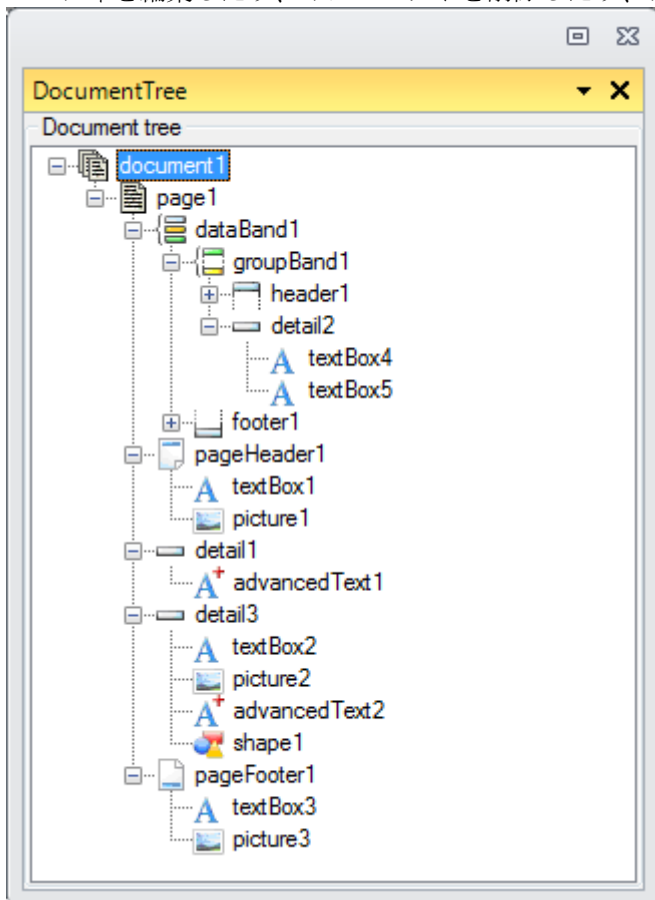
## データソース



- 1** - 「バンド」タブと「データソース」タブを切り替えます。レポートバンドのデータや専用フィールドが「バンド」タブに表示されます。レポートのデータソースのデータが「データソース」タブに表示されます。
- 2** - データソースのフィールドツリーからフィールドをドラッグすることにより、**TextBox** オブジェクトを作成してそこにデータソースのデータを出力できます。

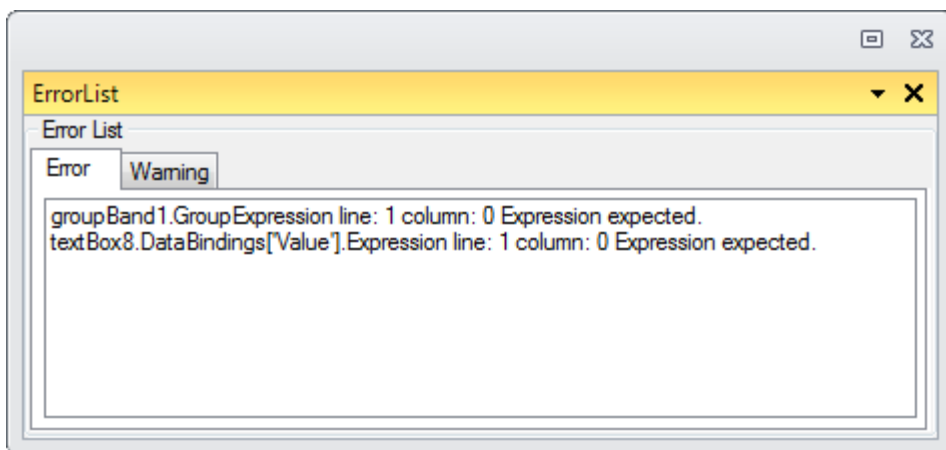
## ドキュメントツリー

ドキュメントツリーはレポート構造順に並べられたオブジェクトのセットです。オブジェクトを選択して、プロパティを編集したり、オブジェクトを削除したり、テンプレート構造を変更したりすることができます。



## エラー一覧

スクリプトのエラー一覧には、エラーの発生したバンド名やエラー内容も表示されます。左マウスボタンでダブルクリックしてスクリプトエディタを開き、式/スクリプトの編集を行えます。



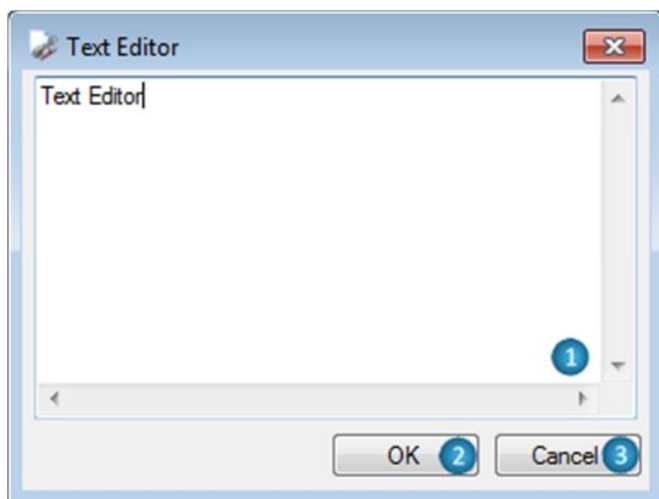
## エディタ

### テキストエディタ

このエディタはテキストの値を入力するためのもので、シンプルなテキストを入力できます。

このエディタは **TextBox** オブジェクトをダブルクリックするか、**TextBox** オブジェクトを選択した時にホットキー : **Ctrl + Enter** を入力するか、プロパティグリッドの **TextBox** プロパティのボタンで開きます。





**1** - テキストを入力する領域

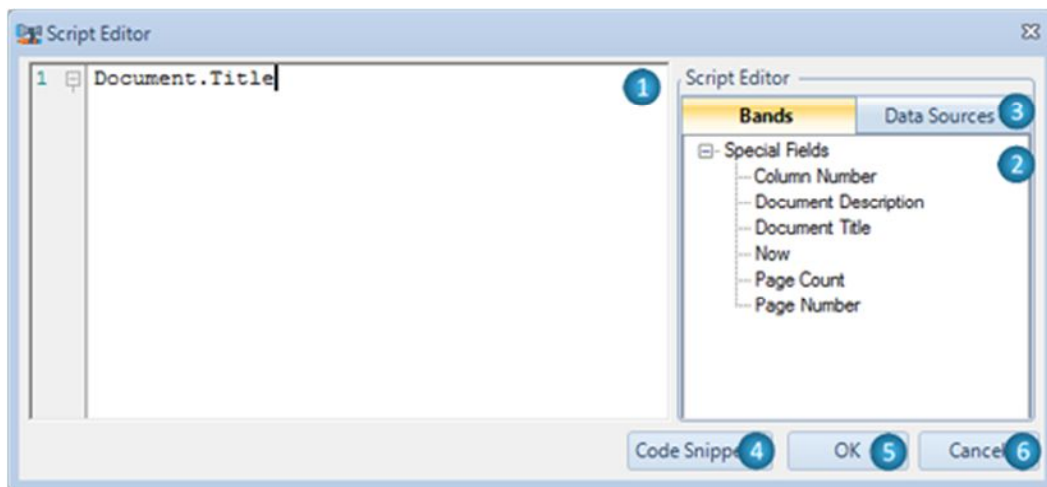
**2** - [OK] ボタン - 変更を適用し、エディタを閉じます。

**3** - [キャンセル] ボタン - 変更をキャンセルし、エディタを閉じます。

### スクリプトエディタ

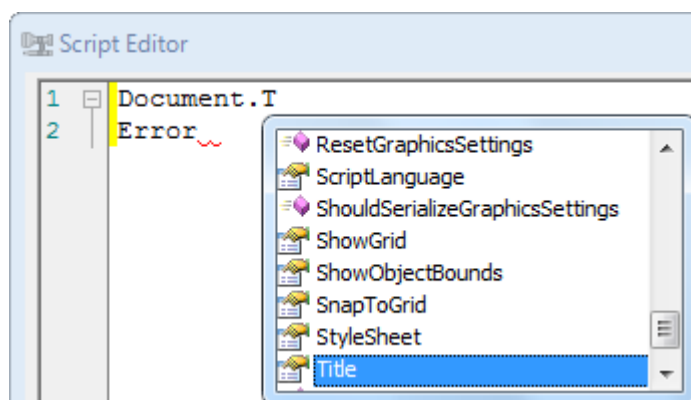
このエディタはスクリプトや式を入力するためのものです。

このエディタは、プロパティグリッドのプロパティのボタンを押すか、バインドバーの「スクリプトエディタ」ボタンを押して開きます。



**1** - スクリプトを入力する領域

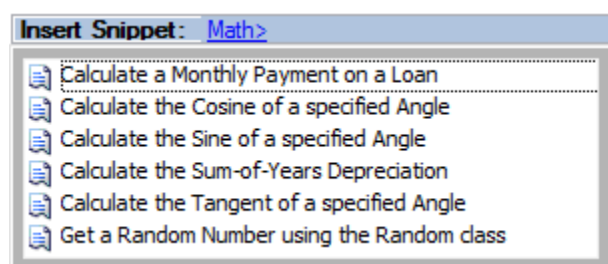
スクリプトエディタは、構文の強調表示、選択の変更、オートコンプリートコードを提供します。



2 - フィールドツリーはデータソースのフィールド（「データソース」タブのデータソースのフィールド部分）、バンドのフィールド（「バンド」タブのバンドのフィールド部分）、特別なフィールド（バンドタブの部分）を表示します。式を素早く入力するには、フィールドツリーのフィールドをスクリプトの入力領域にドラッグするか、フィールドツリーのフィールドをダブルクリックすると行えます。

3 - 「バンド」タブと「データソース」タブを切り替えます。

4 - [コードスニペット] ボタン - テンプレートで選択したオブジェクトに対してコードフラグメントを入力するためのリストを開きます。



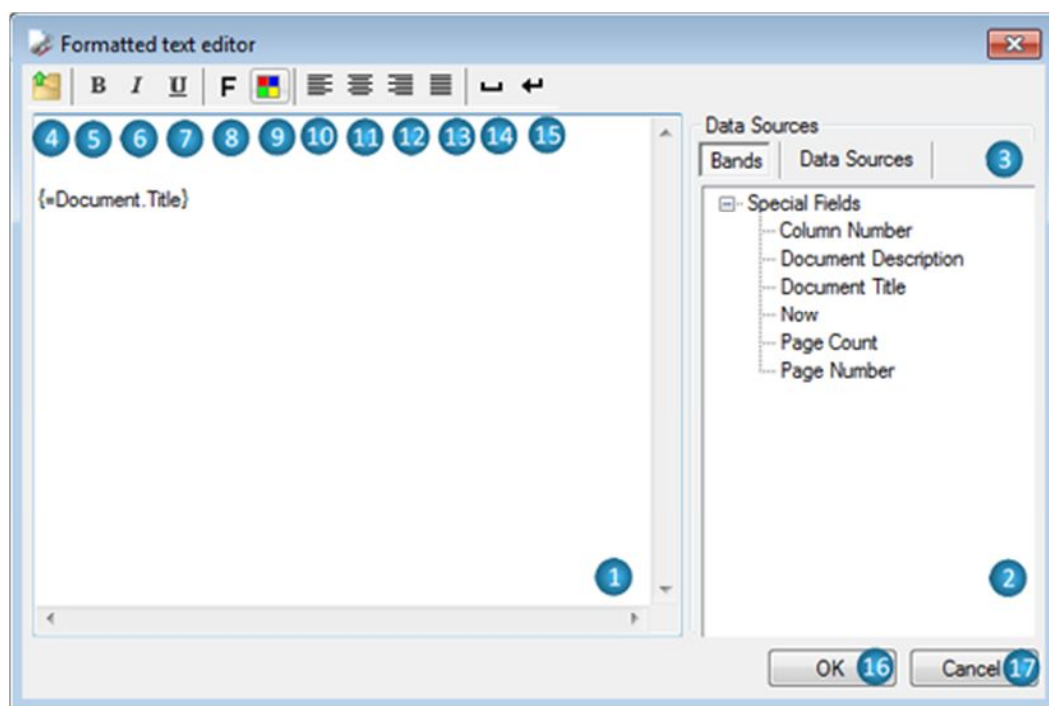
5 - [OK] ボタン - 変更を適用し、エディタを閉じます。

6 - [キャンセル] ボタン - 変更をキャンセルし、エディタを閉じます。

### 書式付きテキストエディタ

このエディタは HTML 形式の書式付きのテキスト、同様のマークアップ言語を入力したり RTF テキストを読み込んだりするためのエディタです。

このエディタは、AdvancedText オブジェクトをダブルクリックするか、AdvancedText オブジェクトを選択した時にホットキー：Ctrl + Enter を入力するか、プロパティグリッドの AdvancedText オブジェクトの Text プロパティのボタンで開きます。



1 - テキストを入力する領域

2 - フィールドツリーはデータソース（「データソース」タブ）やバンドのフィールドを表示したり、特別なフィールド（「バンド」の部分）を表示したりします。式はフィールドツリーからスクリプトの入力領域にドラッグするか、ツリーのフィールドをダブルクリックすることで素早く入力することができます。

3 - 「バンド」タブと「データソース」タブを切り替えます。

4 - [RTF ドキュメントを開く] ボタン - RTF ファイルを選択するための「開く」ダイアログを表示します。

5 - [太字] ボタン - 選択したテキストを太字にするために `<b></b>` タグを挿入します。

6 - [斜体] ボタン - 選択したテキストを斜体にするために `<i></i>` タグを挿入します。

7 - [下線] ボタン - 選択したテキストに下線を付けるために `<u></u>` タグを挿入します。

8 - [フォント] ボタン - 選択したテキストにスタイルやフォントを設定するために「フォントエディタ」を開いて、テキストに `<font=></font>` タグを挿入します。

9 - [色] ボタン - 選択したテキストに色を設定するために「色の設定」エディタを開いて、テキストに `<font color=></font>` タグを挿入します。

10 - [左揃え] ボタン - 選択したテキストを左揃えにするために、テキストに `<p></p>` タグを挿入します。

11 - [中央揃え] ボタン - 選択したテキストを中央揃えにするために、テキストに `<p align="center"></p>` タグを挿入します。

12 - [右揃え] ボタン - 選択したテキストを右揃えにするために、テキストに `<p align="right"></p>` タグを挿入します。

13 - [両端揃え] ボタン - 選択したテキストを両端揃えにするために、テキストに `<p align="justify"></p>` タグを挿入します。

14 - [スペースを挿入] ボタン - スペースを表示するために、テキストに `&nbsp;` タグを挿入します。

15 - [改行を挿入] ボタン - 改行するために、テキストに `<br/>` タグを挿入します。

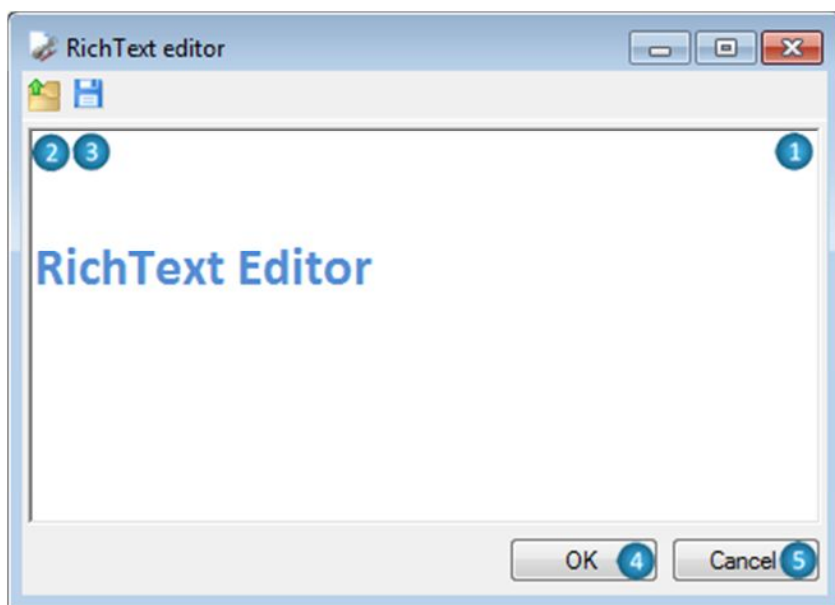
**16** - [OK] ボタン - 変更を適用し、エディタを閉じます。

**17** - [キャンセル] ボタン - 変更をキャンセルし、エディタを閉じます。

### RichText エディタ

このエディタで RTF テキストを読み込んだり編集したりします。

このエディタは RichText オブジェクトをダブルクリックするか、RichText オブジェクトをを選択した時にホットキー : **Ctrl + Enter** を入力するか、プロパティグリッドの RichText オブジェクトの RTFText プロパティのボタンで開きます。



**1** - テキストを入力する領域

**2** - [ファイルを開く] ボタン - RTF ファイルを選択するために「開く」ダイアログを表示します。

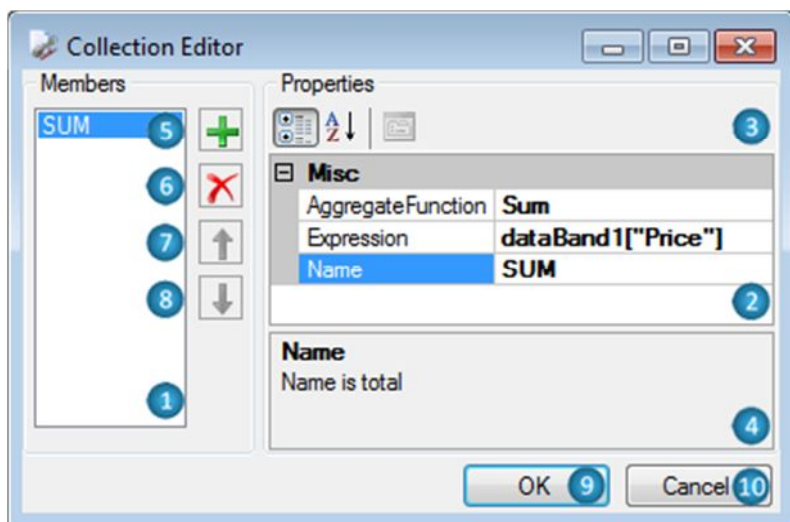
**3** - [ファイルに保存] ボタン - テキストを RTF ファイルに保存するために「名前を付けて保存」ダイアログを表示します。

**4** - [OK] ボタン - 変更を適用して、エディタを閉じます。

**5** - [キャンセル] ボタン - 変更をキャンセルして、エディタを閉じます。

### コレクションエディタ

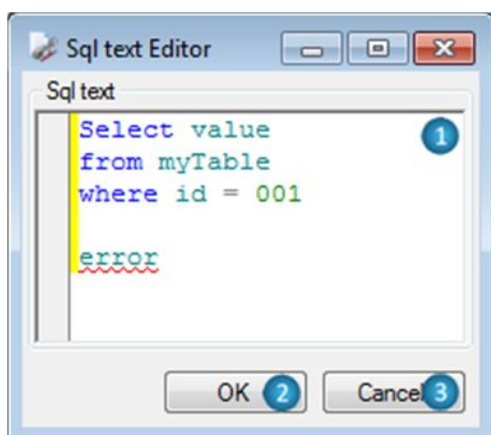
コレクションエディタは、コレクションにメンバーを入力したりそのプロパティを設定したりします。このエディタはプロパティグリッドのプロパティの「(コレクション)」の隣にあるボタンで呼び出されます。



- 1 - コレクションのメンバー一覧。Name プロパティがリストに表示されます。
- 2 - プロパティグリッド - コレクションのメンバーのプロパティの表。プロパティの編集が行えます。
- 3 - プロパティグリッドのプロパティの表示モードの切り替え
- 4 - この領域は、選択したテキストの説明を表示します。
- 5 - このボタンはコレクションに新しいメンバーを追加します。
- 6 - このボタンは選択したメンバーをコレクションから削除します。
- 7 - このボタンはリストのメンバーを上に移動します。
- 8 - このボタンはリストのメンバーを下に移動します。
- 9 - OK - このボタンは変更を適用してエディタを閉じます。
- 10 - キャンセル - このボタンは変更をキャンセルしてエディタを閉じます。

### SQL テキスト エディタ

SQL コマンドを入力するためのエディタ。この SQL テキストエディタはデータソースを設定するためのもので、ドキュメントのデータソース エディタからボタンで開きます。このボタンは、プロパティグリッドの SelectQuery プロパティにあります。



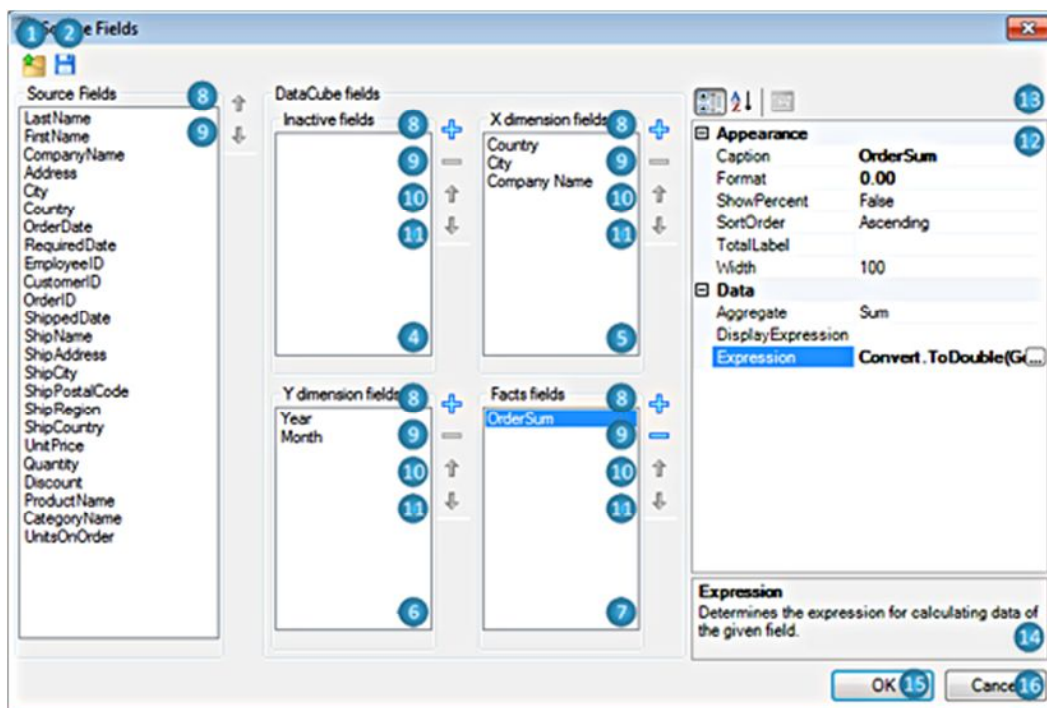
- 1 - SQL コマンドの出力領域。SQL スクリプト コマンド エディタは構文を強調表示したり選択を変更したりします。

2 - OK - 変更を適用してエディタを閉じます。

3 - キャンセル - 変更をキャンセルしてエディタを閉じます。

### ソース フィールド

ピボットテーブルのデータエディタは、データ形式やレイアウトを設定できます。このエディタは PivotTable の Layout プロパティのボタンか、ピボットテーブルウィザードの「レイアウトの構成」ボタンで開きます。



1 - [開く] ボタン - データ設定を読み込むために、「開く」ダイアログウィンドウを開いて Data cube layout ファイル (.dcl) を選択できます。

2 - [保存] ボタン - データ設定を保存するために「名前を付けて保存」ダイアログウィンドウを開きます。

3 - ソースフィールド - データソースのフィールド一覧

4 - 除外するフィールド - 除外するフィールド一覧

5 - X ディメンションフィールド - ピボットテーブルの水平方向の見出しのフィールド一覧

6 - Y ディメンションフィールド - ピボットテーブルの垂直平方向の見出しのフィールド一覧

7 - ファクトフィールド - ピボットテーブルのデータのフィールド一覧

8 - [新規フィールド] ボタン - 一覧に新規フィールドを追加します。

9 - [削除] ボタン - 選択したフィールドを一覧から削除します。

10 - [上] ボタン - 一覧のフィールドを上に移動します。

11 - [下] ボタン - 一覧のフィールドを1つ下に移動します。

12 - プロパティグリッド - プロパティの編集が行えるプロパティグリッドのフィールド

13 - プロパティグリッドのプロパティの表示モードを切り替えます。

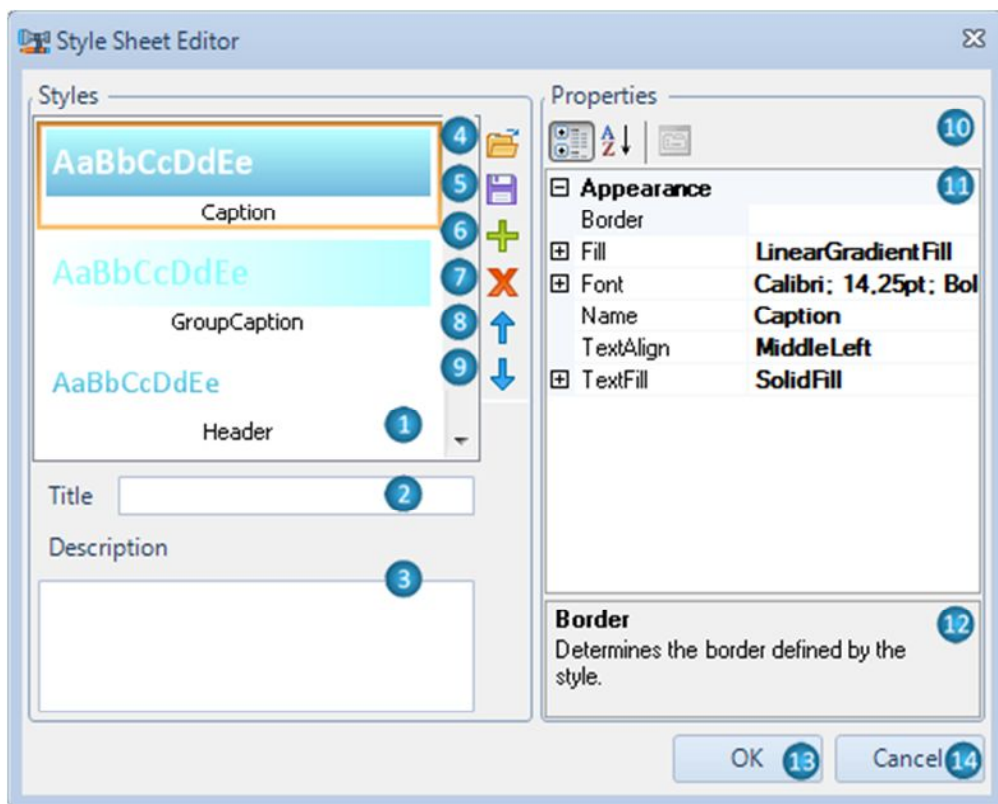
14 - 説明 - 選択したプロパティの説明を表示する領域です。

15 - [OK] ボタン - 変更を適用してエディタを閉じます。

16 - [キャンセル] ボタン - 変更をキャンセルしてエディタを閉じます。

### スタイルシート エディタ

スタイルシートエディタはテンプレートのスタイルを作成したり編集したりすることができます。このエディタは、「ホーム」リボンタブと（オブジェクトを選択すると表示される）「形式」リボンタブの [スタイルの編集] ボタンで開きます。



1 - スタイル - スタイルの一覧

2 - タイトル - 設定したスタイル名

3 - 説明 - 設定したスタイルの説明を表示します。

4 - [開く] ボタン - スタイルを読み込むために、「スタイルシートの読込」ダイアログウィンドウを開いて StyleSheet ファイル (.rss) を選択できます。

5 - [保存] ボタン - 設定したスタイルを保存するために、「スタイルシートの保存」ダイアログウィンドウを開きます。

6 - [スタイルの追加] ボタン - 新しいスタイルを追加します。

7 - [スタイルの削除] ボタン - 選択したスタイルを削除します。

8 - [スタイルを上に移動] ボタン - 一覧で選択したスタイルを上に移動します。

9 - [スタイルを下に移動] ボタン - 一覧で選択したスタイルを下に移動します。

10 - プロパティグリッドのプロパティの表示モードを切り替えます。

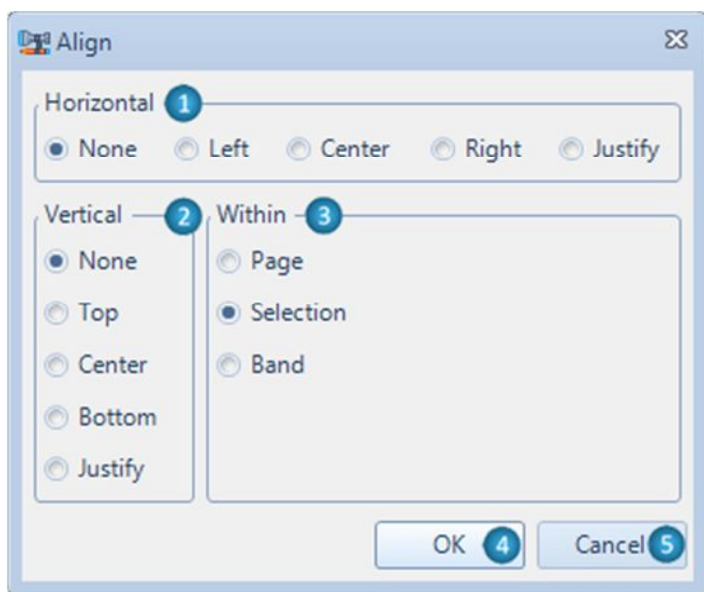
11 - プロパティグリッド - スタイルのプロパティグリッドで、Border、Fill、Font、Name、TextAlign、TextFill プロパティの編集が行えます。



- 12 - 説明 - 選択したプロパティの説明を表示します。
- 13 - [OK] ボタン - 変更を適用してエディタを閉じます。
- 14 - [キャンセル] ボタン - 変更をキャンセルしてエディタを閉じます。

### 整列エディタ

このエディタはテンプレートのオブジェクトの位置を設定できます。このエディタは「レイアウト」リボンタブの [整列] ボタンで開きます。



- 1 - 水平 - 選択したオブジェクトの水平方向の位置やサイズを変更できます。

なし - 位置合わせなし

左 - 最左端のオブジェクトの左枠に揃えます (Location プロパティを設定します)。

中央 - オブジェクトを最右端と最左端の中央に揃えます (Location プロパティを設定します)。

右 - 最右端のオブジェクトの右枠に揃えます (Location プロパティを設定します)。

両端揃え - オブジェクトの幅を右側のオブジェクトから左側のオブジェクトまでの間隔で設定します (Size と Location プロパティを設定します)。

- 2 - 垂直 - 選択したオブジェクトの垂直方向の位置やサイズを変更できます。

なし - 位置合わせなし

上 - 最上端の上枠に揃えます (Location プロパティを設定します)。

中央 - オブジェクトを最上端と最下端の中央に揃えます (Location プロパティを設定します)。

下 - 最下端のオブジェクトの下枠に揃えます (Location プロパティを設定します)。

両端揃え - オブジェクトの高さを上端のオブジェクトから下端のオブジェクトまでの間隔で設定します (Size と Location プロパティを設定します)。

- 3 - 範囲 - 選択した要素の位置を中央に変更できます。

ページ - 選択したオブジェクトが貼り付けられているページの中央に揃えます (Location プロパティを設定します)。



選択 - 選択したオブジェクトの領域を中央に揃えます (Location プロパティを設定します)。

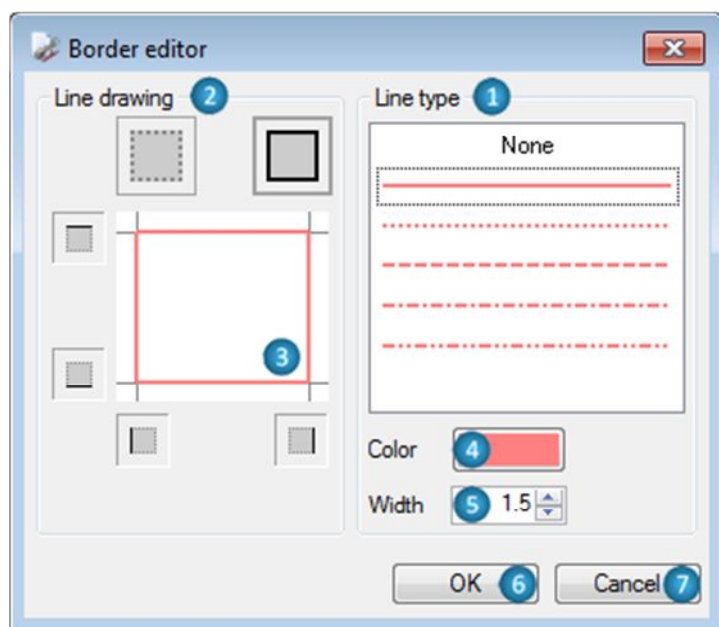
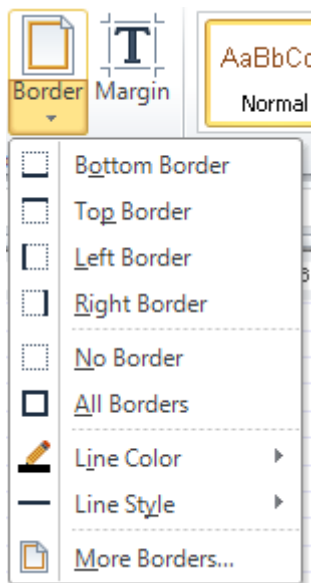
バンド - 選択したオブジェクトが貼り付けられているバンドコンテナの中央に揃えます (Location プロパティを設定します)。

4 - [適用] ボタン - 変更を適用します。

5 - [閉じる] ボタン - エディタを閉じます。

### 枠エディタ

オブジェクトの枠を設定するエディタです。このエディタはプロパティグリッドの **Border** プロパティのボタンか、「ホーム」タブや一部のコンテキストタブの「枠」ボタンで開きます。



1 - 線の種類 - 線の種類の一覧で、枠線を選択できます。

2 - 描画する線- 枠線の位置を選択する領域

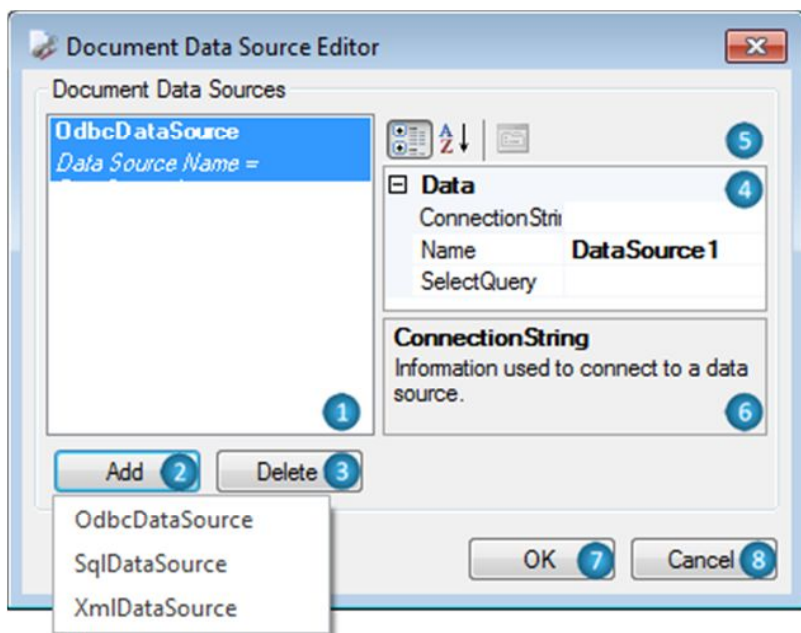
3 - 設定した枠を表示する領域

4 - [色] ボタン - 枠線の色を選択するために色エディタを開きます。

- 5 - 幅 - このカウンタは枠線の幅を設定します。
- 6 - [OK] ボタン - 変更を適用してエディタを閉じます。
- 7 - [キャンセル] ボタン - 変更をキャンセルしてエディタを閉じます。

### ドキュメントのデータソース

データソースを組み込むためのコレクションエディタで、テンプレートにデータソース (Document.DataSources プロパティ) を設定できます。このエディタは、プロパティグリッドの Document オブジェクトの DataSource プロパティのボタンで開きます。



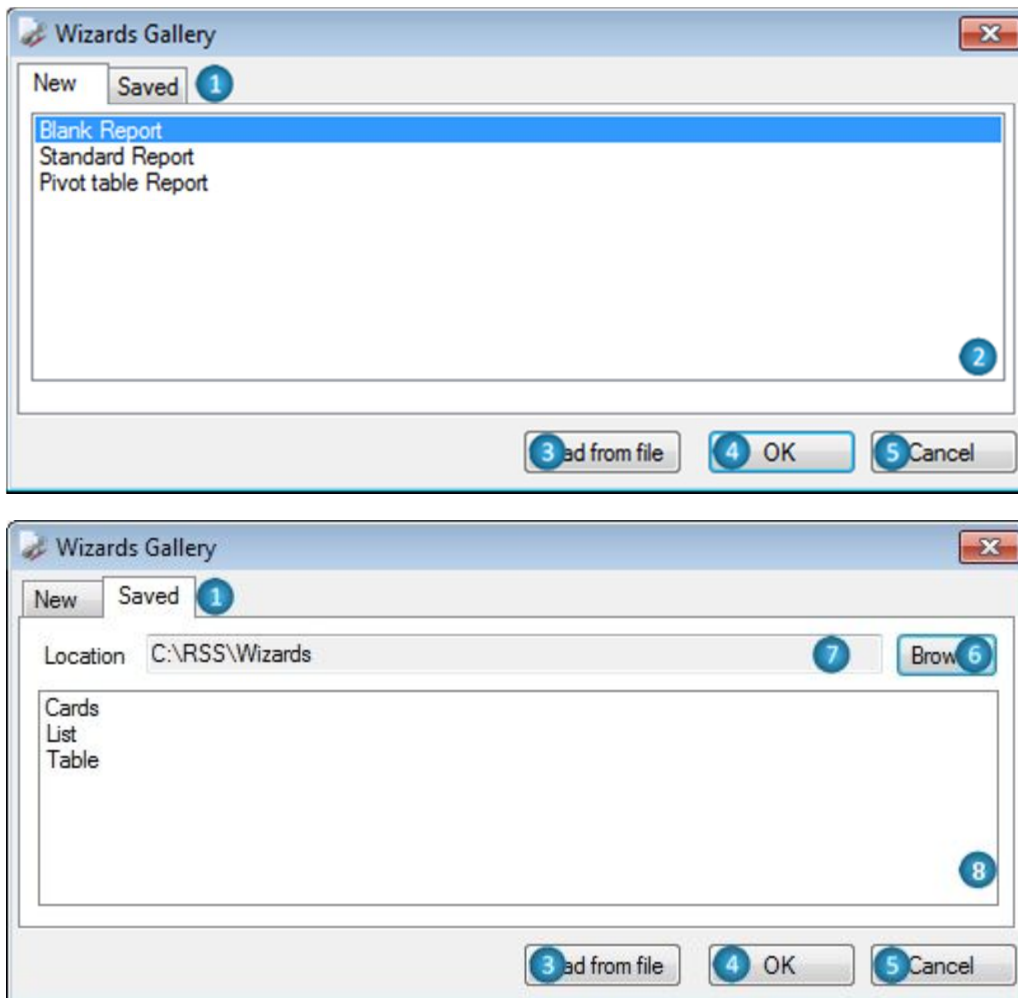
- 1 - ドキュメントのデータソース - ドキュメントのデータソースの一覧
- 2 - [追加] ボタン - テンプレートに追加するデータソース型 (SqlDataSource、OdbcDataSource、XmlDataSource) の一覧を開きます。
- 3 - [削除] ボタン - 選択したデータソースを削除します
- 4 - プロパティグリッド - データソースのプロパティグリッド。プロパティを編集できます。
- 5 - プロパティグリッドのプロパティの表示モードの切り替え
- 6 - 説明 - この領域は選択したテキストの説明を表示します。
- 7 - [OK] ボタン - 変更を適用してエディタを閉じます。
- 8 - [キャンセル] ボタン - 変更をキャンセルしてエディタを閉じます。

## ウィザード

ウィザードは、シンプルレポートを素早く作成するために使用します。

### ウィザードギャラリー

シンプルレポートの作成方法を選択できます。ウィザードギャラリーは[ファイル/新規]メニューまたはキーの組み合わせ (Ctrl + N) で呼び出されます。

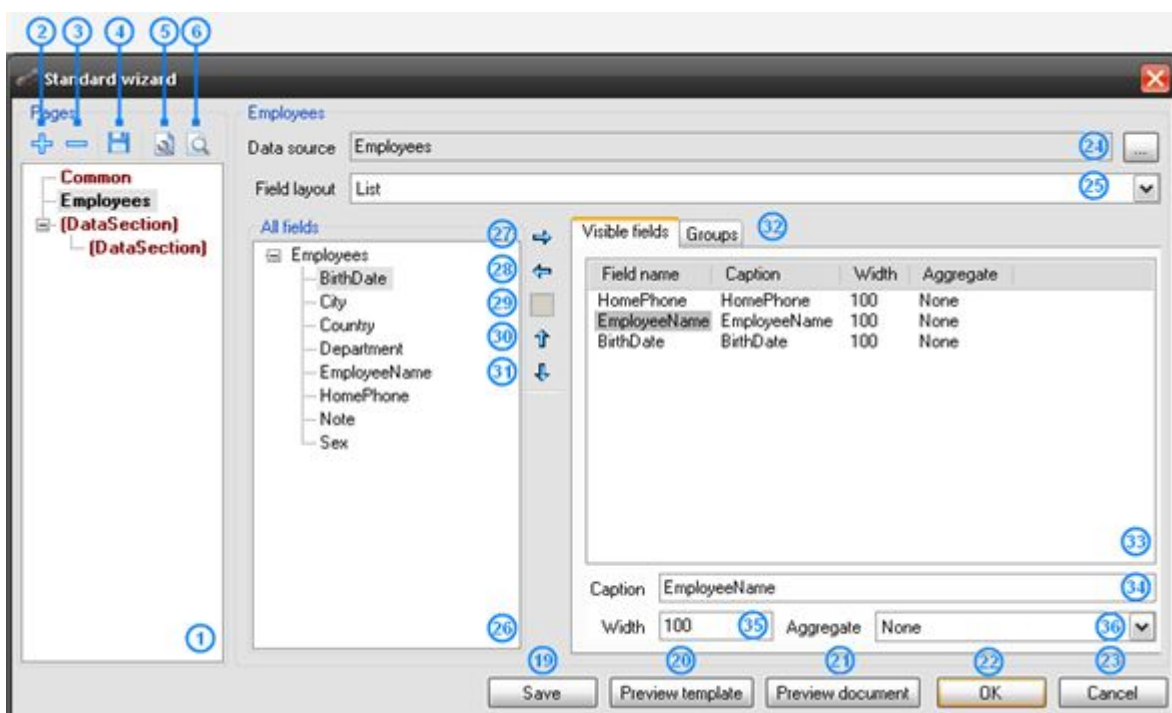
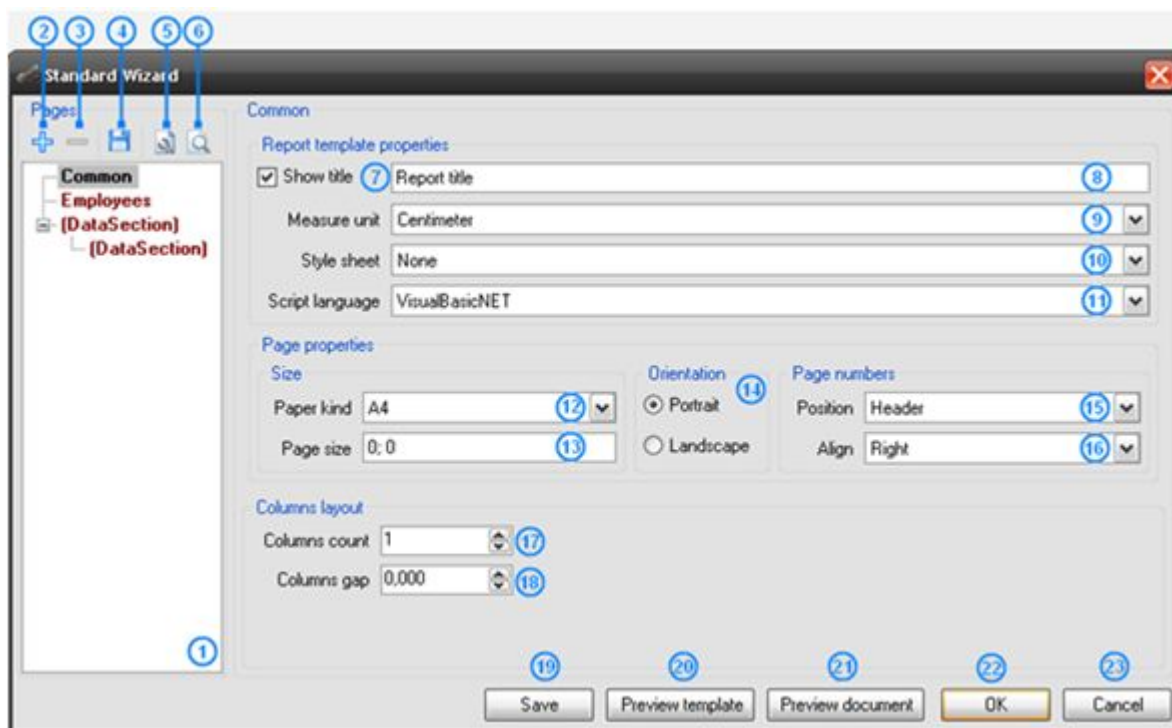


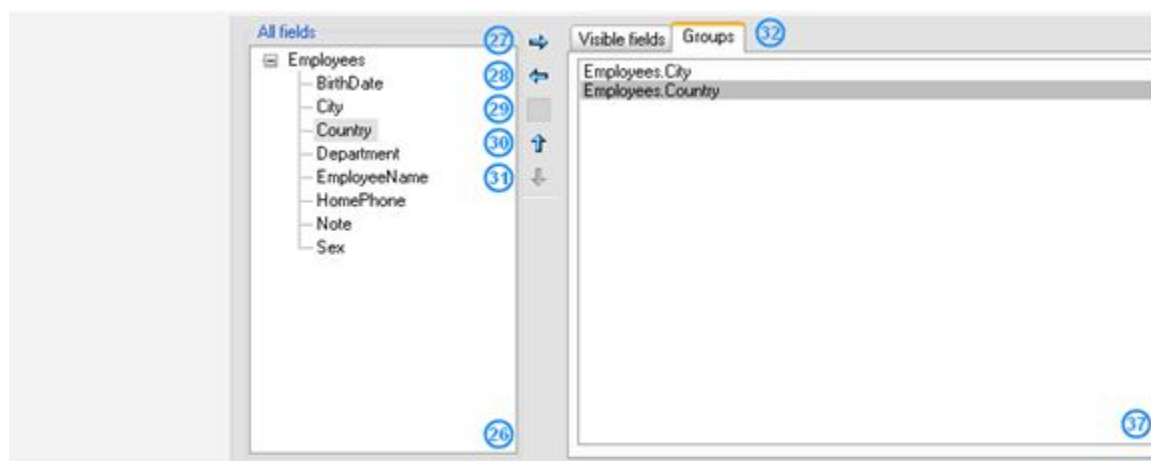
- 1 - 「新規」タブと「保存」タブを切り替えます。
- 2 - 新規レポートの作成方法のリスト：
  - 白紙 - 新しい白紙のレポートを作成します。
  - 標準のレポート - スタンダードウィザードを呼び出します。
  - ピボットテーブルレポート - ピボットテーブルウィザードを呼び出します。
- 3 - [ファイルから読込] ボタン - 「ウィザードファイルを開く」ダイアログを開きます。
- 4 - [OK] ボタン - 新しい白紙のテンプレートを作成するか、選択したパラメータのウィザードを開きます。
- 5 - [キャンセル] ボタン - ウィザードギャラリーを閉じます。
- 6 - [参照] ボタン - 保存してあるウィザードファイルのフォルダを選択するために「フォルダの参照」ダイアログを呼び出します。
- 7 - 位置 - このフィールドは保存してあるウィザードファイルのフォルダのアドレスを表示/設定します。

## 8 - 「位置」フィールドに指定したフォルダにあるウィザードファイルの一覧

### スタンダードウィザード

スタンダードウィザードは標準のレポート（リストやグループを含む）を作成できます。スタンダードウィザードは、ウィザードギャラリーダイアログから呼び出されます。





**1** - このデータツリーはレポートの共通設定と追加したデータセクションを表示します。設定モードの切り替えが可能。

**2** - [データの追加] ボタン - データセクションを追加します。

**3** - [データの削除] ボタン - データセクションを削除します。

**4, 19** - [ウィザードの保存] ボタン - ウィザードの設定を保存先フォルダやファイル名を選択するために、「Save Wizard File」ダイアログウィンドウを開きます。

**5, 20** - [テンプレートのプレビュー] ボタン - 設定したパラメータでテンプレートを作成し、レポートデザイナーに表示します。

**6, 21** - [ドキュメントのプレビュー] ボタン - 設定したパラメータでレポートを生成し、レポートビューアに最終ドキュメントを表示します。

**7** - タイトルを表示 - レポートにタイトルを表示するかを設定します。

**8** - レポートのタイトル - このフィールドはレポート名を設定します (Document.Title プロパティを設定します)。

**9** - 計測単位 - 計測単位のリストで、テンプレートの計測単位を選択できます (Document.MetricUnit を設定します)。

**10** - スタイルシート - スタイルの一覧で、レポートのスタイルを選択できます (Document.StyleSheet プロパティを設定します)。

**11** - スクリプト言語 - サポートされているプログラミング言語の一覧で、レポートに記述するスクリプト言語を選択できます (Document.ScriptLanguage プロパティを設定します)。

**12** - 用紙サイズ - レポートの用紙サイズのリスト (Page.PaperKind プロパティを設定します)。

**13** - ページサイズ - このフィールドはレポートの用紙サイズを設定します (Page.CustomSize プロパティを設定します)。

**14** - 向き - このラジオボタンはレポートの用紙の向きを設定します (Page.Orientation プロパティを設定します)。

**15** - 位置 - 見出しの位置を選択する値リスト。テンプレートにページ番号を表示するための見出しの位置 (なし、ヘッダー、フッター、両方) を選択できます。

**16** - 整列 - 見出しに表示するページ番号 (テキスト) の位置を選択するための値リスト (左、中央、右)。

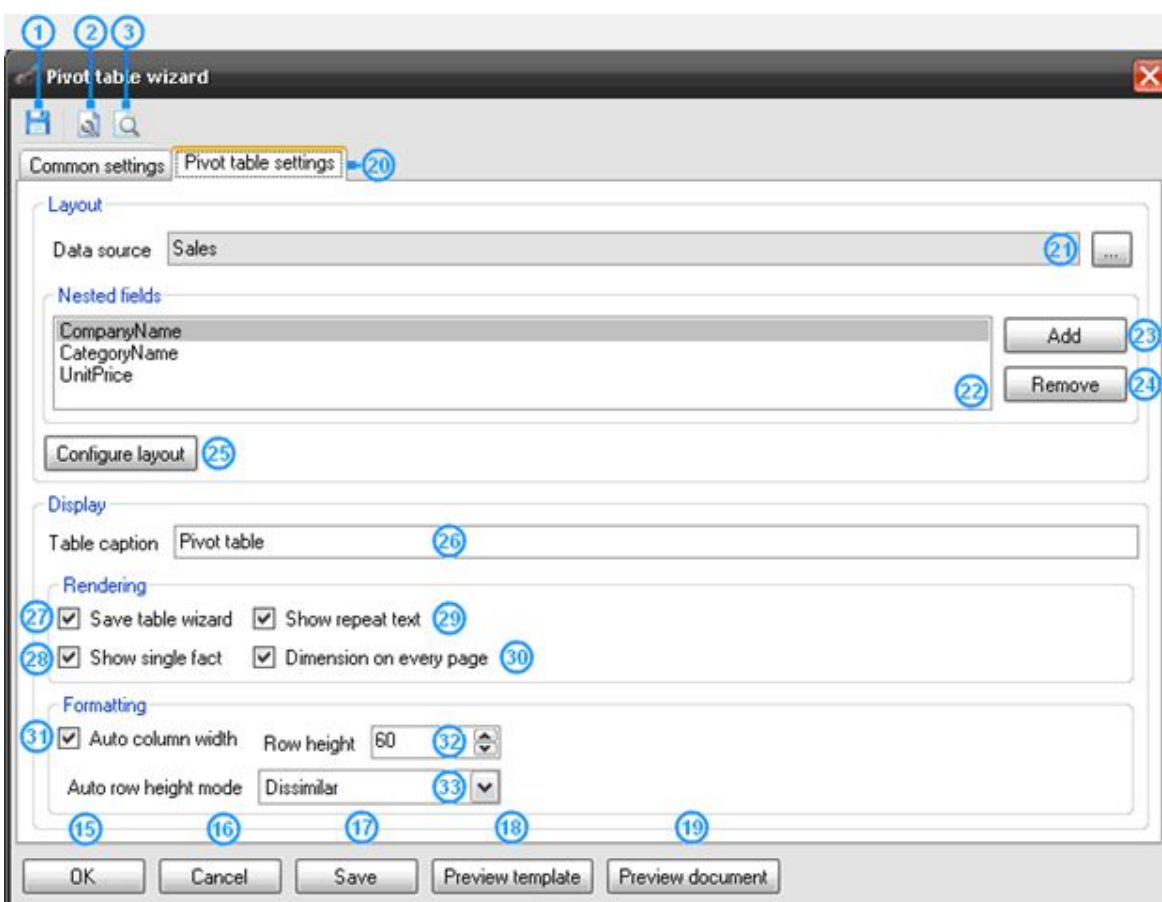
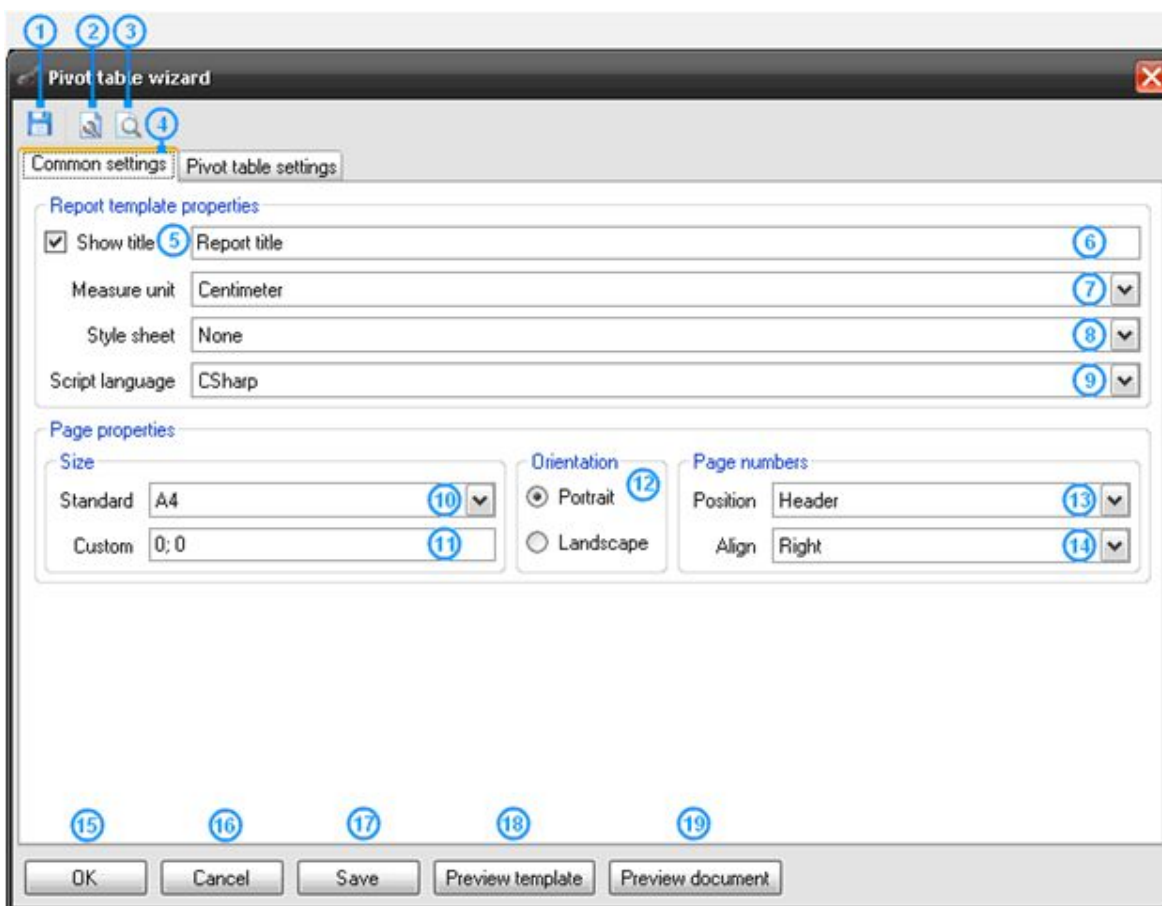


- 17** - 列数 - レポートのリストの列数を設定するカウンタです (`DataBand.ColumnsCount` プロパティを設定します)。
- 18** - 列の間隔 - レポートのリストの列の間隔を設定するカウンタです (`DataBand.ColumnsCount` プロパティを設定します)。
- 22** - [OK] ボタン - ウィザードでの作業を終了し、選択したパラメータでテンプレートを作成し、レポートデザイナーに表示します。
- 23** - [キャンセル] ボタン - ウィザードの設定をキャンセルします。
- 24** - [データソース] ボタン - データソースを選択するデータソースツリーを開きます。フィールドには選択したデータソースの名前が表示されます。
- 25** - フィールドのレイアウト - このリストはデータの表示方法を選択できます。
- 26** - 全フィールド - データソースフィールドのツリー
- 27** - [右に移動] ボタン - データソースツリーから選択したフィールドを「表示フィールド」または「グループ」の一覧に追加します。これにより、`Caption`、`Width`、`Aggregate` プロパティの値は規定値に設定されません。
- 28** - [左に移動] ボタン - 「表示フィールド」または「グループ」から選択したフィールドを削除します。
- 29** - [Add Blank] ボタン - 「表示フィールド」にプロパティが設定されていないフィールドを追加します。
- 30** - [上に移動] ボタン - 「表示フィールド」または「グループ」で選択したフィールドを上に移動します。
- 31** - [下に移動] ボタン - 「表示フィールド」または「グループ」で選択したフィールドを下に移動します。
- 32** - 「表示フィールド」と「グループ」を切り替えます
- 33** - 表示フィールド - これはレポートを作成するためのフィールドの表で、追加したフィールドとそのプロパティを表示します。フィールドはデータソースから追加するか、新しいフィールドを追加できます。`FieldName` プロパティはフィールド名を表示します。`Caption` プロパティは表示フィールドのタイトルを表示します。`Width` プロパティは列幅を設定します。`Aggregates` プロパティは集計関数を設定します。
- 34** - キャプション - このフィールドは「表示フィールド」のレコードの `Caption` プロパティを設定します。レポートに出力するフィールドタイトルを設定するためにオブジェクトの `Textbox.Text` プロパティを設定します。
- 35** - 幅 - このフィールドは「表示フィールド」のレコードの `Width` プロパティを設定し、リストの列幅をパーセントで定義します。
- 36** - 集計 - これは集計関数のリストで、フィールドごとに集計値を計算するための関数を選択できます。値はリストの末尾に出力されます。
- 37** - グループ - リストをグループ化するためのデータソースのフィールド一覧です。

### ピボットテーブル ウィザード

ピボットテーブルウィザードは、ピボットテーブルのレポートを作成できます。ウィザードギャラリー ダイアログから呼び出します。





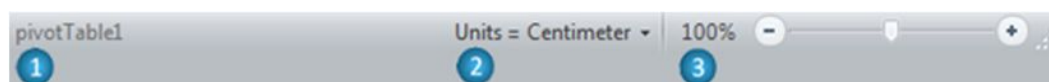


- 1, 17** - [ウィザードの保存] ボタン - ウィザードの設定を保存先フォルダやファイル名を選択するために「名前を付けて保存」ダイアログウィンドウを呼び出します。
- 2, 18** - [テンプレートのプレビュー] ボタン - 選択したパラメータでテンプレートを作成し、レポートデザイナーに表示します。
- 3, 19** - [ドキュメントのプレビュー] ボタン - 選択したパラメータでレポートを生成し、それをレポートビューアに表示します。
- 4** - 共通設定 - レポートの共通設定タブ。
- 5** - タイトルを表示 - レポートにタイトルを表示するかを設定します。
- 6** - レポートのタイトル - レポート名を設定します (**Document.Title** プロパティを設定します)。
- 7** - 計測単位 - テンプレートの計測単位を選択できる計測単位の一覧 (**Document.MetricUnit** プロパティを設定します)。
- 8** - スタイルシート - レポートのスタイルを選択できるスタイルの一覧 (**Document.StyleSheet** プロパティを設定します)。
- 9** - スクリプト言語 - レポートに記述するスクリプト言語を設定できる、サポートされているプログラミング言語の一覧 (**Document.ScriptLanguage** プロパティを設定します)。
- 10** - 標準 - レポートの用紙サイズを選択できる用紙サイズの一覧 (**Page.PaperKind** プロパティを設定します)。
- 11** - カスタム - レポートのページサイズを設定します (**Page.CustomSize** プロパティを設定します)。
- 12** - 向き - レポートのページの向きを設定します (**Page.Orientation** プロパティを設定します)。
- 13** - 位置 - テンプレートの見出しの位置を選択する値リスト。テンプレートにページ番号を表示するための見出しの位置 (なし、ヘッダー、フッター、両方) を選択できます。
- 14** - 整列 - 見出しに表示するテキストの位置を選択できる一覧。
- 15** - [OK] ボタン - ウィザードでの作業を終了し、選択したパラメータでテンプレートを作成し、レポートデザイナーに表示します。
- 16** - [キャンセル] ボタン - ウィザードの設定をキャンセルします。
- 20** - ピボットテーブル設定 - ピボットテーブル設定タブ。
- 21** - [データソース] ボタン - ピボットテーブルのデータソースを選択するツリーを開きます。このフィールドは選択したデータソースを表示します。
- 23** - [追加] ボタン - 入れ子のフィールド一覧に新規フィールドを追加します。
- 24** - [削除] ボタン - 選択したフィールドを入れ子のフィールドの一覧から削除します。
- 25** - [レイアウトの構成] ボタン - ピボットテーブルのデータを編集する [ソースフィールド](#) エディタを開きます。
- 26** - テーブルのタイトル - テーブルのタイトルを設定します (**PivotTable.TableCaption** プロパティを設定します)。
- 28** - 単一ファクトの表示 - ファクト列のタイトルの表示/非表示を設定します (**PivotTable.ShowSingleFact** プロパティを設定します)。

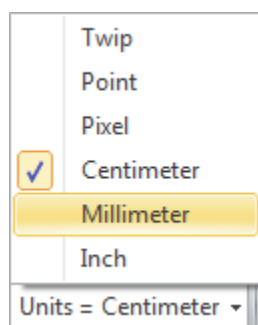


- 29 - テキストの繰り返し表示 - 新しいページにテキストを繰り返し表示するかを設定します (PivotTable.ShowRepeatText プロパティを設定します)。
- 30 - 各ページにディメンションを表示 - 各ページにディメンションを表示するかを設定します (PivotTable.DimensionOnEveryPage プロパティを設定します)。
- 31 - 自動列幅 - 列の規定幅を設定します (PivotTable.AutoColumnWidth プロパティを設定します)。
- 32 - 行の高さ - 行の高さを設定するカウンタ (PivotTable.RowHeight プロパティを設定します)。
- 33 - 自動による行の高さモード - 行の自動高さモードを設定する一覧 (PivotTable.AutoRowHeight プロパティを設定します)。

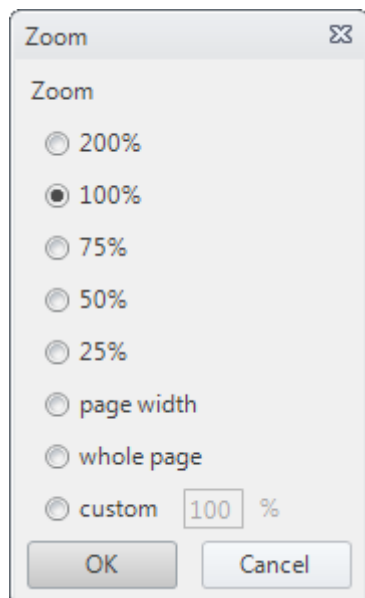
## ステータスバー



- 1 - 現在、カーソルの下にあるオブジェクトの名前。
- 2 - 計測単位を設定するためのコントロール。



- 3 - 手で倍率を入力するために「ズーム」ダイアログを開くボタン。



「ズーム」のスライダは倍率を設定します。



## ホットキー

ホットキー	説明
<b>デザイン</b>	
F2	「選択」モードをオン/オフに切り替えます
F3	「パン」モードをオン/オフに切り替えます
F4	「ズームイン」モードをオン/オフに切り替えます
F5	「ズームアウト」モードをオン/オフに切り替えます
F6	「矩形に合わせてズーム」モードをオン/オフに切り替えます
F7	「ダイナミックズーム」モードをオン/オフに切り替えます
F10	「メニュー」モードをオン/オフに切り替えます
Space + マウスドラッグ	「パン」モードをオン/オフに切り替えます
Ctrl + マウス ホイール	ズーム
<b>クリップボード</b>	
Ctrl + Insert	選択したものをコピーし、クリップボードに貼り付けます。
Ctrl + C	
Shift + Insert	クリップボードの内容を貼り付けます。
Ctrl + V	
Shift + Delete	選択したものを切り取り、クリップボードに貼り付けます。
Ctrl + X	
Ctrl + マウスドラッグ	選択したものをコピーします
Ctrl + Alt + V	クリップボードの内容を新規ページに貼り付けます
<b>テンプレート</b>	
Ctrl + N	新しいテンプレートを作成します
Ctrl + O	ファイルからテンプレート/ドキュメントを読み込みます
Ctrl + S	テンプレートの変更を保存します
<b>ページ</b>	
Ctrl + Page Up	次のページを表示します
Ctrl + Page Down	前のページを表示します
<b>選択</b>	
Ctrl + A	すべて選択
Ctrl + Shift + A	<ol style="list-style-type: none"> <li>1. 現在のコンテナのすべての要素（および子要素）を選択します</li> <li>2. A を押す - 親コンテナを追加します</li> <li>3. A を押す - 親の兄弟要素（およびその子要素）をすべて追加します</li> <li>4. A を押す - 親の親を追加します</li> <li>5. ...</li> </ol>
ドラッグによる移動	（矩形選択によって少なくとも 50% 選択された）すべての要素を選択します
Shift + マウスクリック	選択したものを追加/削除します
Ctrl + マウスクリック	
<b>オブジェクトの操作</b>	
Ctrl + Enter	選択されたオブジェクトのエディタを呼び出します（TextBox、AdvancedText、RichText、Picture、Chart、Widget に適用可能）
Enter	
Ctrl + Enter	変更を適用してエディタを閉じます
何度もクリック	多数のオブジェクトが重なり合っている場合、その重なり合っている領域を何度もクリックすると、手前（z インデックスの小さい）オブジェクト



	が選択されます。（最後のオブジェクトが選択されると、また最前面のオブジェクトが選択されます）
オブジェクトのプロパティ	
Ctrl + B	Font.Bold プロパティを設定します (TextBox、AdvancedText、BarCode に適用可能)
Ctrl + I	Font.Italic プロパティを設定します (TextBox、AdvancedText、BarCode に適用可能)
Ctrl + U	Font.Underline プロパティを設定します (TextBox、AdvancedText、BarCode に適用可能)
Ctrl + Shift + >	フォントサイズを大きくします (TextBox、AdvancedText、BarCode に適用可能)
Ctrl + Shift + <	フォントサイズを小さくします (TextBox、AdvancedText、BarCode に適用可能)
Ctrl + ]	あるポイントのフォントサイズを大きくします (TextBox、AdvancedText、BarCode に適用可能)
Ctrl + [	あるポイントのフォントサイズを小さくします (TextBox、AdvancedText、BarCode に適用可能)
Ctrl + L	テキストを左揃えにします (TextBox、AdvancedText に適用可能)
Ctrl + R	テキストを右揃えにします (TextBox、AdvancedText に適用可能)
Ctrl + E	テキストを中央揃えにします (TextBox、AdvancedText に適用可能)
Ctrl + Shift + F	Font プロパティを設定する「フォント」ダイアログを呼び出します (TextBox、AdvancedText、BarCode に適用可能)
Ctrl + Shift + B	Border プロパティを設定する「枠」エディタを呼び出します (Border プロパティを持っているオブジェクトに適用可能)
Ctrl + Shift + M	Margins プロパティを設定する「余白」エディタを呼び出します (Margins プロパティを持っているオブジェクトに適用可能)
Ctrl + Shift + L	Fill プロパティを設定する「塗りつぶし」エディタを呼び出します (Fill プロパティを持っているオブジェクトに適用可能)
Ctrl + Shift + T	Format プロパティを設定する「フォーマット」エディタを呼び出します (TextBox、AdvancedText に適用可能)
Ctrl + Shift + ]	時計回りに Angle プロパティを設定します (Angle プロパティを持っているオブジェクトに適用可能)
Ctrl + Shift + [	反時計回りに Angle プロパティを設定します (Angle プロパティを持っているオブジェクトに適用可能)
Ctrl + Alt + F	先頭に移動
Ctrl + Alt + B	末尾に移動



## テンプレートのオブジェクト

### 分類

レポート要素はすべて 2 つに分類できます。

バンド : DataBand、PageHeader、PageFooter、Header、Footer、GroupBand、PageOverlay、CrossBand、BandContainer、SideBySide、SubReport、Content、PivotTable

ビジュアルオブジェクト : TextBox、AdvancedText、Picture、Shape、ZipCode、BarCode、RichText、WinFormsControl

### バンド

バンドは、他のオブジェクトのコンテナです。レポートには表示されず、バンドに格納されたオブジェクトの出力方法を定義します。バンドは他のバンドのコンテナとしても使われます。この場合、格納されたオブジェクトの出力方法はそのオブジェクトのコンテナバンドにすべて依存します。バンドの幅は変わらず、常にページ幅と同じになります。ページ上のオブジェクトの位置は、レポート生成時に定義され、レポートのバンド順、バンドの内容、Render グループのプロパティに依存します。

### DataBand

格納されたオブジェクトを複数出力します。行はそれぞれデータソースの 1 レコードと一致します。データバンドのデータソースは DataSource プロパティで指定します。データソースが指定されていない場合は、InstanceCount プロパティに出力回数を設定します。

DataBand のコンテナとして使用可能なバンド : Page、DataBand、BandContainer、GroupBand、SideBySide。

### Detail

コンテナバンドのデータソースのデータにアクセスしたり、様々なバンドでビジュアル オブジェクトを出力したりするために使用します。Detail バンドの高さは、格納されたオブジェクトが表示されるように設定します。

Detail のコンテナとして使用可能なバンド : Page、DataBand、BandContainer、GroupBand、SideBySide

### PageHeader と PageFooter

ページの上部と下部に見出しを付けます。PageHeader/PageFooter はコンテナであり、最終レポートに組み込まれたオブジェクトの前後の間隔は、PageHeader/PageFooter からテンプレート内のバンドの上枠/下枠までの間隔で指定されます。PageHeader/PageFooter の位置は、最終レポートのオブジェクトの位置には影響を及ぼしません。

PageHeader と PageFooter のコンテナとして使えるのは Page だけです。

### Header と Footer

DataBand や GroupBand によって出力されるリストのページにヘッダーとフッターを作成します。Header の内容は、GroupBand によって出力されるリストの先頭または各グループの見出しに表示されます。Footer の内容は、リストまたはグループの末尾に表示されます。

Header と Footers のコンテナとして DataBand と GroupBand を使用できます。

### GroupBand

レポート生成時に計算される式に応じて、データをグループ化します。

DataBand や GroupBand は、GroupBand のコンテナとして使用できます。



## PageOverlay

ページの下層レイヤーを作成します。最終ドキュメントのページの先頭から格納されたオブジェクトまでの距離は、テンプレートの **PageOverlay** の上枠からのインデントにより設定されます。**PageOverlay** は、互いに重なり合うことが可能な他のオブジェクトの位置付けには何の影響も及ぼしません。

**PageOverlay** のコンテナとして使用できるのは **Page** だけです。

## CrossBand

格納されたオブジェクトを左から右方向の行として出力します。反復回数は **InstanceCount** プロパティか、データソースのレコード数で指定します。クロスバンドは表の作成に使用できます。

**CrossBand** のコンテナとして使用可能なバンド : **Header**、**Footer**、**Detail**

## BandContainer

コンテンツを一度に出力します。様々な可変バンドを結合するために、複数のバンドを 1 つの結合された **SideBySide** にまとめるために使用します。

**BandContainer** のコンテナとして使用可能なバンド : **Page**、**DataBand**、**BandContainer**、**GroupBand**、**SideBySide**

## SideBySide

さまざまなバンドのオブジェクトを並列出力にするために使用します。

**SideBySide** のコンテナとして使用可能なバンド: **Page**、**DataBand**、**BandContainer**、**GroupBand**、**SideBySide**

## SubReport

一緒に生成する、埋め込まれたレポートを追加できます。サブレポートとして使用するテンプレートから、バンド (例 : **DataBand**、**Detail** など) だけが追加されます。

**SubReport** のコンテナとして使用可能なバンド: **Page**、**DataBand**、**BandContainer**、**GroupBand**、**SideBySide**

## Content

親レポートで使用します。このバンドは最終レポートのコンテンツを置く位置を指定します。

**Content** のコンテナとして使用可能なバンド : **Page**、**DataBand**、**BandContainer**、**GroupBand**、**SideBySide**

## PivotTable

静的なデータを基にしたクロスレポートや集計表の作成にも使用されます。**PivotTable** の使用結果は、上述の **CrossBand** のビジュアル表現を含んでいるレポート部分です。

クロス集計の特別なエディタを使用して、計算に使用するデータを指定できます。スクリプトを使用して、計算命令を作成できます。これは、データソースから複数のフィールドを合計することから、必要とするレポートにさまざまなグループ化を行うことまで、表の設定機能を提供します。

**PivotTable** のコンテナとして使用可能なバンド : **Page**、**DataBand**、**BandContainer**、**GroupBand**、**SideBySide**

## ビジュアルオブジェクト

レポートでビジュアル表現を持ったオブジェクト。バンドに格納することにより位置を指定します。ビジュアルオブジェクトのサイズや表示/非表示はプロパティで指定します。

## TextBox

テキスト情報を出力します。テキスト情報は、静的またはレポートの生成時に作成されます。



### AdvancedText

データを書式付きのテキストとして出力します。段落やテキストスタイルを設定したり、記述したテキストに直接式を使用するオプションがあります。書式付きのテキストは、HTML のようなマークアップ言語や RTF 形式のサブセットを使用して設定できます。そのために書式付きテキストエディタを使用します。静的な RTF テキストを読み込みます。AdvancedText は、一部の RTF のサブセットのみサポートします。

### Picture

イメージを出力します。ハードドライブまたはデータソースからイメージを読み込むことができます。

### Shape

様々な幾何学的図形を出力します。

### ZipCode

郵便番号を出力します。

### BarCode

様々なバーコード型を出力します。使用するバーコード型に応じて、設定がいくつかあります。

### RichText

テキストを RTF 形式で出力します。エディタで RTF ファイルを選択できます。

### WinFormsControl

レポートに WinFormsControl コンポーネントを格納します。コントロールのイメージだけがレポートに格納されます（この機能は使用できません）。

## ユーザーガイド

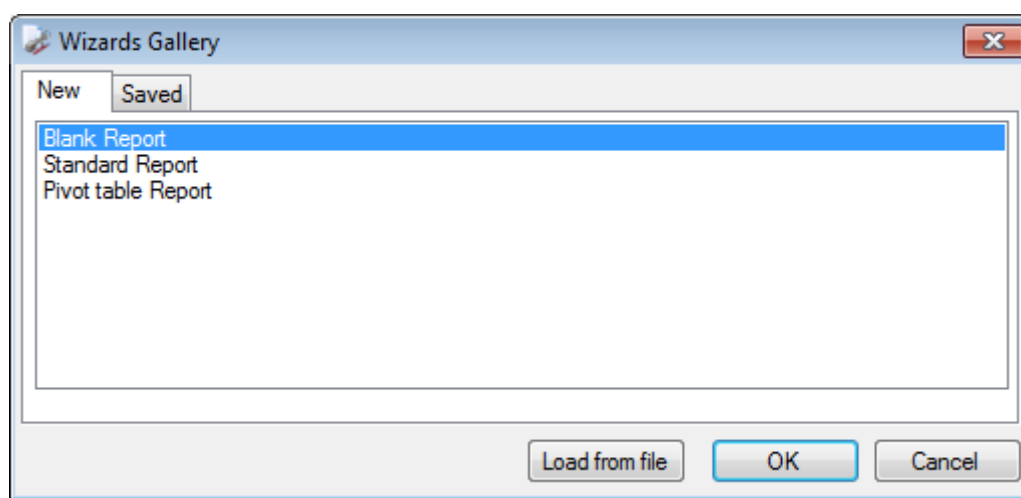
### テンプレート

レポートテンプレートは、バンドやオブジェクトを含んでいる複数のページです。

### テンプレートの制御

#### テンプレートの作成

メインメニューの [ファイル] -> [新規] を選択するか、ツールバーの「新規ドキュメント」ボタンを押すと、新しいテンプレートを作成するためのウィザードギャラリーが表示され、そこでテンプレートの作成方法を選択できます。



レポート	説明
白紙のレポート	白紙のテンプレートを作成します
標準のレポート	標準のテンプレート（リストまたはデータカード）を作成するためにスタンダードウィザードを開きます
ピボットテーブルレポート	ピボットテーブルレポートを作成するためにピボットテーブルウィザードを開きます

#### テンプレートを開く

ダイアログ ウィンドウを開くには、メインメニューの [ファイル] -> [開く] を選択するか、ツールバーの「テンプレートを開く」ボタンを押してください。このウィンドウで、開きたいレポートを選択できます。また、Report Design アプリケーションに関連付けられている .rst テンプレートファイルをダブルクリックすることもできます。

#### テンプレートの保存

テンプレートを保存するには、メインメニューの [ファイル] -> [保存] を選択するか、ツールバーの「テンプレートの保存」ボタンを押してください。既にドキュメントが保存されている場合は同じファイルが変更されます。新しいテンプレートを編集する場合、テンプレートのフォルダとファイル名を選択する「テンプレートの保存」ダイアログウィンドウが表示されます。[ファイル] -> [名前を付けて保存] は常に「テンプレートの保存」ダイアログを開きます。

#### テンプレートの生成

テンプレートを生成するには、メインメニューの [レポート] -> [プレビュー] を選択するか、ツールバーの「プレビュー」ボタンを押してください。



## テンプレートのプロパティ

プロパティグリッドでテンプレートのプロパティを設定できます。プロパティグリッドにテンプレートのプロパティを表示するには、オブジェクト一覧から **Document** を選択するか、クイックアクセスツールバーの「ドキュメントのプロパティ」を押してください。

## イメージ設定

GraphicsSettings のグループには次のプロパティがあります。

プロパティ	説明
CompositingQuality	合成処理で使用する品質レベルを設定します
InterpolationMode	補間モードを設定します
SmoothingMode	スムージングモードを設定します
TextRenderingHint	テキストの表示モードを設定します

## オブジェクトの表示とグリッド

ShowObjectBounds プロパティは、テンプレートにオブジェクトの境界を表示するかを設定できます。

プロパティ	レポートテンプレート								
<table border="1"> <tr><td colspan="2"><b>Appearance</b></td></tr> <tr><td>GraphicsSettings</td><td>PerpetuumSoft.Reporting.DOM.G</td></tr> <tr><td>MasterReport</td><td></td></tr> <tr><td>ShowObjectBounds</td><td>True</td></tr> </table>	<b>Appearance</b>		GraphicsSettings	PerpetuumSoft.Reporting.DOM.G	MasterReport		ShowObjectBounds	True	
<b>Appearance</b>									
GraphicsSettings	PerpetuumSoft.Reporting.DOM.G								
MasterReport									
ShowObjectBounds	True								
<table border="1"> <tr><td colspan="2"><b>Appearance</b></td></tr> <tr><td>GraphicsSettings</td><td>PerpetuumSoft.Reporting.DOM.G</td></tr> <tr><td>MasterReport</td><td></td></tr> <tr><td>ShowObjectBounds</td><td>False</td></tr> </table>	<b>Appearance</b>		GraphicsSettings	PerpetuumSoft.Reporting.DOM.G	MasterReport		ShowObjectBounds	False	
<b>Appearance</b>									
GraphicsSettings	PerpetuumSoft.Reporting.DOM.G								
MasterReport									
ShowObjectBounds	False								

ShowGrid プロパティは、グリッドを表示するかを定義し、GridStep プロパティはグリッド間隔を設定します。

プロパティ	レポートテンプレート								
<table border="1"> <tr><td colspan="2"><b>Grid Settings</b></td></tr> <tr><td>GridStep</td><td>0,25 cm</td></tr> <tr><td>ShowGrid</td><td>False</td></tr> <tr><td>SnapToGrid</td><td>True</td></tr> </table>	<b>Grid Settings</b>		GridStep	0,25 cm	ShowGrid	False	SnapToGrid	True	
<b>Grid Settings</b>									
GridStep	0,25 cm								
ShowGrid	False								
SnapToGrid	True								
<table border="1"> <tr><td colspan="2"><b>Grid Settings</b></td></tr> <tr><td>GridStep</td><td>0,25 cm</td></tr> <tr><td>ShowGrid</td><td>True</td></tr> <tr><td>SnapToGrid</td><td>True</td></tr> </table>	<b>Grid Settings</b>		GridStep	0,25 cm	ShowGrid	True	SnapToGrid	True	
<b>Grid Settings</b>									
GridStep	0,25 cm								
ShowGrid	True								
SnapToGrid	True								
<table border="1"> <tr><td colspan="2"><b>Grid Settings</b></td></tr> <tr><td>GridStep</td><td>0,5 cm</td></tr> <tr><td>ShowGrid</td><td>True</td></tr> <tr><td>SnapToGrid</td><td>True</td></tr> </table>	<b>Grid Settings</b>		GridStep	0,5 cm	ShowGrid	True	SnapToGrid	True	
<b>Grid Settings</b>									
GridStep	0,5 cm								
ShowGrid	True								
SnapToGrid	True								

SnapToGrid プロパティは、オブジェクトの作成時やオブジェクトのサイズ/位置変更時にオブジェクトをグリッドに合わせるために使用します。

ツールバーのボタンを使用して、これらのプロパティを編集できます。

## 計測単位

MetricUnit プロパティは、テンプレート全体の計測単位の設定に使用します。

<b>Misc</b>	
DataSources	<b>(Collection)</b>
Description	
MetricUnit	Centimeter
Parameters	Inch
Title	Millimeter
<b>Render</b>	
StyleSheet	Pixel
<b>Script</b>	
CommonScript	Point
CompilerVersion	Twip

### テンプレートのデータソース

アプリケーションはデータソースになり得ますが、レポートデザイナーの **Document.DataSources** プロパティ（内蔵されたデータソースのコレクション）を使用して、現在のテンプレートのデータソースを設定できます。このプロパティを設定するには、ドキュメントのデータソースエディタを使用します。

### テンプレートのパラメータ

**Parameters** コレクションを使用して、レポート パラメータを設定できます。

### ページ

レポートテンプレートはページのセットです。各ページには、データを表示するオブジェクトが含まれています。規定の何も入っていないレポートにはページが **1** ページ含まれています。

### ページのプロパティ

プロパティグリッドでページのプロパティを設定できます。プロパティグリッドにページのプロパティを表示するには、オブジェクト一覧で **Page** を選択してください。

### ページサイズ

ページサイズを設定するには、**PaperKind** プロパティを使用します。このプロパティを使用して一般的な用紙サイズ（例：A4（デフォルト）、B5、レターなど）を選択できます。ページの向きを縦または横に設定できます。この場合、**Size** や **TemplateSize** プロパティは選択した用紙のサイズ設定を取得できます。尚、変更することはできません。

<b>Page Settings</b>	
CustomSize	0; 0 cm
Margins	<b>1; 1.2; 1.5; 1 cm</b>
Orientation	Portrait
PaperKind	<b>B4</b>
TemplateSize	25; 35,3 cm

ユーザー定義の用紙サイズを作成したい場合は、**PaperKind** プロパティの値を「Custom」にし、**CustomSize** プロパティの用紙サイズのパラメータを設定してください。

<b>Page Settings</b>	
CustomSize	<b>5; 5 cm</b>
X	<b>5 cm</b>
Y	<b>5 cm</b>
Margins	<b>1; 1.2; 1.5; 1 cm</b>
Orientation	Portrait
PaperKind	<b>Custom</b>
TemplateSize	5; 5 cm

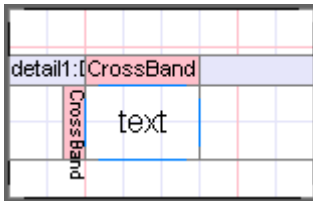
### ページの余白

ページの **Margins** プロパティは、ページの境界からバンド（コンテナ）に貼り付けたオブジェクトまでのインデントを設定します。**Margins** プロパティはページの余白を設定します。テンプレートでは、余白は赤線で表示されます。



ページの Margins プロパティの使用例 :

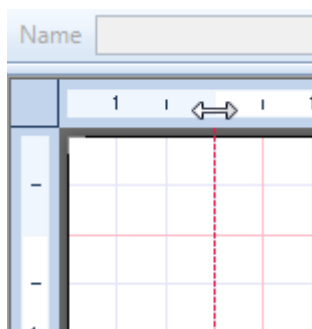
レポートテンプレート :



crossBand1.InstanceCount = 2

プロパティ	レポート																
<table border="1"> <tr><td colspan="2">Page Settings</td></tr> <tr><td>CustomSize</td><td>0; 0 cm</td></tr> <tr><td>Margins</td><td>0.5; 0.5; 1; 1 cm</td></tr> <tr><td>Bottom</td><td>0.5 cm</td></tr> <tr><td>Left</td><td>1 cm</td></tr> <tr><td>Right</td><td>1 cm</td></tr> <tr><td>Top</td><td>0.5 cm</td></tr> <tr><td>Orientation</td><td>Portrait</td></tr> </table>	Page Settings		CustomSize	0; 0 cm	Margins	0.5; 0.5; 1; 1 cm	Bottom	0.5 cm	Left	1 cm	Right	1 cm	Top	0.5 cm	Orientation	Portrait	
Page Settings																	
CustomSize	0; 0 cm																
Margins	0.5; 0.5; 1; 1 cm																
Bottom	0.5 cm																
Left	1 cm																
Right	1 cm																
Top	0.5 cm																
Orientation	Portrait																
<table border="1"> <tr><td colspan="2">Page Settings</td></tr> <tr><td>CustomSize</td><td>0; 0 cm</td></tr> <tr><td>Margins</td><td>0; 0; 1; 1 cm</td></tr> <tr><td>Bottom</td><td>0 cm</td></tr> <tr><td>Left</td><td>1 cm</td></tr> <tr><td>Right</td><td>1 cm</td></tr> <tr><td>Top</td><td>0 cm</td></tr> <tr><td>Orientation</td><td>Portrait</td></tr> </table>	Page Settings		CustomSize	0; 0 cm	Margins	0; 0; 1; 1 cm	Bottom	0 cm	Left	1 cm	Right	1 cm	Top	0 cm	Orientation	Portrait	
Page Settings																	
CustomSize	0; 0 cm																
Margins	0; 0; 1; 1 cm																
Bottom	0 cm																
Left	1 cm																
Right	1 cm																
Top	0 cm																
Orientation	Portrait																
<table border="1"> <tr><td colspan="2">Page Settings</td></tr> <tr><td>CustomSize</td><td>0; 0 cm</td></tr> <tr><td>Margins</td><td>0; 0; 0; 0 cm</td></tr> <tr><td>Bottom</td><td>0 cm</td></tr> <tr><td>Left</td><td>0 cm</td></tr> <tr><td>Right</td><td>0 cm</td></tr> <tr><td>Top</td><td>0 cm</td></tr> <tr><td>Orientation</td><td>Portrait</td></tr> </table>	Page Settings		CustomSize	0; 0 cm	Margins	0; 0; 0; 0 cm	Bottom	0 cm	Left	0 cm	Right	0 cm	Top	0 cm	Orientation	Portrait	
Page Settings																	
CustomSize	0; 0 cm																
Margins	0; 0; 0; 0 cm																
Bottom	0 cm																
Left	0 cm																
Right	0 cm																
Top	0 cm																
Orientation	Portrait																

ルーラを使用してページの Margins を変更できます。



### ページスタイル

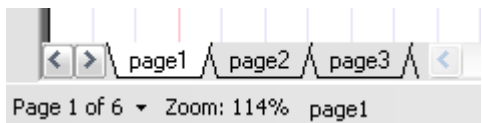
**StyleName** プロパティは、ページの全オブジェクトに統一したスタイルを設定できます。スタイルシートエディタでスタイルを作成し、そのスタイルを **StyleName** プロパティにパラメータとして選択すると、ページに追加されたすべてのオブジェクトやバンドに規定値としてそのスタイルが設定されます。編集する時に個別にスタイルを変えなければ、ページの全オブジェクトにそのスタイルが適用されます。

### マルチページ レポート

レポートに必要な分のページを追加できます (例: 表紙付きのレポート)。用意されたレポートは、ページのテンプレートに従って作成されたレポートの組み合わせです。レポート部分はテンプレートのページ順に表示されます。



レポートにページを追加するには、[挿入] タブの「新規ページ」ボタンをクリックしてください。ページを削除すると、そのページに含まれているオブジェクトもすべて削除されます。レポートデザイナーの左下端のタブでページ間を移動できます。

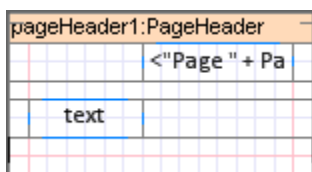


### ページの見出し

PageHeader と PageFooter は、それぞれヘッダー/フッターを設定するために使用します。バンド内容は、ページのバンドの位置に関係なくページの上部/下部に表示されます。Mode プロパティは見出しの生成モード（全ページ、偶数/奇数ページ）を設定します。

ページの見出しの使用例：

レポートテンプレート：



textBoxPageHeader.Value = "Page " + PageNumber

Mode プロパティ	レポート										
<table border="1"> <tr><td colspan="2"><b>Render</b></td></tr> <tr><td>CanGrow</td><td>False</td></tr> <tr><td>CanShrink</td><td>False</td></tr> <tr><td>Mode</td><td>AllPages</td></tr> <tr><td>Visible</td><td>True</td></tr> </table>	<b>Render</b>		CanGrow	False	CanShrink	False	Mode	AllPages	Visible	True	
<b>Render</b>											
CanGrow	False										
CanShrink	False										
Mode	AllPages										
Visible	True										
<table border="1"> <tr><td colspan="2"><b>Render</b></td></tr> <tr><td>CanGrow</td><td>False</td></tr> <tr><td>CanShrink</td><td>False</td></tr> <tr><td>Mode</td><td>OddPages</td></tr> <tr><td>Visible</td><td>True</td></tr> </table>	<b>Render</b>		CanGrow	False	CanShrink	False	Mode	OddPages	Visible	True	
<b>Render</b>											
CanGrow	False										
CanShrink	False										
Mode	OddPages										
Visible	True										
<table border="1"> <tr><td colspan="2"><b>Render</b></td></tr> <tr><td>CanGrow</td><td>False</td></tr> <tr><td>CanShrink</td><td>False</td></tr> <tr><td>Mode</td><td>EvenPages</td></tr> <tr><td>Visible</td><td>True</td></tr> </table>	<b>Render</b>		CanGrow	False	CanShrink	False	Mode	EvenPages	Visible	True	
<b>Render</b>											
CanGrow	False										
CanShrink	False										
Mode	EvenPages										
Visible	True										

## ページ番号

**PageNumber** プロパティは、ページ番号を取得するために使用します。**PageCount** プロパティはレポートのページ総数を格納します。最初のパス時はページ総数はわからないので、**PageCount** プロパティは **PageNumber** プロパティと同じになります。**PageCount** プロパティがページ総数を取得するには、Document オブジェクトの **DoublePass** プロパティを "True" に設定してください。

ヘッダーにページ番号を表示する例：

```
dataBand1.InstanceCount = 3;
```

```
textBoxHeader.Value = "Page " + PageNumber.ToString() + " of " + PageCount.ToString();
```

```
textBoxDetail.Value = "Line Number " + dataBand1.LineNumber().ToString();
```

レポートテンプレート	レポート
pageHeader1:PageHeader <"Page " + PageNumber.ToString() + " of " + PageCount.ToString();	Page 1 of 2
dataBand1:DataBand detail1:Detail <"Line Number " + dataBand1.LineNumber().ToString();	Line Number 1 Line Number 2
end of dataBand1	Page 2 of 2 Line Number 3

## ページの重ね合わせ

**PageOverlay** は、背景にデータを表示しなければならない場合に使用します。オブジェクトの位置は、ページの **PageOverlay** の位置に関係なく、バンドの位置に従って計算されます。このバンドは他のバンドの位置には影響を及ぼしません。**PageOverlay** オブジェクトの上に他のバンドの内容が表示されます。

レポートテンプレート	レポート
pageOverlay1:PageOverlay text	text
pageOverlay1:PageOverlay text	text
pageOverlay1:PageOverlay text	text

**Mode** プロパティは、重ね合わせモード（全ページ、奇数/偶数ページ）を設定します。

## オブジェクトの操作

### オブジェクトの追加

オブジェクトを追加するには「挿入」タブを使用します。次に、マウスポインタをページに移動させ左クリックしてオブジェクトを追加します。あるバンドをクリックすると、プログラムはそのバンドにオブジェクトを挿入できるか確認します（詳細は利用可能なバンドコンテナを参照してください）。バンドにオブジェクトを挿入できる場合はテンプレートのバンドに追加されます。そうでなければエラーメッセージが表示されます。

出力したい項目をツールウィンドウのデータソースタブのデータツリーからテンプレートにドラッグアンドドロップすることにより、**TextBox** オブジェクトにデータソースのデータを出力できます。その場合、**TextBox.Value** プロパティは自動的に設定されます。

### オブジェクトの削除

オブジェクトを削除する方法はいくつかあります。オブジェクトを左クリック（またはオブジェクト一覧またはツールウィンドウのドキュメントツリーからオブジェクトを選択）してキーボードの「Delete」ボタンを押します。バンドを削除すると、その中に格納されているオブジェクトもすべて削除されます。

### オブジェクトのプロパティ

オブジェクトのプロパティは、プロパティ グリッドを使用して設定できます。プロパティグリッドにプロパティを表示するには、テンプレートのオブジェクトを選択します。複数のオブジェクトを選択すると、プロパティグリッドに共通のプロパティが表示されます。

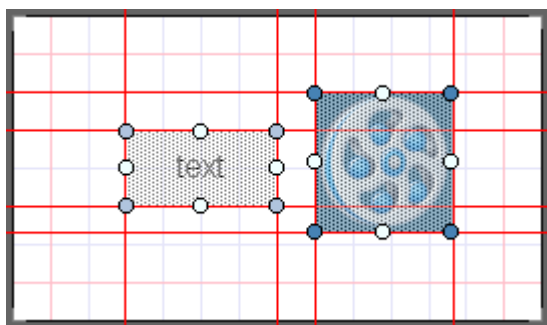
### ドキュメントツリー

ドキュメントツリーには、ドキュメントの全オブジェクト一覧が含まれており、その構造を表します。ドキュメント ツリーを使用して、オブジェクトを選択/削除したり、（オブジェクトを別のバンドにドラッグして）階層を変更したり、オブジェクトの順番を変更することができます。

### オブジェクトの位置

テンプレートのオブジェクトの位置は **Size** プロパティで設定されます。2つのオブジェクトが重なっている場合は前面のオブジェクトが表示されます。[編集/配列] メニューをクリックするか、ツールバーのボタンを使用してオブジェクトの順番を変更します。複数のオブジェクトをまとめて揃えるには、整列エディタを使用します。

同じコンテナに含まれている複数のオブジェクトを **Shift** キーを押しながらマウスポインタで選択することにより一斉に位置を変更することができます。



キーボードのナビゲーションボタン（↑←↓→）を使って、オブジェクトの位置を変更できます。Ctrl キーを押したままナビゲーションボタンを使用すると、グリッドから外れた位置にオブジェクトを移動できます。

### オブジェクトのプロパティ

オブジェクトのプロパティは、レポート生成時のオブジェクトの見た目や動作を定義します。オブジェクトのプロパティはプロパティグリッドで設定できます。プロパティグリッドにオブジェクトのプロパティを表示するには、テンプレート内のオブジェクトを左マウスボタンで選択するか、オブジェクト一覧から選択する必要があります。プロパティの規定値は普通のフォントで表示され、編集したプロパティ値は太字で表示されます。複数のオブジェクトを選択するには **Shift** ボタンを押しながら選択します。その場合、プロパティグリッドに









は選択したオブジェクトの共通のプロパティが表示されます。オブジェクトのプロパティはすべて次のようなグループに分けられます。

グループ	説明
表示	オブジェクトの表示や見た目を定義するプロパティ
データ	データ処理を定義するプロパティ
デザイン	レポートデザイナーでテンプレートを編集する時に、オブジェクトの表示を定義するプロパティ
レイアウト	オブジェクトの位置やサイズを定義するプロパティ
その他	レポートで作成されたオブジェクトのプロパティ
Navigation (ナビゲーション)	レポートのナビゲーションを作成するプロパティ
PageSetting (ページ設定)	ページのプロパティ
Render (生成)	レポート生成時のオブジェクトの動作命令を定義するプロパティ
Script (スクリプト)	クリプト - オブジェクトの処理

### Angle プロパティ (表示)

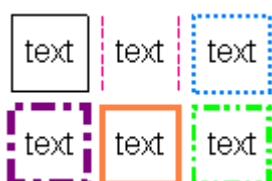
その領域に格納されているオブジェクトの回転角度を設定するために使用します。設定可能な値は 0、90、180、270 度です。Picture と TextBox オブジェクトに回転角度を設定した例：

プロパティ	レポート
<input type="checkbox"/> <b>Appearance</b> Angle 0 Border <b>Border</b> Fill [None] <input checked="" type="checkbox"/> Margins 0; 0; 0; 0 cm StyleName ToolTip	 
<input type="checkbox"/> <b>Appearance</b> Angle <b>90</b> Border <b>Border</b> Fill [None] <input checked="" type="checkbox"/> Margins 0; 0; 0; 0 cm StyleName ToolTip	 
<input type="checkbox"/> <b>Appearance</b> Angle <b>180</b> Border <b>Border</b> Fill [None] <input checked="" type="checkbox"/> Margins 0; 0; 0; 0 cm StyleName ToolTip	 

### Border プロパティ (表示)

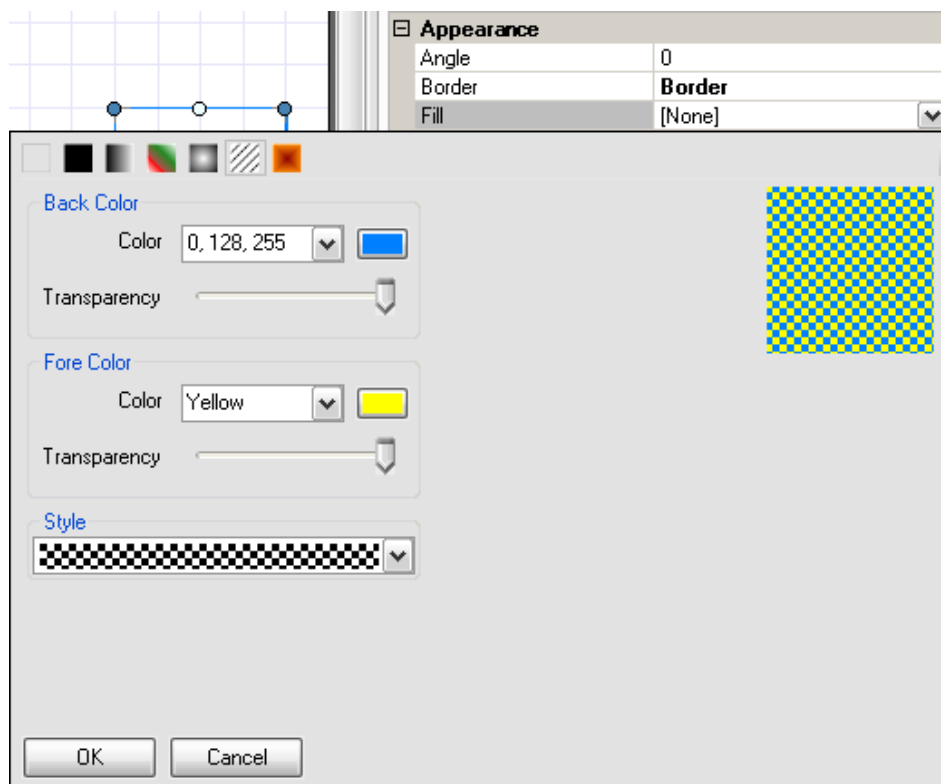
オブジェクトの枠の表示と線の描画スタイル (枠エディタ) - 線種、幅、色を設定します。

TextBox オブジェクトの枠の例：

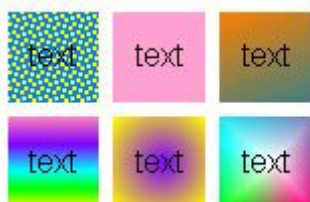


## Fill プロパティ (表示)

オブジェクトの領域を塗りつぶす色やスタイルを設定します。このプロパティを変更するには、塗りつぶしエディタを使用して塗りつぶし型、色、透明度やその他のパラメータを設定できます。

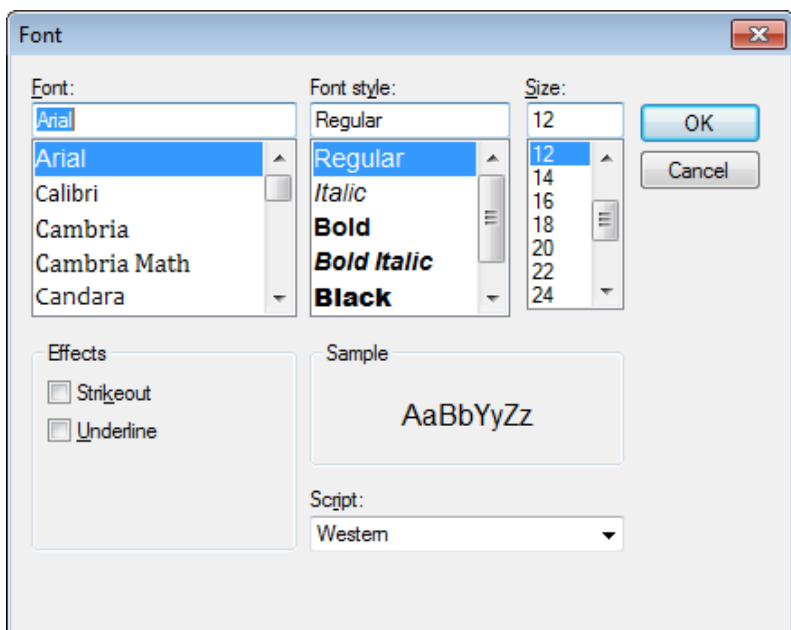


TextBox オブジェクトの塗りつぶしの使用例 :

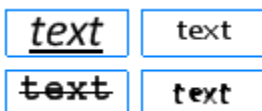


## Font プロパティ (表示)

オブジェクトのフォント型を設定できます。プロパティを変更するためのフォントエディタがあります。それを使用して、フォント、フォントスタイル、サイズを設定できます。



TextBox オブジェクトのフォントの使用例 :



注意 : テキストを **AdvancedText** で出力し、フォントを書式付きテキストエディタで設定した場合、Font プロパティは無視されます。

### Margins プロパティ (表示)

オブジェクトの余白を設定します。

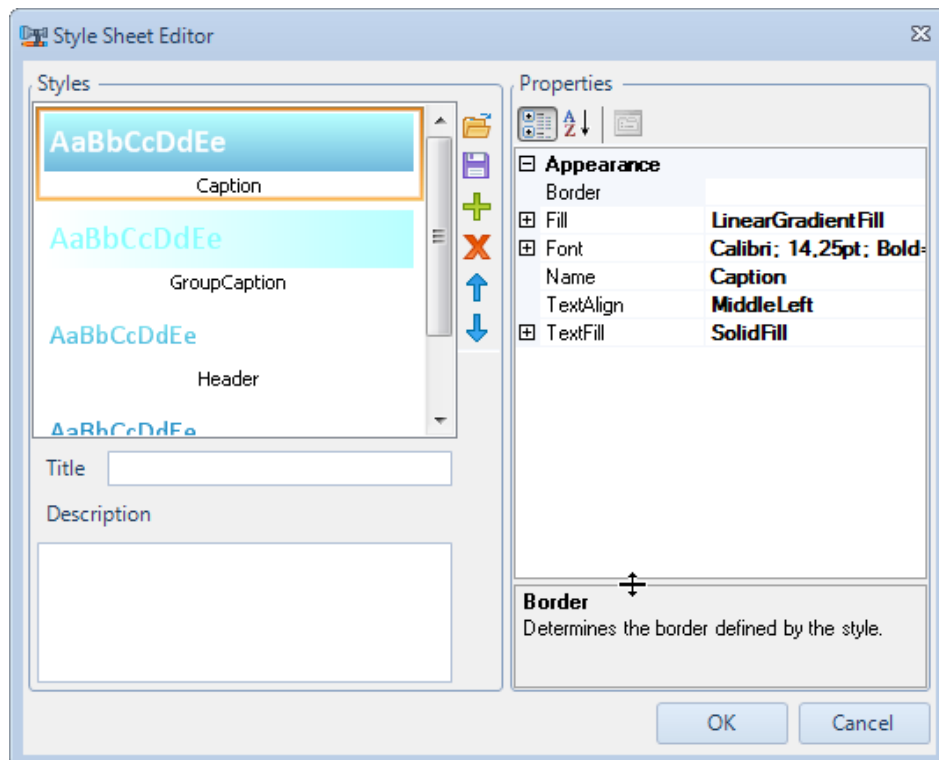
Picture オブジェクトの余白の変更例 :

プロパティ	レポート																						
<table border="1"> <tr><td colspan="2"><b>Appearance</b></td></tr> <tr><td>Angle</td><td>0</td></tr> <tr><td>Border</td><td></td></tr> <tr><td>Fill</td><td>[None]</td></tr> <tr><td>Image</td><td> System.Drawing.Bitmap</td></tr> <tr><td>ImageAlign</td><td>TopLeft</td></tr> <tr><td colspan="2"><b>Margins</b></td></tr> <tr><td>Bottom</td><td>0 cm</td></tr> <tr><td>Left</td><td>0 cm</td></tr> <tr><td>Right</td><td>0 cm</td></tr> <tr><td>Top</td><td>0 cm</td></tr> </table>	<b>Appearance</b>		Angle	0	Border		Fill	[None]	Image	System.Drawing.Bitmap	ImageAlign	TopLeft	<b>Margins</b>		Bottom	0 cm	Left	0 cm	Right	0 cm	Top	0 cm	
<b>Appearance</b>																							
Angle	0																						
Border																							
Fill	[None]																						
Image	System.Drawing.Bitmap																						
ImageAlign	TopLeft																						
<b>Margins</b>																							
Bottom	0 cm																						
Left	0 cm																						
Right	0 cm																						
Top	0 cm																						
<table border="1"> <tr><td colspan="2"><b>Appearance</b></td></tr> <tr><td>Angle</td><td>0</td></tr> <tr><td>Border</td><td></td></tr> <tr><td>Fill</td><td>[None]</td></tr> <tr><td>Image</td><td> System.Drawing.Bitmap</td></tr> <tr><td>ImageAlign</td><td>TopLeft</td></tr> <tr><td colspan="2"><b>Margins</b></td></tr> <tr><td>Bottom</td><td><b>0,2 cm</b></td></tr> <tr><td>Left</td><td><b>0,2 cm</b></td></tr> <tr><td>Right</td><td><b>0,2 cm</b></td></tr> <tr><td>Top</td><td><b>0,2 cm</b></td></tr> </table>	<b>Appearance</b>		Angle	0	Border		Fill	[None]	Image	System.Drawing.Bitmap	ImageAlign	TopLeft	<b>Margins</b>		Bottom	<b>0,2 cm</b>	Left	<b>0,2 cm</b>	Right	<b>0,2 cm</b>	Top	<b>0,2 cm</b>	
<b>Appearance</b>																							
Angle	0																						
Border																							
Fill	[None]																						
Image	System.Drawing.Bitmap																						
ImageAlign	TopLeft																						
<b>Margins</b>																							
Bottom	<b>0,2 cm</b>																						
Left	<b>0,2 cm</b>																						
Right	<b>0,2 cm</b>																						
Top	<b>0,2 cm</b>																						

## StyleName プロパティ (表示)

1 つまたは複数のオブジェクトに既存のスタイルを適用して、オブジェクトのスタイルをすばやく編集します。

「ホーム」タブの「スタイルの編集」ボタンでスタイルシートエディタを開きます。



スタイルを開いたり保存するには、「開く」や「保存」ボタンを使用します。ファイルは XML 形式で、ファイル名の拡張子は .rss になります。「スタイルの追加」や「スタイルの削除」ボタンは、スタイルを追加したり削除します。

スタイルの次のプロパティを使用できます。

プロパティ	説明
Border	オブジェクトの枠
Font	オブジェクトのテキストスタイル
Name	スタイル名
TextAlign	オブジェクトのテキストの位置合わせ
TextFill	オブジェクトのテキストの塗りつぶし

オブジェクトにこのスタイルを適用するには、必要とするスタイル名を StyleName プロパティのパラメータとして設定してください。

TextBox や Shape オブジェクトの StyleName プロパティの使用例：

レポートテンプレート	作成したスタイル														
	<table border="1"> <thead> <tr> <th colspan="2">Appearance</th> </tr> </thead> <tbody> <tr> <td>Border</td> <td>Border</td> </tr> <tr> <td>Fill</td> <td>LinearGradientFill</td> </tr> <tr> <td>Font</td> <td>Calibri; 20,25pt; Bold</td> </tr> <tr> <td>Name</td> <td>New Style</td> </tr> <tr> <td>TextAlign</td> <td>MiddleCenter</td> </tr> <tr> <td>TextFill</td> <td>LinearGradientFill</td> </tr> </tbody> </table>	Appearance		Border	Border	Fill	LinearGradientFill	Font	Calibri; 20,25pt; Bold	Name	New Style	TextAlign	MiddleCenter	TextFill	LinearGradientFill
Appearance															
Border	Border														
Fill	LinearGradientFill														
Font	Calibri; 20,25pt; Bold														
Name	New Style														
TextAlign	MiddleCenter														
TextFill	LinearGradientFill														
プロパティ															

<div style="border: 1px solid gray; padding: 2px;"> <p><b>Appearance</b></p> <p>Fill [None]</p> <p>StyleName <span style="float: right;">▼</span></p> <p>ToolTip</p> </div>	
<div style="border: 1px solid gray; padding: 2px;"> <p><b>Appearance</b></p> <p>Fill LinearGradientFill</p> <p>StyleName <b>New Style</b> <span style="float: right;">▼</span></p> <p>ToolTip</p> </div>	

注意 1：スタイルをコンテナに適用すると、含まれているオブジェクトすべてに適用されます。オブジェクトのプロパティ値が規定値（つまり、手動で変更していない場合）の場合のみ、そのスタイルが持っているのと同じ値を取得します。規定値ではない値は太字で表示されます。

注意 2：AdvancedText で出力されるテキストとフォントが書式付きテキストエディタで設定された場合、Font プロパティは無効となります。

### TextAlign プロパティ (表示)

オブジェクトの領域に応じてテキストを配置します。

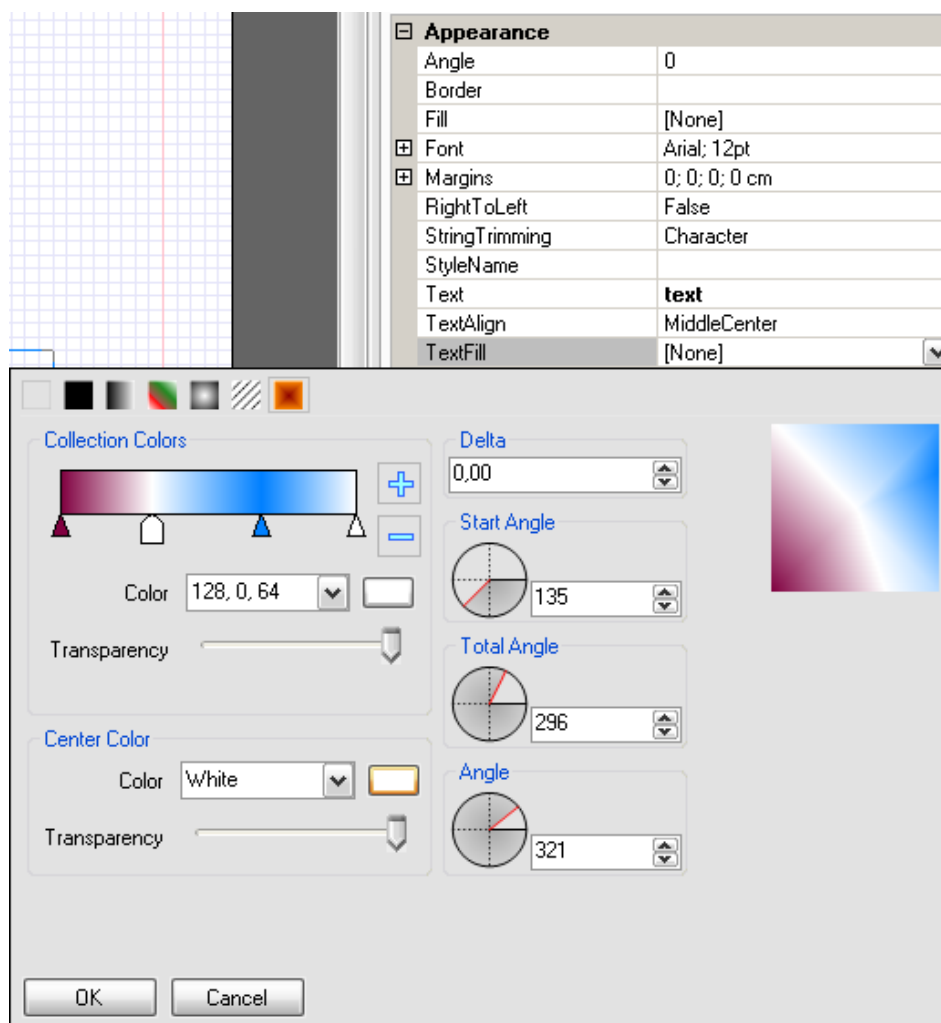
TextBox オブジェクトの配置例：

プロパティ	レポート												
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>StyleName</td><td></td></tr> <tr><td><b>Text</b></td><td><b>text</b></td></tr> <tr><td><b>TextAlign</b></td><td><b>TopLeft</b> ▼</td></tr> <tr><td>TextFill</td><td>[None]</td></tr> <tr><td>TextFormat</td><td></td></tr> <tr><td>ToolTip</td><td></td></tr> </table>	StyleName		<b>Text</b>	<b>text</b>	<b>TextAlign</b>	<b>TopLeft</b> ▼	TextFill	[None]	TextFormat		ToolTip		
StyleName													
<b>Text</b>	<b>text</b>												
<b>TextAlign</b>	<b>TopLeft</b> ▼												
TextFill	[None]												
TextFormat													
ToolTip													
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>StyleName</td><td></td></tr> <tr><td><b>Text</b></td><td><b>text</b></td></tr> <tr><td><b>TextAlign</b></td><td><b>MiddleCenter</b> ▼</td></tr> <tr><td>TextFill</td><td>[None]</td></tr> <tr><td>TextFormat</td><td></td></tr> <tr><td>ToolTip</td><td></td></tr> </table>	StyleName		<b>Text</b>	<b>text</b>	<b>TextAlign</b>	<b>MiddleCenter</b> ▼	TextFill	[None]	TextFormat		ToolTip		
StyleName													
<b>Text</b>	<b>text</b>												
<b>TextAlign</b>	<b>MiddleCenter</b> ▼												
TextFill	[None]												
TextFormat													
ToolTip													
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>StyleName</td><td></td></tr> <tr><td><b>Text</b></td><td><b>text</b></td></tr> <tr><td><b>TextAlign</b></td><td><b>BottomRight</b></td></tr> <tr><td>TextFill</td><td>[None]</td></tr> <tr><td>TextFormat</td><td></td></tr> <tr><td>ToolTip</td><td></td></tr> </table>	StyleName		<b>Text</b>	<b>text</b>	<b>TextAlign</b>	<b>BottomRight</b>	TextFill	[None]	TextFormat		ToolTip		
StyleName													
<b>Text</b>	<b>text</b>												
<b>TextAlign</b>	<b>BottomRight</b>												
TextFill	[None]												
TextFormat													
ToolTip													

### TextFill プロパティ (表示)

テキストの塗りつぶしを設定します。

塗りつぶしエディタで塗りつぶしのプロパティを設定します。塗りつぶし型、色、透明度、その他のパラメータを取得できます。



TextBox オブジェクトのテキストの塗りつぶし例 :

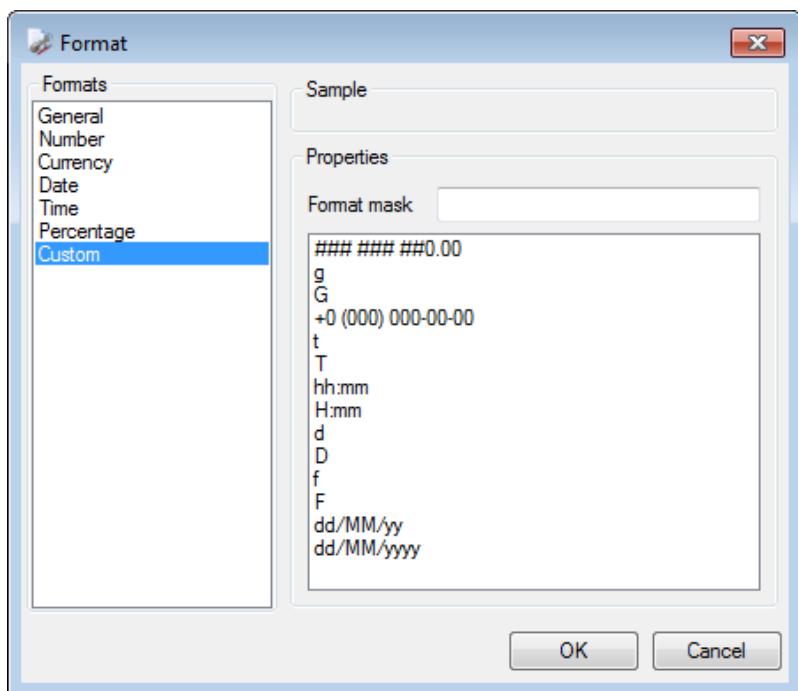


注意 : AdvancedText に出力するテキストやフォントが書式付きテキストエディタで設定された場合、Font プロパティは無視されます。

### TextFormat プロパティ (表示)

テキストを設定した形式 (数値、通貨、日付、時刻、パーセント) に変更します。ユーザー定義の形式も作成できます。

フォーマットエディタのテキスト形式をユーザー定義に設定 :



Textbox オブジェクトに使用したテキスト形式の例 :

プロパティ		レポート
StyleName		
Text	34.5	£34.50
TextAlign	MiddleCenter	
TextFill	[None]	
TextFormat	Currency	
ToolTip		
StyleName		
Text	02.17.2010	17 February 2010
TextAlign	MiddleCenter	
TextFill	[None]	
TextFormat	Date	
ToolTip		
StyleName		
Text	123456	123,456.00
TextAlign	MiddleCenter	
TextFill	[None]	
TextFormat	Number	
ToolTip		

### ToolTip プロパティ (表示)

オブジェクトにマウスポインタを置いた時に表示されるヒントを作成します。

Shape オブジェクトでのツールチップの使用例 :

プロパティ		レポート
[-] Appearance		
[+] Fill	ConicalFill	
[+] Line	0 Solid Black	
[+] Shadow	Shadow	
[+] ShapeStyle	StarShape	
StyleName		
ToolTip	StarShape	



## Tag プロパティ (データ)

ユーザーが作成したオブジェクトに追加情報を持たせることができます。

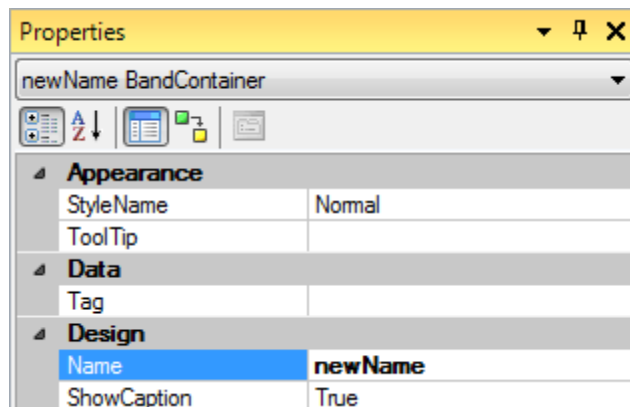
タグの使用例 :



## Name プロパティ (デザイン)

オブジェクトの名前を設定します。すべてのオブジェクト名は一意でなければなりません。

BandContainer に名前を設定 :



## ShowBottom プロパティ (デザイン)

デザイナーでバンド下部の境界線を表示/非表示に設定できます。

DataBand の ShowBottom プロパティの使用例 :

プロパティ	レポート								
<table border="1"> <tr> <td colspan="2">Design</td> </tr> <tr> <td>Name</td> <td>dataBand2</td> </tr> <tr> <td>ShowBottom</td> <td>True</td> </tr> <tr> <td>ShowCaption</td> <td>True</td> </tr> </table>	Design		Name	dataBand2	ShowBottom	True	ShowCaption	True	
Design									
Name	dataBand2								
ShowBottom	True								
ShowCaption	True								
<table border="1"> <tr> <td colspan="2">Design</td> </tr> <tr> <td>Name</td> <td>dataBand2</td> </tr> <tr> <td>ShowBottom</td> <td>False</td> </tr> <tr> <td>ShowCaption</td> <td>True</td> </tr> </table>	Design		Name	dataBand2	ShowBottom	False	ShowCaption	True	
Design									
Name	dataBand2								
ShowBottom	False								
ShowCaption	True								

## ShowCaption プロパティ (デザイン)

デザイナーモードでのバンドのヘッダー部分の表示/非表示を設定します。

DataBand の ShowCaption プロパティの使用例 :

プロパティ	レポート								
<table border="1"> <tr> <td colspan="2">Design</td> </tr> <tr> <td>Name</td> <td>dataBand2</td> </tr> <tr> <td>ShowBottom</td> <td>True</td> </tr> <tr> <td>ShowCaption</td> <td>True</td> </tr> </table>	Design		Name	dataBand2	ShowBottom	True	ShowCaption	True	
Design									
Name	dataBand2								
ShowBottom	True								
ShowCaption	True								

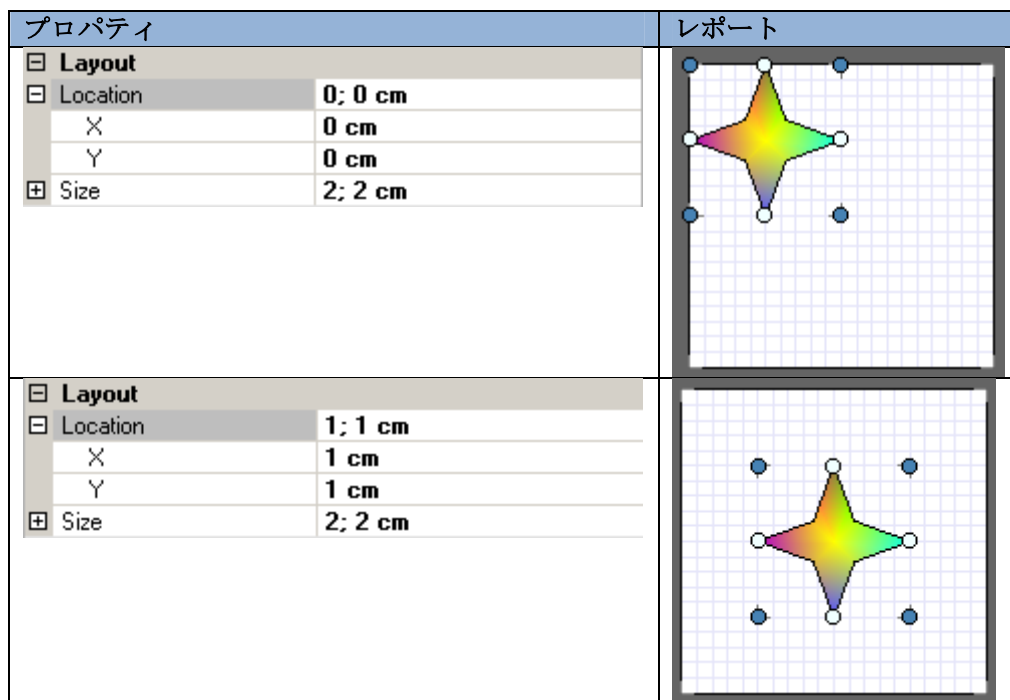


### Location プロパティ (レイアウト)

コンテナに対するオブジェクトの左上端の位置を設定します。

オブジェクトの位置は、マウスポインタまたはプロパティグリッドで変更できます。

Shape オブジェクトの位置変更例：



注意 1：X 座標は 0 と等しいので変更できません。

注意 2：「グリッドに合わせる」オプションは、マウスポインタで位置を変更する時にオブジェクトをグリッドに合わせるかどうかを定義します。ツールバーの下図のボタンをクリックしてこのオプションを切り替えてください。



### Size プロパティ (レイアウト)

オブジェクトのサイズを設定します。

このプロパティは、オブジェクトの枠をマウスポインタまたはプロパティグリッドから移動する時にサイズを変更します。

Shape オブジェクトのサイズ変更例 :

プロパティ	レポート										
<table border="1"> <tr><td colspan="2">Layout</td></tr> <tr><td>Location</td><td>1; 1 cm</td></tr> <tr><td>Size</td><td>2; 2 cm</td></tr> <tr><td>  X</td><td>2 cm</td></tr> <tr><td>  Y</td><td>2 cm</td></tr> </table>	Layout		Location	1; 1 cm	Size	2; 2 cm	X	2 cm	Y	2 cm	
Layout											
Location	1; 1 cm										
Size	2; 2 cm										
X	2 cm										
Y	2 cm										
<table border="1"> <tr><td colspan="2">Layout</td></tr> <tr><td>Location</td><td>1; 1 cm</td></tr> <tr><td>Size</td><td>1; 1 cm</td></tr> <tr><td>  X</td><td>1 cm</td></tr> <tr><td>  Y</td><td>1 cm</td></tr> </table>	Layout		Location	1; 1 cm	Size	1; 1 cm	X	1 cm	Y	1 cm	
Layout											
Location	1; 1 cm										
Size	1; 1 cm										
X	1 cm										
Y	1 cm										

注意 1 : バンドの X 座標が 0 と等しいので、変わりません。

注意 2 : 「グリッドに合わせる」は、オブジェクトの位置を変更する時にグリッドに合わせるか定義します。このオプションの切り替えには「グリッドに合わせる」ボタンを使用してください。

### CanBreak プロパティ (Render)

オブジェクトがページサイズを超えた場合にオブジェクトの内容をページ分割します。

Detail と AdvancedText オブジェクトの CanBreak プロパティの例 :

プロパティ	レポート												
<table border="1"> <tr><td colspan="2">Render</td></tr> <tr><td>CanBreak</td><td>False</td></tr> <tr><td>CanGrow</td><td>True</td></tr> <tr><td>CanShrink</td><td>False</td></tr> <tr><td>GrowToBottom</td><td>False</td></tr> <tr><td>Visible</td><td>True</td></tr> </table>	Render		CanBreak	False	CanGrow	True	CanShrink	False	GrowToBottom	False	Visible	True	<p>dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad</p>
Render													
CanBreak	False												
CanGrow	True												
CanShrink	False												
GrowToBottom	False												
Visible	True												
<table border="1"> <tr><td colspan="2">Render</td></tr> <tr><td>CanBreak</td><td>True</td></tr> <tr><td>CanGrow</td><td>True</td></tr> <tr><td>CanShrink</td><td>False</td></tr> <tr><td>GrowToBottom</td><td>False</td></tr> <tr><td>Visible</td><td>True</td></tr> </table>	Render		CanBreak	True	CanGrow	True	CanShrink	False	GrowToBottom	False	Visible	True	<p>dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt erat volutpat. Ut wisi enim ad</p> <p>minim veniam, quis nostrud exercitation ullamcorper suscipit lobortis nisi ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in</p>
Render													
CanBreak	True												
CanGrow	True												
CanShrink	False												
GrowToBottom	False												
Visible	True												

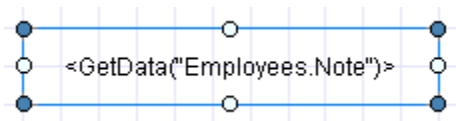
注意 : 別のセクションに貼り付けたオブジェクトの CanBreak プロパティを設定する場合、その内容をページ分割するには、コンテナバンドに CanBreak プロパティを設定しなければなりません。

### CanGrow プロパティ (Render)

コンテンツがテンプレートのオブジェクトに指定したサイズに収まりきらない場合にオブジェクトの高さを大きくします。

## TextBox オブジェクトの CanGrow プロパティの使用例

レポートテンプレート :



プロパティ	レポート										
<table border="1"> <tr><td colspan="2"><b>Render</b></td></tr> <tr><td>CanGrow</td><td>False</td></tr> <tr><td>CanShrink</td><td>False</td></tr> <tr><td>GrowToBottom</td><td>False</td></tr> <tr><td>Visible</td><td>True</td></tr> </table>	<b>Render</b>		CanGrow	False	CanShrink	False	GrowToBottom	False	Visible	True	<div style="border: 1px solid blue; padding: 5px;">Education includes a BA in psychology from Colorado State Uni</div>
<b>Render</b>											
CanGrow	False										
CanShrink	False										
GrowToBottom	False										
Visible	True										
<table border="1"> <tr><td colspan="2"><b>Render</b></td></tr> <tr><td>CanGrow</td><td><b>True</b></td></tr> <tr><td>CanShrink</td><td>False</td></tr> <tr><td>GrowToBottom</td><td>False</td></tr> <tr><td>Visible</td><td>True</td></tr> </table>	<b>Render</b>		CanGrow	<b>True</b>	CanShrink	False	GrowToBottom	False	Visible	True	<div style="border: 1px solid blue; padding: 5px;">Education includes a BA in psychology from Colorado State University in 1970. She also completed The Art of the Cold Call. Nancy is a member of Toastmasters International.</div>
<b>Render</b>											
CanGrow	<b>True</b>										
CanShrink	False										
GrowToBottom	False										
Visible	True										

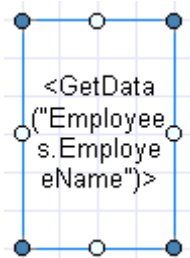
注意：別のバンド内のオブジェクトの **CanGrow** プロパティを設定し、そのオブジェクトの高さに応じてコンテナバンドの高さを変えるには、コンテナバンドの **CanGrow** プロパティも設定しなければなりません。

## CanShrink プロパティ (Render)

レポート生成時に挿入されたコンテンツがコンテナのサイズよりも小さい場合にオブジェクトの高さを小さくします。

## TextBox オブジェクトの CanShrink プロパティの使用例

レポートテンプレート :



プロパティ	レポート										
<table border="1"> <tr><td colspan="2"><b>Render</b></td></tr> <tr><td>CanGrow</td><td>False</td></tr> <tr><td>CanShrink</td><td>False</td></tr> <tr><td>GrowToBottom</td><td>False</td></tr> <tr><td>Visible</td><td>True</td></tr> </table>	<b>Render</b>		CanGrow	False	CanShrink	False	GrowToBottom	False	Visible	True	<div style="border: 1px solid blue; padding: 5px; text-align: center;">Nancy Davolio</div>
<b>Render</b>											
CanGrow	False										
CanShrink	False										
GrowToBottom	False										
Visible	True										
<table border="1"> <tr><td colspan="2"><b>Render</b></td></tr> <tr><td>CanGrow</td><td>False</td></tr> <tr><td>CanShrink</td><td><b>True</b></td></tr> <tr><td>GrowToBottom</td><td>False</td></tr> <tr><td>Visible</td><td>True</td></tr> </table>	<b>Render</b>		CanGrow	False	CanShrink	<b>True</b>	GrowToBottom	False	Visible	True	<div style="border: 1px solid blue; padding: 5px; text-align: center;">Nancy Davolio</div>
<b>Render</b>											
CanGrow	False										
CanShrink	<b>True</b>										
GrowToBottom	False										
Visible	True										

注意：別のバンド内のオブジェクトの **CanShrink** プロパティを設定し、そのオブジェクトの高さに応じてコンテナの高さを変更するには、コンテナバンドの **CanShrink** プロパティも設定しなければなりません。

### MinSpaceAfter プロパティ (Render)

バンドの下枠からページの下端領域までの最小間隔を設定します。

Detail の MinSpaceAfter を使用したリストの表示例 :

レポートテンプレート :

```
dataBand1:DataBand DataSource = Em
detail1:Detail
  <dataBandL
    ("EmployeeName")>
end of dataBand1
```

プロパティ	レポート								
<table border="1"> <tr><td>CanGrow</td><td>False</td></tr> <tr><td>CanShrink</td><td>False</td></tr> <tr><td><b>MinSpaceAfter</b></td><td><b>0,75 cm</b></td></tr> <tr><td>NewColumnAfter</td><td>False</td></tr> </table>	CanGrow	False	CanShrink	False	<b>MinSpaceAfter</b>	<b>0,75 cm</b>	NewColumnAfter	False	
CanGrow	False								
CanShrink	False								
<b>MinSpaceAfter</b>	<b>0,75 cm</b>								
NewColumnAfter	False								

### Mode プロパティ (Render)

見出しや重ね合わせに設定でき、オブジェクトを奇数、偶数、全ページに表示するかを定義できます。

ページヘッダーの Mode プロパティの使用例

レポートテンプレート :

```
pageHeader1:PageHeader
  <PageNumber>
```

プロパティ	レポート										
<table border="1"> <tr><td colspan="2"><b>Render</b></td></tr> <tr><td>CanGrow</td><td>False</td></tr> <tr><td>CanShrink</td><td>False</td></tr> <tr><td><b>Mode</b></td><td>AllPages</td></tr> <tr><td>Visible</td><td>True</td></tr> </table>	<b>Render</b>		CanGrow	False	CanShrink	False	<b>Mode</b>	AllPages	Visible	True	
<b>Render</b>											
CanGrow	False										
CanShrink	False										
<b>Mode</b>	AllPages										
Visible	True										
<table border="1"> <tr><td colspan="2"><b>Render</b></td></tr> <tr><td>CanGrow</td><td>False</td></tr> <tr><td>CanShrink</td><td>False</td></tr> <tr><td><b>Mode</b></td><td>OddPages</td></tr> <tr><td>Visible</td><td>True</td></tr> </table>	<b>Render</b>		CanGrow	False	CanShrink	False	<b>Mode</b>	OddPages	Visible	True	
<b>Render</b>											
CanGrow	False										
CanShrink	False										
<b>Mode</b>	OddPages										
Visible	True										

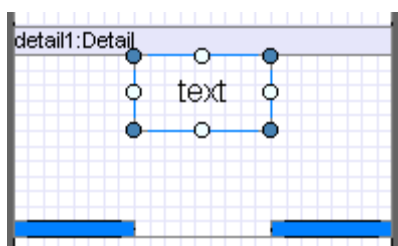
Render	
CanGrow	False
CanShrink	False
Mode	EvenPages
Visible	True

### GrowToBottom プロパティ (Render)

オブジェクトをコンテナバンドの下枠まで引き伸ばします。

TextBox.GrowToBottom プロパティの使用例

レポートテンプレート :



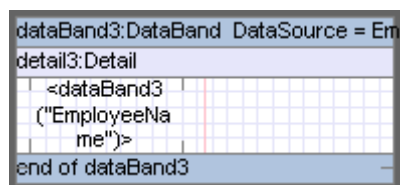
プロパティ	レポート										
<table border="1"> <thead> <tr> <th colspan="2">Render</th> </tr> </thead> <tbody> <tr> <td>CanGrow</td> <td>False</td> </tr> <tr> <td>CanShrink</td> <td>False</td> </tr> <tr> <td>GrowToBottom</td> <td>False</td> </tr> <tr> <td>Visible</td> <td>True</td> </tr> </tbody> </table>	Render		CanGrow	False	CanShrink	False	GrowToBottom	False	Visible	True	
Render											
CanGrow	False										
CanShrink	False										
GrowToBottom	False										
Visible	True										
<table border="1"> <thead> <tr> <th colspan="2">Render</th> </tr> </thead> <tbody> <tr> <td>CanGrow</td> <td>False</td> </tr> <tr> <td>CanShrink</td> <td>False</td> </tr> <tr> <td>GrowToBottom</td> <td>True</td> </tr> <tr> <td>Visible</td> <td>True</td> </tr> </tbody> </table>	Render		CanGrow	False	CanShrink	False	GrowToBottom	True	Visible	True	
Render											
CanGrow	False										
CanShrink	False										
GrowToBottom	True										
Visible	True										

### NewColumnAfter プロパティ (Render)

バンド内容の後に新しい列を追加するかを設定します。

Detail の内容の後に新しい列を追加する例:

レポートテンプレート :





プロパティ	レポート																				
<table border="1"> <tr><td colspan="2"><b>Render</b></td></tr> <tr><td>CanBreak</td><td>False</td></tr> <tr><td>CanGrow</td><td><b>True</b></td></tr> <tr><td>CanShrink</td><td><b>True</b></td></tr> <tr><td>MinSpaceAfter</td><td>0 cm</td></tr> <tr><td>NewColumnAfter</td><td>False</td></tr> <tr><td>NewColumnBefore</td><td>False</td></tr> </table>	<b>Render</b>		CanBreak	False	CanGrow	<b>True</b>	CanShrink	<b>True</b>	MinSpaceAfter	0 cm	NewColumnAfter	False	NewColumnBefore	False	<table border="1"> <tr> <td>Nancy Davolio Andrew Fuller Anne Dodsworth</td> <td>Steven Buchana Laura Callahan Anne Yorke</td> </tr> </table>	Nancy Davolio Andrew Fuller Anne Dodsworth	Steven Buchana Laura Callahan Anne Yorke				
<b>Render</b>																					
CanBreak	False																				
CanGrow	<b>True</b>																				
CanShrink	<b>True</b>																				
MinSpaceAfter	0 cm																				
NewColumnAfter	False																				
NewColumnBefore	False																				
Nancy Davolio Andrew Fuller Anne Dodsworth	Steven Buchana Laura Callahan Anne Yorke																				
<table border="1"> <tr><td colspan="2"><b>Render</b></td></tr> <tr><td>CanBreak</td><td>False</td></tr> <tr><td>CanGrow</td><td><b>True</b></td></tr> <tr><td>CanShrink</td><td><b>True</b></td></tr> <tr><td>MinSpaceAfter</td><td>0 cm</td></tr> <tr><td>NewColumnAfter</td><td><b>True</b></td></tr> <tr><td>NewColumnBefore</td><td>False</td></tr> </table>	<b>Render</b>		CanBreak	False	CanGrow	<b>True</b>	CanShrink	<b>True</b>	MinSpaceAfter	0 cm	NewColumnAfter	<b>True</b>	NewColumnBefore	False	<table border="1"> <tr> <td>Nancy Davolio</td> <td>Andrew Fuller</td> </tr> <tr> <td colspan="2"> </td> </tr> <tr> <td>Anne Dodsworth</td> <td>Steven Buchana</td> </tr> </table>	Nancy Davolio	Andrew Fuller			Anne Dodsworth	Steven Buchana
<b>Render</b>																					
CanBreak	False																				
CanGrow	<b>True</b>																				
CanShrink	<b>True</b>																				
MinSpaceAfter	0 cm																				
NewColumnAfter	<b>True</b>																				
NewColumnBefore	False																				
Nancy Davolio	Andrew Fuller																				
Anne Dodsworth	Steven Buchana																				

### NewColumnBefore プロパティ (Render)

バンドの内容の前に新しい列を追加するかを設定します。

Detail (detail4) の内容の前に新しい列を追加する例 :

レポートテンプレート :

```
dataBand3:DataBand DataSource = Em
detail4:Detail
<dataBand3("E
detail3:Detail
<dataBand3("H
end of dataBand3
```

プロパティ	レポート																		
<table border="1"> <tr><td colspan="2"><b>Render</b></td></tr> <tr><td>CanBreak</td><td><b>True</b></td></tr> <tr><td>CanGrow</td><td><b>True</b></td></tr> <tr><td>CanShrink</td><td>False</td></tr> <tr><td>MinSpaceAfter</td><td>0 cm</td></tr> <tr><td>NewColumnAfter</td><td>False</td></tr> <tr><td>NewColumnBefore</td><td>False</td></tr> </table>	<b>Render</b>		CanBreak	<b>True</b>	CanGrow	<b>True</b>	CanShrink	False	MinSpaceAfter	0 cm	NewColumnAfter	False	NewColumnBefore	False	<table border="1"> <tr> <td>Nancy Davolio (206)555-9857 Andrew Fuller</td> <td>(206) 555-9482 Anne Dodsworth (71) 555-4444</td> </tr> <tr> <td>Steven Buchana (71) 555-4444 Laura Callahan</td> <td>(71) 555-4444 Anne Yorke (71) 555-4444</td> </tr> </table>	Nancy Davolio (206)555-9857 Andrew Fuller	(206) 555-9482 Anne Dodsworth (71) 555-4444	Steven Buchana (71) 555-4444 Laura Callahan	(71) 555-4444 Anne Yorke (71) 555-4444
<b>Render</b>																			
CanBreak	<b>True</b>																		
CanGrow	<b>True</b>																		
CanShrink	False																		
MinSpaceAfter	0 cm																		
NewColumnAfter	False																		
NewColumnBefore	False																		
Nancy Davolio (206)555-9857 Andrew Fuller	(206) 555-9482 Anne Dodsworth (71) 555-4444																		
Steven Buchana (71) 555-4444 Laura Callahan	(71) 555-4444 Anne Yorke (71) 555-4444																		
<table border="1"> <tr><td colspan="2"><b>Render</b></td></tr> <tr><td>CanBreak</td><td><b>True</b></td></tr> <tr><td>CanGrow</td><td><b>True</b></td></tr> <tr><td>CanShrink</td><td>False</td></tr> <tr><td>MinSpaceAfter</td><td>0 cm</td></tr> <tr><td>NewColumnAfter</td><td>False</td></tr> <tr><td>NewColumnBefore</td><td><b>True</b></td></tr> </table>	<b>Render</b>		CanBreak	<b>True</b>	CanGrow	<b>True</b>	CanShrink	False	MinSpaceAfter	0 cm	NewColumnAfter	False	NewColumnBefore	<b>True</b>	<table border="1"> <tr> <td>Nancy Davolio (206)555-9857</td> <td>Andrew Fuller (206) 555-9482</td> </tr> <tr> <td>Anne Dodsworth (71) 555-4444</td> <td>Steven Buchana (71) 555-4444</td> </tr> </table>	Nancy Davolio (206)555-9857	Andrew Fuller (206) 555-9482	Anne Dodsworth (71) 555-4444	Steven Buchana (71) 555-4444
<b>Render</b>																			
CanBreak	<b>True</b>																		
CanGrow	<b>True</b>																		
CanShrink	False																		
MinSpaceAfter	0 cm																		
NewColumnAfter	False																		
NewColumnBefore	<b>True</b>																		
Nancy Davolio (206)555-9857	Andrew Fuller (206) 555-9482																		
Anne Dodsworth (71) 555-4444	Steven Buchana (71) 555-4444																		



### NewPageAfter プロパティ (Render)

バンド内容の後に新しいページを追加するかを設定します。

Detail の内容の後に新しいページを追加する例 :

レポートテンプレート :

```
dataBand3:DataBand DataSource = Em
detail3:Detail
  <dataBand3("EmployeeNa
end of dataBand3
```

プロパティ		レポート		
MinSpaceAfter	0 cm	<table border="1"> <tr><td>Nancy Davolio Andrew Fuller</td></tr> <tr><td>Anne Dodsworth Steven Buchana</td></tr> </table>	Nancy Davolio Andrew Fuller	Anne Dodsworth Steven Buchana
Nancy Davolio Andrew Fuller				
Anne Dodsworth Steven Buchana				
NewColumnAfter	False			
NewColumnBefore	False			
NewPageAfter	False			
NewPageBefore	False			
Visible	True	<table border="1"> <tr><td>Nancy Davolio</td></tr> <tr><td>Andrew Fuller</td></tr> </table>	Nancy Davolio	Andrew Fuller
Nancy Davolio				
Andrew Fuller				
MinSpaceAfter	0 cm			
NewColumnAfter	False			
NewColumnBefore	False			
NewPageAfter	<b>True</b>			
NewPageBefore	False			
Visible	True			

### NewPageBefore プロパティ (Render)

バンド内容の前に新しいページを追加するかどうかを設定します。

Detail (detail3) の内容の前に新しいページを追加する例 :

レポートテンプレート :

```
dataBand3:DataBand DataSource = Em
detail4:Detail
  <dataBand3("EmployeeName")>
detail3:Detail
  <dataBand3("HomePhone")>
end of dataBand3
```

プロパティ		レポート		
MinSpaceAfter	0 cm	<table border="1"> <tr><td>Nancy Davolio (206)555-9857 Andrew Fuller</td></tr> <tr><td>(206) 555-9482 Anne Dodsworth (71) 555-4444</td></tr> </table>	Nancy Davolio (206)555-9857 Andrew Fuller	(206) 555-9482 Anne Dodsworth (71) 555-4444
Nancy Davolio (206)555-9857 Andrew Fuller				
(206) 555-9482 Anne Dodsworth (71) 555-4444				
NewColumnAfter	False			
NewColumnBefore	False			
NewPageAfter	False			
NewPageBefore	False			
Visible	True			

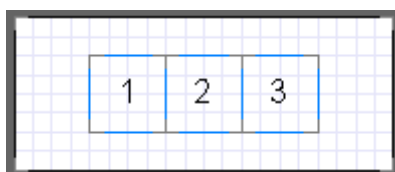
MinSpaceAfter	0 cm	<table border="1"> <tr> <td colspan="2">Nancy Davolio</td> </tr> <tr> <td colspan="2">(206)555-9857</td> </tr> <tr> <td colspan="2">Andrew Fuller</td> </tr> <tr> <td colspan="2">(206) 555-9482</td> </tr> <tr> <td colspan="2">Anne Dodsworth</td> </tr> </table>	Nancy Davolio		(206)555-9857		Andrew Fuller		(206) 555-9482		Anne Dodsworth	
Nancy Davolio												
(206)555-9857												
Andrew Fuller												
(206) 555-9482												
Anne Dodsworth												
NewColumnAfter	False											
NewColumnBefore	False											
NewPageAfter	False											
NewPageBefore	True											
Visible	True											

### Visible プロパティ (Render)

レポートのオブジェクトの表示/非表示を設定します。

数字「2」のテキストボックスの Visible プロパティの例

レポートテンプレート :



プロパティ	レポート										
<table border="1"> <tr> <td colspan="2"><b>Render</b></td> </tr> <tr> <td>CanGrow</td> <td>False</td> </tr> <tr> <td>CanShrink</td> <td>False</td> </tr> <tr> <td>GrowToBottom</td> <td>False</td> </tr> <tr> <td>Visible</td> <td>True</td> </tr> </table>	<b>Render</b>		CanGrow	False	CanShrink	False	GrowToBottom	False	Visible	True	
<b>Render</b>											
CanGrow	False										
CanShrink	False										
GrowToBottom	False										
Visible	True										
<table border="1"> <tr> <td colspan="2"><b>Render</b></td> </tr> <tr> <td>CanGrow</td> <td>False</td> </tr> <tr> <td>CanShrink</td> <td>False</td> </tr> <tr> <td>GrowToBottom</td> <td>False</td> </tr> <tr> <td>Visible</td> <td>False</td> </tr> </table>	<b>Render</b>		CanGrow	False	CanShrink	False	GrowToBottom	False	Visible	False	
<b>Render</b>											
CanGrow	False										
CanShrink	False										
GrowToBottom	False										
Visible	False										

注意：コンテナバンドが非表示の場合、そこに格納されているオブジェクトもすべて非表示になります。

### テキスト出力

SharpShooter Reports では、以下のオブジェクトを使用してテキスト入力が行われます。

オブジェクト	説明
TextBox	普通のテキスト情報を入力でき、テキスト表示を調整できます。
AdvancedText	書式付きテキストを出力します
RichText	RTF テキストを出力します

### 静的なテキスト

Text プロパティは、TextBox オブジェクトに出力するテキストを入力するために使用します。このプロパティはテキストエディタで定義されます。テキストエディタを開くには、テキストボックスをダブルクリックするか、プロパティグリッドの Text プロパティのボタンをクリックしてください。また、テキストセルに直接テキストを入力することもできます。

<b>Appearance</b>	
Angle	0
Border	
Fill	[None]
Font	Arial; 12pt
Margins	0; 0; 0; 0 cm
RightToLeft	False
StringTrimming	Character
StyleName	
Text	text
TextAlign	MiddleCenter
TextFill	[None]
TextFormat	
ToolTip	

### データソースのテキスト

データソースのテキストを入力するには、**TextBox** オブジェクトの **Value** プロパティ（選択したフィールドの値を取得します）を設定してください。**GetData** メソッドか **DataBand** のインデクサを使用できます。

### セル枠、セルのテキスト

**Border** プロパティを使用して、オブジェクトの枠を定義します。テキストを表として入力する例：

Caption		
Column	Cell#1	Cell#2
	Cell#3	

### テキストの出力方向

**TextBox** や **AdvancedText** オブジェクトのテキストの出力方向を変更できます。それには、**RightToLeft** プロパティを **True** に設定します。

Text プロパティ = "7 8 9 10 11"

プロパティ	レポート								
<table border="1"> <tr><td>RightToLeft</td><td>False</td></tr> <tr><td>StringTrimming</td><td>Character</td></tr> <tr><td>StyleName</td><td></td></tr> <tr><td>Text</td><td>7 8 9 10 11</td></tr> </table>	RightToLeft	False	StringTrimming	Character	StyleName		Text	7 8 9 10 11	
RightToLeft	False								
StringTrimming	Character								
StyleName									
Text	7 8 9 10 11								
<table border="1"> <tr><td>RightToLeft</td><td>True</td></tr> <tr><td>StringTrimming</td><td>Character</td></tr> <tr><td>StyleName</td><td></td></tr> <tr><td>Text</td><td>7 8 9 10 11</td></tr> </table>	RightToLeft	True	StringTrimming	Character	StyleName		Text	7 8 9 10 11	
RightToLeft	True								
StringTrimming	Character								
StyleName									
Text	7 8 9 10 11								

### テキストの切り取り

出力するテキストがオブジェクトの領域に収まらない場合は **StringTrimming** プロパティを使用してテキストの切り取りを設定できます。

プロパティ	レポート								
表示可能な最後の文字の後のテキストは切り取られます									
<table border="1"> <tr><td>RightToLeft</td><td>False</td></tr> <tr><td>StringTrimming</td><td>Character</td></tr> <tr><td>StyleName</td><td></td></tr> <tr><td>Text</td><td>one two three</td></tr> </table>	RightToLeft	False	StringTrimming	Character	StyleName		Text	one two three	
RightToLeft	False								
StringTrimming	Character								
StyleName									
Text	one two three								
表示可能な最後の単語の後のテキストは切り取られます									

RightToLeft	False	one two
StringTrimming	Word	
StyleName		
Text	one two three	
表示可能な最後の文字の後のテキストは切り取られ、省略記号が追加されます		
RightToLeft	False	one tw...
StringTrimming	EllipsisCharacter	
StyleName		
Text	one two three	
表示可能な最後の単語の後のテキストが切り取られ、省略記号が追加されます		
RightToLeft	False	one ...
StringTrimming	EllipsisWord	
StyleName		
Text	one two three	
先頭と末尾のテキストが表示され、その間は省略記号が追加されます		
RightToLeft	False	on...ree
StringTrimming	EllipsisWord	
StyleName		
Text	one two three	

**ClipText** プロパティは、TextBox オブジェクトの領域に入り切らないテキスト行の処理方法を定義します。

プロパティ		レポート
Border	Border	
ClipText	False	
Fill	[None]	
Font	Calibri; 39,75pt; Bold=Off; It	
Border	Border	TEXT
ClipText	True	
Fill	[None]	
Font	Calibri; 39,75pt; Bold=Off; It	

### テキストオブジェクトのサイズ操作

**Size** プロパティは、オブジェクトのサイズを定義します。生成時にオブジェクトを適切なサイズに変更する機能があります。これは、テンプレートの作成時にオブジェクトのコンテンツが取得されない場合やレポート生成時にコンテンツが変わる場合に役立ちます。**CanShrink** や **CanGrow** プロパティは内容に応じてオブジェクトを大きくする（またはプロパティの値に応じて大きくさせないようにする）ことができます。

**GrowToBottom** プロパティはオブジェクトのサイズをコンテナバンドの下枠まで大きくするかを定義します。

### テキストスタイル

レポートを作成するための機能が多数あります。**Fill** プロパティは背景色を定義します。**Font** プロパティはフォント型、テキストスタイル（標準、太字、斜体、下線、打ち消し線）などのテキストの見た目を設定します。テキストボックスのテキストの揃え方や位置は **TextAlign** プロパティで定義します。**TextFill** プロパティを使用してスタイルやテキストの背景色を設定できます。

また、複数のオブジェクトに 1 つのスタイルを適用させることができます。スタイルシートエディタはスタイルエディタとして使用でき、**StyleName** プロパティはオブジェクトにスタイルを適用します。

### テキスト形式

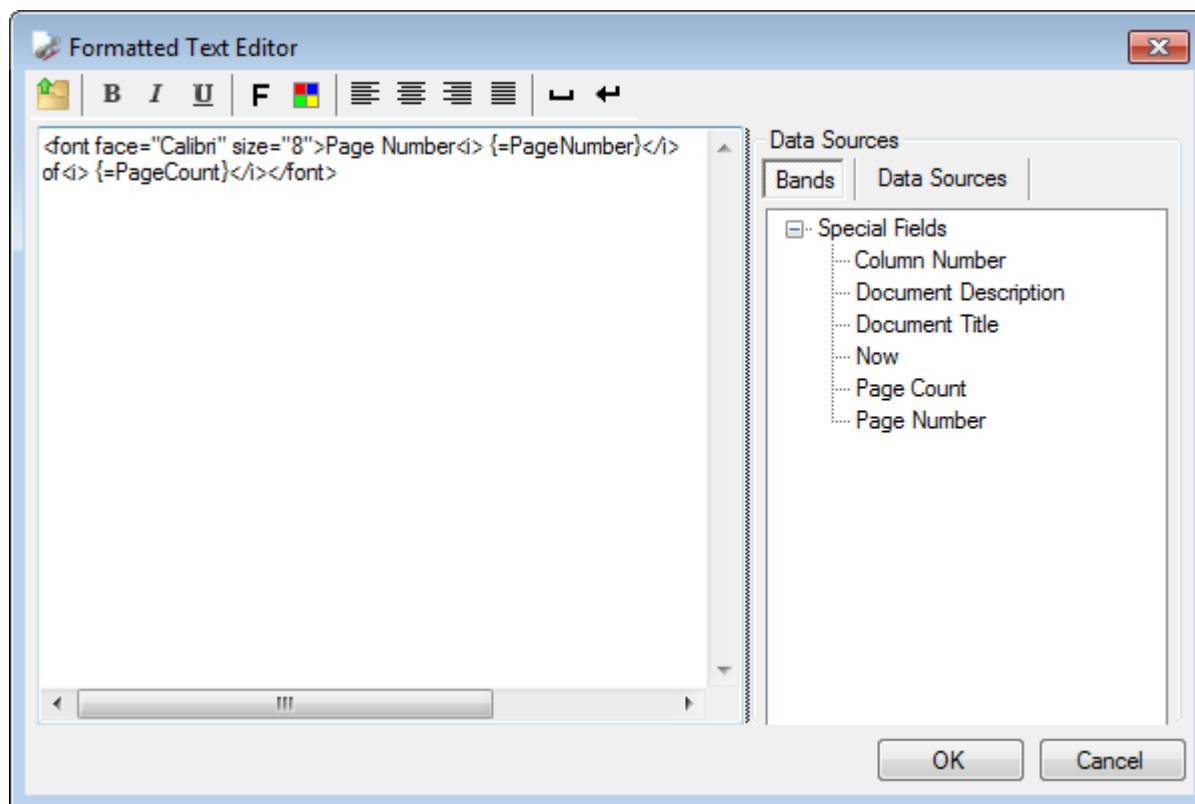
テキスト形式を指定して表示するには、**TextFormat** プロパティを使用します。フォーマットエディタで複数のテキスト形式（標準、数値、通貨、日付、時刻、パーセンテージ、ユーザー定義）を取得できます。

### 書式付きテキスト

**AdvancedText** コンポーネントは、データを書式付きテキストとして入力できます。段落、テキストスタイルの設定や記述したテキストにステートメントを正しく使うオプションがあります。書式付きテキストは、

HTML のようなマークアップ言語（Text プロパティ）や RTF 形式のサブセット（RTF プロパティ）から指定できます。

書式付きテキストエディタ（Text プロパティ）で HTML のようなタグを使ってテキストの書式設定が行えます。



### 段落の書式設定

`<P></P>` タグは、選択したテキストを別の段落にします。

Align 属性は、段落の水平方向のテキストの位置合わせを指定します。使用可能な値は次の通りです。

align = "left" - 段落の左揃え

align = "right" - 段落の右揃え

align = "center" - 中央揃え

align = "justify" - 両端揃え

Align の規定値は「left」（左揃え）です。編集するテキストを選択し、ツールバーの「左揃え」、「中央揃え」、「右揃え」、「両端揃え」ボタンを使用して揃え方を設定してください。

### テキストの書式設定

`<Font>` タグは、この後に設定されたテキストの出力スタイルを設定し、終了タグ `</font>` はこの設定を解除します。`<font>` タグの囲い込みが可能です。テキストスタイルは次の属性を使って指定できます。

face - フォント名を設定します。

size - フォントサイズを設定します。

スタイルを瞬時に定義するには、「フォント」ボタンを押してスタイルエディタを開いてください。

color - フォントの色を設定します。



利用可能な値：色名（例：color="red"）、RGB の各カラーコンポーネントの 16 進形式（例：color="#FF0000"）。色を瞬時に選択するには、「色」ボタンを押して ColorPicker を開いてください。

Tag <B> - この後のテキストは太字になります。終了タグ </B> はこの設定を解除します（「太字」ボタン）。

Tag <I> - この後のテキストは斜体になります。終了タグ </I> はこの設定を解除します（「斜体」ボタン）。

Tag <U> - この後のテキストには下線が付きます。終了タグ </U> はこの設定を解除します（「下線」ボタン）。

Tag <BR> - 段落内で行を折り返します（「改行を挿入する」ボタン）。

### 記号の追加

"&nbsp;" - スペース（「スペースを挿入する」ボタン）

"&" - 「&」記号

"&lt;" - 「<」記号

"&#ddd;" - ddd コード付きの記号

注意：テキストが段落として書かれていない場合、TextAlign プロパティの位置合わせが適用されます。明白なフォントが設定されていない場合、Font プロパティの設定が適用されます。テキストの色が指定されていない場合、TextFill プロパティの設定が適用されます。テキスト操作時の改行、複数のスペース、連続/不明なタグや属性は無視されます。タグ名の大文字/小文字の区別はありません。

### 式を使用する

書式付きテキストに命令文を直接使用でき、その結果はテキストに追加されます。式は {=<Statement>} のように指定します（{=<Statement>} - 設定したスクリプト言語の式）。利用可能な型やオブジェクトをすべて式に使用できます。テキストを挿入する前に、Format プロパティで指定したマスクの書式が式の結果に適用されます。エラーが発生した場合、エラーメッセージが式の結果としてテキストに挿入されます。

DataSources セクションは、ドキュメントやデータソースフィールドに特別なプロパティを追加します。

### RTF プロパティの使用

RTF 形式のサブセットがテキストの書式に使用されます。RTF 形式の次の構造が使用されます：フォントテーブル、色テーブル、コメント、段落書式の演算子（"\par"、"\pard"、"\ql"、"\qr"、"\qc"、"\qj"、"\line"）、テキスト書式の演算子（"\fxx"、"\fsxx"、"\cfxx"、"\b"、"\i"、"\u"）。残りのすべての構造や演算子は無視されます。Rtf プロパティは、書き込み専用でデザインされています。Rtf プロパティは、Text プロパティに接続されます。Rtf プロパティが指定されると、Text プロパティの値が自動的に指定されます。

RTF ファイルからテキストを読み込むには、「RTF ドキュメントを開く」ボタンを押してください。

### テキストのページ分割

AdvancedText オブジェクトは、1 つのオブジェクトのテキストを 1 ページまたは複数ページに出力します。それには CanBreak プロパティを使用してください。

### RTF 形式のテキスト

RichText オブジェクトは、RTF 形式のテキストを出力するために使用します。RichText エディタを使って、ファイルからテキストを読み込みます。

## リスト

### シンプルリスト

**DataBand** を使用してリストを作成します。**DataBand** はその中に格納されたオブジェクトを繰り返し出力します。ビジュアルオブジェクトをリストとして表示するには、**DataBand** 内の **Detail** に格納しなければなりません。**Detail** は行の高さを設定します。**InstanceCount** プロパティは何回表示するかを設定します。

Data	
DataSource	
FilterExpression	
InstanceCount	5
Sort	(Collection)
Tag	
Totals	(Collection)

命令文 : `<DataBand name>.LineNumber()` は、行番号を取得するために使用します。

行番号を出力するリスト例 :

```
dataBand1.InstanceCount = 5;
```

```
textBox.Value = "line number = " + dataBand1.LineNumber().ToString() + " of " + dataBand1.InstanceCount.ToString();
```

レポートテンプレート	レポート
<pre>dataBand1:DataBand detail1:Detail &lt;"line number: " + dataBa end of dataBand1</pre>	
<pre>dataBand1:DataBand detail1:Detail &lt;"line number: " + dataBa end of dataBand1</pre>	

### データソースのリスト

**DataBand** はソースのデータに基づいてリストを作成します。データソースは、**DataSource** プロパティで設定されます。

注意 : 出力される行数はデータソースの行数によるので、**InstanceCount** プロパティは無視されます。

Data	
DataSource	Employees ▼
FilterExpression	
InstanceCount	0
Sort	(Collection)
Tag	
Totals	(Collection)

現在の（出力しなければならない）フィールドにアクセスするには、`GetData(“<Data source name>.<Field name>”)` または `<DataBand name>[“<Field name>”]` のどちらかのスクリプトを使用します。テキスト情報は、データソースタブのドキュメントツリーから対象のフィールドを（**DataBand** 内の）**Detail** にドラッグすることで出力できます。



担当者名を出力するリスト例：

```
dataBand.DataSource = Employees;
```

```
textBox.Value = dataBand["EmployeeName"];
```

レポートテンプレート	レポート
<pre>dataBand:DataBand DataSource = Emp detail:Detail &lt;dataBand["EmployeeNa end of dataBand</pre>	<p>Nancy Davolio Andrew Fuller Anne Dodsworth Steven Buchana Laura Callahan</p>

### 段組みのリスト

レポートの領域を減らしてレポートをより視覚的にするには、リストを段組で出力する方法があります。**DataBand** の **ColumnsCount** プロパティに列数を設定します。また、リストの列間隔も **ColumnsGap** プロパティで設定できます。

社員のリスト例：

```
dataBand.DataSource = Employees;
```

```
textBox.Value = dataBand["EmployeeName"];
```

レポートテンプレート：

<pre>dataBand:DataBand DataSource = Emp</pre>
<pre>detail:Detail</pre>
<pre>&lt;dataBand["</pre>
<pre>end of dataBand</pre>

プロパティ	レポート
<p><b>Layout</b></p> <p>ColumnsCount: 0</p> <p>ColumnsGap: 0 cm</p> <p>Location: 0; 0 cm</p> <p>Size: 5; 2 cm</p>	<p>Nancy Davolio Andrew Fuller Anne Dodsworth</p> <p>Steven Buchana Laura Callahan Anne Yorke</p>
<p><b>Layout</b></p> <p>ColumnsCount: 2</p> <p>ColumnsGap: 0 cm</p> <p>Location: 0; 0 cm</p> <p>Size: 6; 1.6 cm</p>	<p>Nancy Davolio      Steven Buchana Andrew Fuller      Laura Callahan Anne Dodsworth      Anne Yorke</p>

<b>Layout</b>		Nancy	Steven
ColumnsCount	2	Davolio	Buchana
ColumnsGap	0,25 cm	Andrew	Laura
Location	0; 0 cm	Fuller	Callahan
Size	5; 1,5 cm	Anne	Anne Yorke
		Dodsworth	

### リストのヘッダー

Header はリスト、ページ、または列の先頭にデータを一回出力するために使用します。ヘッダーはそのため  
に使用され、DataBand 内に貼り付けなければなりません。RepeatEveryColumn や RepeatEveryPage プ  
ロパティはヘッダーの反復モードに役立ちます。

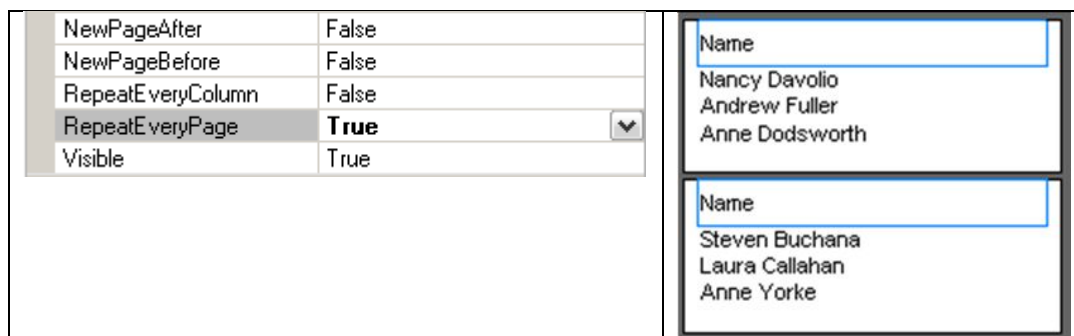
リストのヘッダーの使用例：

```
dataBand3.DataSource = Employees; textBox1.Text = "Name"; textBox2.Value =  
dataBand3("EmployeeName");
```

レポートテンプレート：

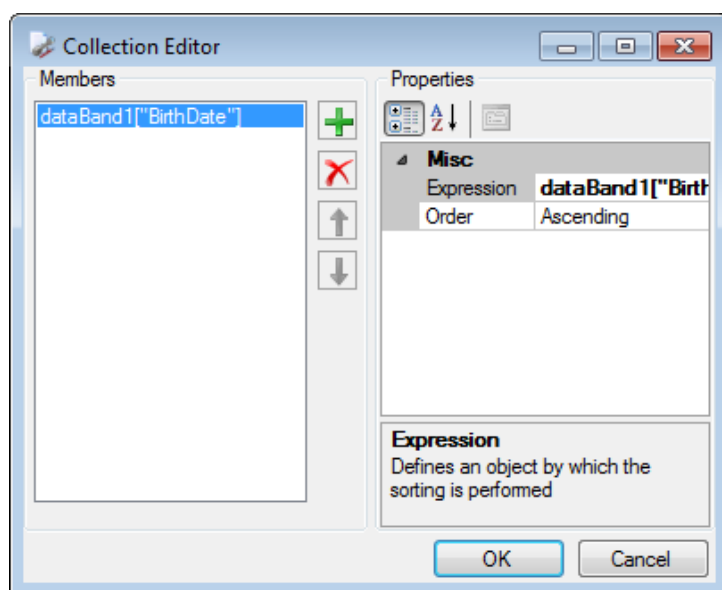
```
dataBand:DataBand DataSource = Emp  
header5:Header  
Name  
detail:Detail  
<dataBand["E  
end of dataBand
```

プロパティ	レポート												
<table border="1"> <tr><td>NewColumnBefore</td><td>False</td></tr> <tr><td>NewPageAfter</td><td>False</td></tr> <tr><td>NewPageBefore</td><td>False</td></tr> <tr><td>RepeatEveryColumn</td><td>False</td></tr> <tr><td>RepeatEveryPage</td><td>False</td></tr> <tr><td>Visible</td><td>True</td></tr> </table>	NewColumnBefore	False	NewPageAfter	False	NewPageBefore	False	RepeatEveryColumn	False	RepeatEveryPage	False	Visible	True	
NewColumnBefore	False												
NewPageAfter	False												
NewPageBefore	False												
RepeatEveryColumn	False												
RepeatEveryPage	False												
Visible	True												
<table border="1"> <tr><td>NewColumnBefore</td><td>False</td></tr> <tr><td>NewPageAfter</td><td>False</td></tr> <tr><td>NewPageBefore</td><td>False</td></tr> <tr><td>RepeatEveryColumn</td><td><b>True</b></td></tr> <tr><td>RepeatEveryPage</td><td>False</td></tr> <tr><td>Visible</td><td>True</td></tr> </table>	NewColumnBefore	False	NewPageAfter	False	NewPageBefore	False	RepeatEveryColumn	<b>True</b>	RepeatEveryPage	False	Visible	True	
NewColumnBefore	False												
NewPageAfter	False												
NewPageBefore	False												
RepeatEveryColumn	<b>True</b>												
RepeatEveryPage	False												
Visible	True												
<table border="1"> <tr><td>NewPageAfter</td><td>False</td></tr> <tr><td>NewPageBefore</td><td>False</td></tr> <tr><td>RepeatEveryColumn</td><td>False</td></tr> <tr><td>RepeatEveryPage</td><td>False</td></tr> <tr><td>Visible</td><td>True</td></tr> </table>	NewPageAfter	False	NewPageBefore	False	RepeatEveryColumn	False	RepeatEveryPage	False	Visible	True			
NewPageAfter	False												
NewPageBefore	False												
RepeatEveryColumn	False												
RepeatEveryPage	False												
Visible	True												



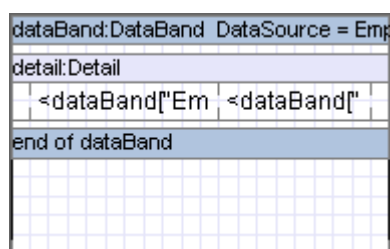
### リストの並べ替え

DataBand はデータソースに保存されている項目順で出力します。DataBand の Sort プロパティは特別な出力順を設定します。Sort プロパティはコレクションエディタを使用して定義します。Expression プロパティは並べ替えの式を設定するために使用します（スクリプトを式として使用しなければなりません）。Order プロパティは並べ替えの順番（昇順または降順）を定義します。



並べ替え条件に応じて社員の名前と誕生日を表示するリスト例：

レポートテンプレート：



```
dataBand.DataSource = Employees;
```

```
textBoxName.Value = dataBand["EmployeeName"];
```

```
textBoxDate.Value = dataBand["BirthDate"];
```

並べ替え条件	レポート
--------	------



なし	<table border="1"> <tr><td>Nancy Davolio</td><td>11.11.1976</td></tr> <tr><td>Andrew Fuller</td><td>25.02.1969</td></tr> <tr><td>Anne Dodswort</td><td>30.06.1981</td></tr> <tr><td>Steven Buchan</td><td>15.12.1984</td></tr> <tr><td>Laura Callahan</td><td>28.09.1987</td></tr> </table>	Nancy Davolio	11.11.1976	Andrew Fuller	25.02.1969	Anne Dodswort	30.06.1981	Steven Buchan	15.12.1984	Laura Callahan	28.09.1987						
Nancy Davolio	11.11.1976																
Andrew Fuller	25.02.1969																
Anne Dodswort	30.06.1981																
Steven Buchan	15.12.1984																
Laura Callahan	28.09.1987																
<div style="border: 1px solid gray; padding: 2px;"> <p>☐ Misc</p> <table border="1"> <tr><td>Expression</td><td>dataBand["EmployeeName"]</td><td>...</td></tr> <tr><td>Order</td><td colspan="2">Ascending</td></tr> </table> </div>	Expression	dataBand["EmployeeName"]	...	Order	Ascending		<table border="1"> <tr><td>Andrew Fuller</td><td>25.02.1969</td></tr> <tr><td>Anne Dodswort</td><td>30.06.1981</td></tr> <tr><td>Anne Yorke</td><td>03.05.1969</td></tr> <tr><td>Laura Callahan</td><td>28.09.1987</td></tr> <tr><td>Nancy Davolio</td><td>11.11.1976</td></tr> </table>	Andrew Fuller	25.02.1969	Anne Dodswort	30.06.1981	Anne Yorke	03.05.1969	Laura Callahan	28.09.1987	Nancy Davolio	11.11.1976
Expression	dataBand["EmployeeName"]	...															
Order	Ascending																
Andrew Fuller	25.02.1969																
Anne Dodswort	30.06.1981																
Anne Yorke	03.05.1969																
Laura Callahan	28.09.1987																
Nancy Davolio	11.11.1976																
<div style="border: 1px solid gray; padding: 2px;"> <p>☐ Misc</p> <table border="1"> <tr><td>Expression</td><td>dataBand["BirthDate"]</td><td>...</td></tr> <tr><td>Order</td><td colspan="2">Descending</td></tr> </table> </div>	Expression	dataBand["BirthDate"]	...	Order	Descending		<table border="1"> <tr><td>Laura Callahan</td><td>28.09.1987</td></tr> <tr><td>Steven Buchan</td><td>15.12.1984</td></tr> <tr><td>Anne Dodswort</td><td>30.06.1981</td></tr> <tr><td>Nancy Davolio</td><td>11.11.1976</td></tr> <tr><td>Anne Yorke</td><td>03.05.1969</td></tr> </table>	Laura Callahan	28.09.1987	Steven Buchan	15.12.1984	Anne Dodswort	30.06.1981	Nancy Davolio	11.11.1976	Anne Yorke	03.05.1969
Expression	dataBand["BirthDate"]	...															
Order	Descending																
Laura Callahan	28.09.1987																
Steven Buchan	15.12.1984																
Anne Dodswort	30.06.1981																
Nancy Davolio	11.11.1976																
Anne Yorke	03.05.1969																

### リストのフィルタ処理

ある条件を満たしているデータを表示する場合には、DataBand の FilterExpression プロパティを使用できます。

性別によって社員の名前を表示するリスト例：

レポートテンプレート：

dataBand:DataBand DataSource = Emp
detail:Detail
<dataBand["EmployeeNa
end of dataBand

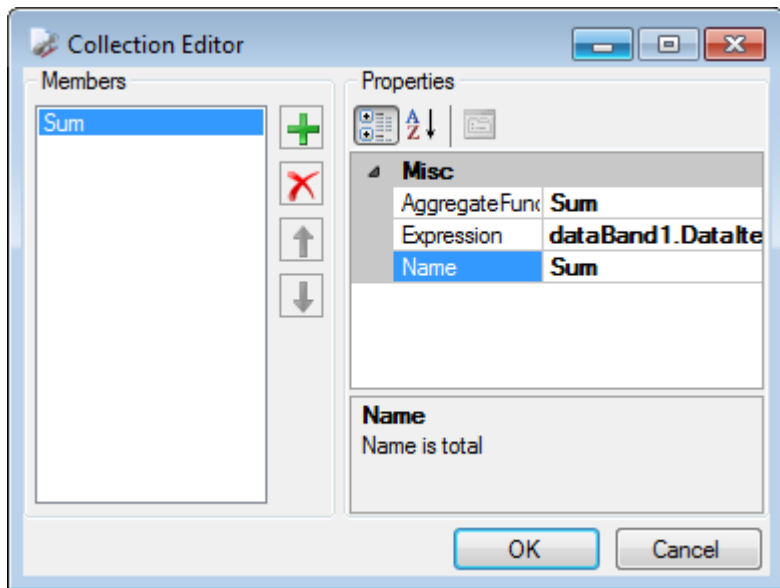
```
dataBand.DataSource = Employees;
```

```
textBoxName.Value = dataBand["EmployeeName"];
```

プロパティ	レポート				
dataBand.FilterExpression = dataBand1["Sex"].ToString() == "f"	<table border="1"> <tr><td>Nancy Davolio</td></tr> <tr><td>Anne Dodsworth</td></tr> <tr><td>Laura Callahan</td></tr> <tr><td>Anne Yorke</td></tr> </table>	Nancy Davolio	Anne Dodsworth	Laura Callahan	Anne Yorke
Nancy Davolio					
Anne Dodsworth					
Laura Callahan					
Anne Yorke					
dataBand.FilterExpression = dataBand1["Sex"].ToString() == "m"	<table border="1"> <tr><td>Andrew Fuller</td></tr> <tr><td>Steven Buchana</td></tr> </table>	Andrew Fuller	Steven Buchana		
Andrew Fuller					
Steven Buchana					

## 最終的なリストの値

リストの合計を計算する場合、集計リストを使用できます。DataBand にデータを入力するには、Totals プロパティのコレクションエディタコレクションエディタでリストの集計関数を指定できます。



**AggregateFunction** プロパティは集計関数を指定します。集計式は **Expression** プロパティで指定します。関数名は **Name** プロパティに指定します。スクリプト : **GetTotal("<aggregate name>")** は、指定した関数を呼び出してその合計を取得するために使用します。**GetRunningTotal("<aggregate name>")** は、現在の累計を取得するために使用します。集計値を出力する前に集計したデータを取得することができます。**GetTotal** の結果はこの関数を呼び出す要素の位置（前または後）には依存しません。**Detail** に **GetRunningTotal** の値を出力することをお勧めいたします。**Header** からの呼び出しは累計された値がないので役に立ちません。**Footer** から呼び出した **GetRunningTotal** の値は **GetTotal** の値と一致します。

レポートジェネレータは、組み込まれている次の 12 の集計関数をサポートしています。

集計関数の使用例 :

レポートテンプレート :

```
dataBand.DataSource = Numbers;
detail:Detail
<dataBand["Value"]>
footer:Footer
<GetTotal("Func")>
end of dataBand
```

```
dataBand.DataSource = Numbers;
textBox1.Value = dataBand["Value"];
textBox2.Value = GetTotal("Func");
```

プロパティ	レポート												
合計 :													
<table border="1"> <tr> <td colspan="2">Misc</td> </tr> <tr> <td>AggregateFunction</td> <td>Sum</td> </tr> <tr> <td>Expression</td> <td>dataBand["Value"]</td> </tr> <tr> <td>Name</td> <td>Func</td> </tr> </table>	Misc		AggregateFunction	Sum	Expression	dataBand["Value"]	Name	Func	<table border="1"> <tr><td>1</td></tr> <tr><td>13</td></tr> <tr><td>77</td></tr> <tr><td>91</td></tr> </table>	1	13	77	91
Misc													
AggregateFunction	Sum												
Expression	dataBand["Value"]												
Name	Func												
1													
13													
77													
91													
平均 :													

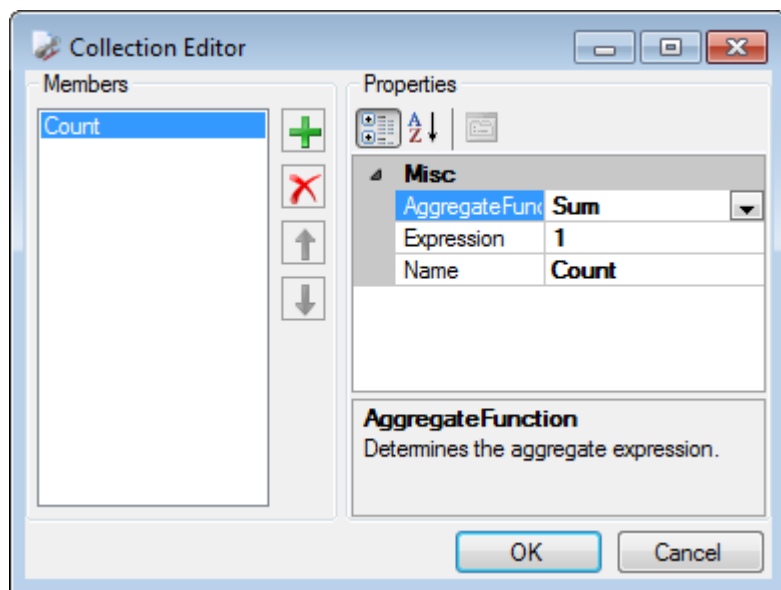


<div style="border: 1px solid gray; padding: 2px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>☐ Misc</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">AggregateFunction</td> <td>Avg</td> </tr> <tr> <td>Expression</td> <td>dataBand["Value"]</td> </tr> <tr> <td>Name</td> <td>Func</td> </tr> </table> </div>	AggregateFunction	Avg	Expression	dataBand["Value"]	Name	Func	<div style="border: 1px solid black; padding: 5px; width: 100px;">           1 13 77 <span style="border: 1px solid blue; padding: 2px;">30,33333</span> </div>
AggregateFunction	Avg						
Expression	dataBand["Value"]						
Name	Func						
最小値 :							
<div style="border: 1px solid gray; padding: 2px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>☐ Misc</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">AggregateFunction</td> <td>Min</td> </tr> <tr> <td>Expression</td> <td>dataBand["Value"]</td> </tr> <tr> <td>Name</td> <td>Func</td> </tr> </table> </div>	AggregateFunction	Min	Expression	dataBand["Value"]	Name	Func	<div style="border: 1px solid black; padding: 5px; width: 100px;">           1 13 77 <span style="border: 1px solid blue; padding: 2px;">1</span> </div>
AggregateFunction	Min						
Expression	dataBand["Value"]						
Name	Func						
最大値 :							
<div style="border: 1px solid gray; padding: 2px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>☐ Misc</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">AggregateFunction</td> <td>Max</td> </tr> <tr> <td>Expression</td> <td>dataBand["Value"]</td> </tr> <tr> <td>Name</td> <td>Func</td> </tr> </table> </div>	AggregateFunction	Max	Expression	dataBand["Value"]	Name	Func	<div style="border: 1px solid black; padding: 5px; width: 100px;">           1 13 77 <span style="border: 1px solid blue; padding: 2px;">77</span> </div>
AggregateFunction	Max						
Expression	dataBand["Value"]						
Name	Func						
個数 :							
<div style="border: 1px solid gray; padding: 2px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>☐ Misc</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">AggregateFunction</td> <td>Count</td> </tr> <tr> <td>Expression</td> <td>dataBand["Value"]</td> </tr> <tr> <td>Name</td> <td>Func</td> </tr> </table> </div>	AggregateFunction	Count	Expression	dataBand["Value"]	Name	Func	<div style="border: 1px solid black; padding: 5px; width: 100px;">           1 13 77 <span style="border: 1px solid blue; padding: 2px;">3</span> </div>
AggregateFunction	Count						
Expression	dataBand["Value"]						
Name	Func						
標準偏差 :							
<div style="border: 1px solid gray; padding: 2px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>☐ Misc</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">AggregateFunction</td> <td>Standard Deviatio</td> </tr> <tr> <td>Expression</td> <td>dataBand["Value"]</td> </tr> <tr> <td>Name</td> <td>Func</td> </tr> </table> </div>	AggregateFunction	Standard Deviatio	Expression	dataBand["Value"]	Name	Func	<div style="border: 1px solid black; padding: 5px; width: 100px;">           1 13 77 <span style="border: 1px solid blue; padding: 2px;">33,35998</span> </div>
AggregateFunction	Standard Deviatio						
Expression	dataBand["Value"]						
Name	Func						
推定標準偏差 :							
<div style="border: 1px solid gray; padding: 2px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>☐ Misc</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">AggregateFunction</td> <td>Estimate of Standa</td> </tr> <tr> <td>Expression</td> <td>dataBand["Value"]</td> </tr> <tr> <td>Name</td> <td>Func</td> </tr> </table> </div>	AggregateFunction	Estimate of Standa	Expression	dataBand["Value"]	Name	Func	<div style="border: 1px solid black; padding: 5px; width: 100px;">           1 13 77 <span style="border: 1px solid blue; padding: 2px;">40,85747</span> </div>
AggregateFunction	Estimate of Standa						
Expression	dataBand["Value"]						
Name	Func						
分散 :							
<div style="border: 1px solid gray; padding: 2px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>☐ Misc</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">AggregateFunction</td> <td>Variance</td> </tr> <tr> <td>Expression</td> <td>dataBand["Value"]</td> </tr> <tr> <td>Name</td> <td>Func</td> </tr> </table> </div>	AggregateFunction	Variance	Expression	dataBand["Value"]	Name	Func	<div style="border: 1px solid black; padding: 5px; width: 100px;">           1 13 77 <span style="border: 1px solid blue; padding: 2px;">1112,888</span> </div>
AggregateFunction	Variance						
Expression	dataBand["Value"]						
Name	Func						
標本分散 :							
<div style="border: 1px solid gray; padding: 2px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>☐ Misc</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">AggregateFunction</td> <td>Estimate of varian</td> </tr> <tr> <td>Expression</td> <td>dataBand["Value"]</td> </tr> <tr> <td>Name</td> <td>Func</td> </tr> </table> </div>	AggregateFunction	Estimate of varian	Expression	dataBand["Value"]	Name	Func	<div style="border: 1px solid black; padding: 5px; width: 100px;">           1 13 77 <span style="border: 1px solid blue; padding: 2px;">1669,333</span> </div>
AggregateFunction	Estimate of varian						
Expression	dataBand["Value"]						
Name	Func						
中央値 :							

<div style="border: 1px solid gray; padding: 2px;"> <div style="background-color: #e0e0e0; padding: 2px;"> <span style="font-size: 12px;">[-] Misc</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">AggregateFunction</td> <td>Median</td> </tr> <tr> <td>Expression</td> <td>dataBand["Value"]</td> </tr> <tr> <td>Name</td> <td>Func</td> </tr> </table> </div>	AggregateFunction	Median	Expression	dataBand["Value"]	Name	Func	<div style="border: 1px solid black; padding: 5px; width: 60px; margin: 0 auto;">             1 13 77 <span style="border: 1px solid blue; padding: 2px;">13</span> </div>
AggregateFunction	Median						
Expression	dataBand["Value"]						
Name	Func						
歪度 :							
<div style="border: 1px solid gray; padding: 2px;"> <div style="background-color: #e0e0e0; padding: 2px;"> <span style="font-size: 12px;">[-] Misc</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">AggregateFunction</td> <td>Skewness</td> </tr> <tr> <td>Expression</td> <td>dataBand["Value"]</td> </tr> <tr> <td>Name</td> <td>Func</td> </tr> </table> </div>	AggregateFunction	Skewness	Expression	dataBand["Value"]	Name	Func	<div style="border: 1px solid black; padding: 5px; width: 60px; margin: 0 auto;">             1 13 77 <span style="border: 1px solid blue; padding: 2px;">0,639105</span> </div>
AggregateFunction	Skewness						
Expression	dataBand["Value"]						
Name	Func						
尖度 :							
<div style="border: 1px solid gray; padding: 2px;"> <div style="background-color: #e0e0e0; padding: 2px;"> <span style="font-size: 12px;">[-] Misc</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">AggregateFunction</td> <td>Kurtosis</td> </tr> <tr> <td>Expression</td> <td>dataBand["Value"]</td> </tr> <tr> <td>Name</td> <td>Func</td> </tr> </table> </div>	AggregateFunction	Kurtosis	Expression	dataBand["Value"]	Name	Func	<div style="border: 1px solid black; padding: 5px; width: 60px; margin: 0 auto;">             1 13 77 <span style="border: 1px solid blue; padding: 2px;">-1,5</span> </div>
AggregateFunction	Kurtosis						
Expression	dataBand["Value"]						
Name	Func						

### ページの項目数

DataBand の Total プロパティはそのページの項目数を数えるために使用します。次のように、コレクションエディタで Count 関数を定義できます。



PageFooter の Textbox オブジェクトは個数の出力に使用します。Text プロパティは PageFooter.GenerateScript で設定されます。

```

if (Engine.IsDoublePass)
{
    textBoxFooter.Text = "Records " + dataBand5.GetPageTotal("Count") + " of "
+ dataBand5.GetTotal("Count");
}

```

個数を計算するには、ドキュメントの DoublePass プロパティを True に設定してください。



レポートテンプレート	レポート
<pre>dataBand5:DataBand DataSource = Emp detail6:Detail &lt;dataBand5["EmployeeNam end of dataBand5 pageFooter1:PageFooter</pre>	<p>Nancy Davolio Andrew Fuller Anne Dodsworth Steven Buchana</p> <hr/> <p>Records 4 of 6</p> <p>Laura Callahan Anne Yorke</p> <hr/> <p>Records 2 of 6</p>

## グループ化

条件を指定してデータをグループ化することで、データ行を論理的にグループ化し、情報をより良く表現することができます。GroupBand はそのために使用します。グループ化の基準は、GroupExpression プロパティで設定します。このプロパティはスクリプトエディタで順番に設定されます。

### グループヘッダー

グループヘッダーを表示するには、グループバンドにヘッダーを配置します。グループ値を表示するには、GroupExpression プロパティを使用します。

名前の頭文字ごとに担当者の一覧をグループ化する例：

レポートテンプレート：

dataBand:DataBand DataSource = Emp
groupBand:GroupBand Group = dataB
header:Header
<Group>
detail7:Detail
<dataBand["EmployeeNa
end of groupBand
end of dataBand

```
dataBand.DataSource = Employees; groupBand1.GroupExpression =
dataBand6["EmployeeName"].ToString().Substring(0,1); textBoxHeader.Value = Group;
textBoxDetail.Value = dataBand["EmployeeName"]
```

レポート：

<u>N</u>	Nancy Davolio
<u>A</u>	Andrew Fuller Anne Dodsworth Anne Yorke
<u>S</u>	Steven Buchana
<u>L</u>	Laura Callahan

### グループの並べ替え

GroupBand はレポートの項目をデータソースに保存されている順番で表示します。GroupBand の Order プロパティは出力順を設定します。このプロパティには次の3つの値：KeepOriginal（最初のまま）、Ascending（昇順）、Descending（降順）があります。

Data	
GroupExpression	dataBand["EmployeeName"]
Order	Ascending
SortExpression	
Tag	

SortExpression プロパティは並べ替えの基準を設定します。

社員の名前の頭文字でグループ化し並べ替えをするリストの例：

```
dataBand.DataSource = Employees; groupBand.GroupExpression = dataBand["EmployeeName"].ToString().Substring(0,1); groupBand.Order = Ascending; textBoxHeader.Value = Group; textBoxDetail.Value = dataBand["EmployeeName"]
```

レポートテンプレート	レポート								
dataBand:DataBand DataSource = Emp	<table border="1"> <tr> <td><u>A</u></td> <td>Andrew Fuller Anne Dodsworth Anne Yorke</td> </tr> <tr> <td><u>L</u></td> <td>Laura Callahan</td> </tr> <tr> <td><u>N</u></td> <td>Nancy Davolio</td> </tr> <tr> <td><u>S</u></td> <td>Steven Buchana</td> </tr> </table>	<u>A</u>	Andrew Fuller Anne Dodsworth Anne Yorke	<u>L</u>	Laura Callahan	<u>N</u>	Nancy Davolio	<u>S</u>	Steven Buchana
<u>A</u>		Andrew Fuller Anne Dodsworth Anne Yorke							
<u>L</u>		Laura Callahan							
<u>N</u>		Nancy Davolio							
<u>S</u>		Steven Buchana							
groupBand:GroupBand Group = dataBe									
header:Header									
<Group>									
detail7:Detail									
<dataBand["EmployeeNa									
end of groupBand									
end of dataBand									

### グループのカスケード表示

複雑なグループ構造のレポートを作成するためのオプションがあります。それには、複数の GroupBand を入れ子にして使用します。

まず国名、次に都市名でグループ化された社員リストの例：

```
dataBand.DataSource = Employees; groupBandCounty.GroupExpression = dataBand["Country"];
groupBandCity.GroupExpression = dataBand["City"]; textBoxDetail.Value =
dataBand["EmployeeName"].
```

レポートテンプレート	レポート
dataBand:DataBand DataSource = Employees	
groupBandCounty:GroupBand Group = dataBand["Country"]	<u>Italy</u>
header1:Header	<u>Rome</u>
<Group>	Nancy Davolio
groupBandCity:GroupBand Group = dataBand["City"]	<u>Milan</u>
header2:Header	Anne Dodswort
<Group>	<u>France</u>
detail8:Detail	<u>Paris</u>
<dataBand["Em	Andrew Fuller
end of groupBandCity	
end of groupBandCounty	<u>England</u>
end of dataBand	<u>London</u>
	Steven Buchan
	<u>Liverpool</u>
	Laura Callahan
	Anne Yorke

### グループの合計

グループの合計値を取得するには、グループで GetTotal を呼び出します。グループで GetRunningTotal を呼び出すと、グループの累計値を取得します。

グループ化された数値のリスト例：まず、各グループの合計が計算され、最後に全グループの合計が表示されます。

```
dataBand1.DataSource = Numbers;
```

```
集計関数 : AggregateFunction = Sum; Expression = dataBand1 ["Value"]; Name = SUM;
```

```
groupBand4.GroupExpression = dataBand1["Type"]; textBoxDetail.Value = dataBand1["Value"];
textBoxFooter.Value = GetTotal("SUM");
```

レポートテンプレート	レポート
dataBand1:DataBand DataSource	1
groupBand4:GroupBand Group	13
detail9:Detail	12
<data	group sum 26
footer2:Footer	-34
group sum <Get	-56
end of groupBand4	108
footer3:Footer	group sum 18
total sum <Get	total sum 44
end of dataBand1	

## 水平方向のリストと表

### 水平方向のリスト

水平方向のリストを作成するには、**CrossBand** を使用します。**CrossBand** はこのバンド内のオブジェクトを水平方向（左から右）に反復表示します。**CrossBand** は、レポート内の **CrossBand** の位置を定義する **Detail**、**Header** または **Footer** に入れなければなりません。反復回数は **InstanceCount** プロパティで定義します。

Data	
DataSource	
InstanceCount	5
Tag	

命令文 : `<CrossBand name>. LineNumber()` は、項目番号を取得するために使用します。**CrossBand** に出されるすべての項目が 1 行に収まり切らない場合は、次のページに表示されます。

項目番号を出力する水平方向のリストの例 :

```
crossBand1.InstanceCount = 4;
textBox.Value = crossBand1.LineNumber().ToString + " of " +
crossBand1.InstanceCount().ToString;
```

レポートテンプレート	レポート
<pre> detailCrossBand1 &lt;crossBand CrossBand </pre>	<div style="border: 1px solid black; padding: 5px;"> <span style="border: 1px solid blue; padding: 2px;">1 of 4</span> <span style="border: 1px solid blue; padding: 2px;">2 of 4</span> <span style="border: 1px solid blue; padding: 2px;">3 of 4</span> <span style="border: 1px solid blue; padding: 2px;">4 of 4</span> </div>
<pre> detailCrossBand2 &lt;crossBand CrossBand </pre>	<div style="border: 1px solid black; padding: 5px;"> <span style="border: 1px solid blue; padding: 2px;">1 of 4</span> <span style="border: 1px solid blue; padding: 2px;">2 of 4</span> <span style="border: 1px solid blue; padding: 2px;">3 of 4</span> </div>
	<div style="border: 1px solid black; padding: 5px;"> <span style="border: 1px solid blue; padding: 2px;">4 of 4</span> </div>

### データソースのデータを使った水平方向のリスト

**CrossBand** はデータソースのデータに基づいて水平方向のリストを作成できます。データソースは **DataSource** プロパティに設定します。出力項目数はソースの項目数に依存するので、**InstanceCount** プロパティは無視されます。

Data	
DataSource	Numbers
InstanceCount	0
Tag	

(出力する) 現在のフィールドにアクセスするには、スクリプト : `GetData( "<Data source name>.<Field name>" )` を使用します。

1 行の数値を出力する水平方向のリストの例 :

`crossBand1.DataSource = Numbers; textBox.Value = GetData("Numbers.Value")`

レポートテンプレート	レポート					
<pre> dataBand:DataBand   &lt;Get </pre>	<table border="1"> <tr> <td>1</td> <td>-34</td> <td>13</td> <td>12</td> <td>-56</td> </tr> </table>	1	-34	13	12	-56
1	-34	13	12	-56		

### 表

DataBand に CrossBand を入れて使用することで、データを表として出力することができます。

カレンダーの出力例 :

`dataBand.InstanceCount = 5; crossBand.InstanceCount = 7; textBox.Value = ((crossBand.LineNumber + ((dataBand.LineNumber - 1) * 7)) > 31 ? "" : (crossBand.LineNumber + ((dataBand.LineNumber - 1) * 7)).ToString());`

レポートテンプレート	レポート																																																	
<pre> dataBand:DataBand header:Header   March, 2010   Sun Mon Tue Wed Thu Fri Sat dataBand:DataBand   &lt;cr end of dataBand </pre>	<table border="1"> <thead> <tr> <th colspan="7">March, 2010</th> </tr> <tr> <th>Sun</th> <th>Mon</th> <th>Tue</th> <th>Wed</th> <th>Thu</th> <th>Fri</th> <th>Sat</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td>8</td> <td>9</td> <td>10</td> <td>11</td> <td>12</td> <td>13</td> <td>14</td> </tr> <tr> <td>15</td> <td>16</td> <td>17</td> <td>18</td> <td>19</td> <td>20</td> <td>21</td> </tr> <tr> <td>22</td> <td>23</td> <td>24</td> <td>25</td> <td>26</td> <td>27</td> <td>28</td> </tr> <tr> <td>29</td> <td>30</td> <td>31</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	March, 2010							Sun	Mon	Tue	Wed	Thu	Fri	Sat	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
March, 2010																																																		
Sun	Mon	Tue	Wed	Thu	Fri	Sat																																												
1	2	3	4	5	6	7																																												
8	9	10	11	12	13	14																																												
15	16	17	18	19	20	21																																												
22	23	24	25	26	27	28																																												
29	30	31																																																

### SideBySide バンド

SideBySide はさまざまなバンドのオブジェクトを並列表示するために使用します。SideBySide 内のオブジェクトはすべて、バンドの先頭から出力されます。BandContainer はその中に格納されている複数のセクションを (先頭からではなく) その隣から並列表示するために使用します。

さまざまなバンドから並列表示する例 :

`dataBand1.InstanceCount = 5; dataBand2.InstanceCount = 4; dataBand3.InstanceCount = 2;`

レポートテンプレート	テンプレートの構造	レポート

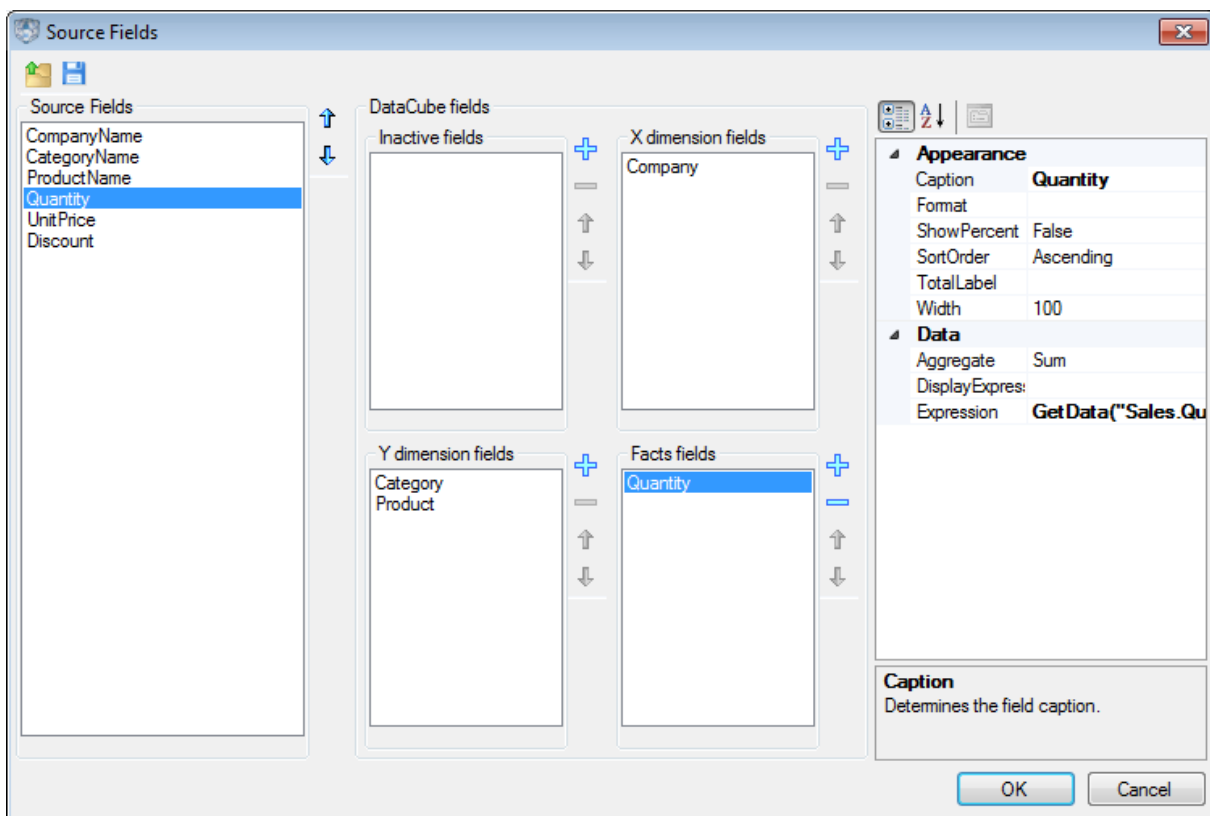
<p>sideBySide1:SideBySide</p> <p>dataBand1:DataBand</p> <p>detail1:Detail</p> <table border="1"> <tr><td>1</td></tr> </table> <p>end of dataBand1</p> <p>dataBand2:DataBand</p> <p>detail2:Detail</p> <table border="1"> <tr><td>2</td></tr> </table> <p>end of dataBand2</p> <p>dataBand3:DataBand</p> <p>detail3:Detail</p> <table border="1"> <tr><td>3</td></tr> </table> <p>end of dataBand3</p>	1	2	3	<p>Document tree</p> <pre> document1 ├── page1 │   ├── sideBySide1 │   │   ├── dataBand1 │   │   │   ├── detail1 │   │   │   ├── dataBand2 │   │   │   │   ├── detail2 │   │   │   └── dataBand3 │   │   │       └── detail3 </pre>	<table border="1"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>1</td><td>2</td><td></td></tr> <tr><td>1</td><td>2</td><td></td></tr> <tr><td>1</td><td></td><td></td></tr> </table>	1	2	3	1	2	3	1	2		1	2		1					
1																							
2																							
3																							
1	2	3																					
1	2	3																					
1	2																						
1	2																						
1																							
<p>sideBySide1:SideBySide</p> <p>dataBand1:DataBand</p> <p>detail1:Detail</p> <table border="1"> <tr><td>1</td></tr> </table> <p>end of dataBand1</p> <p>bandContainer1:BandContainer</p> <p>dataBand2:DataBand</p> <p>detail2:Detail</p> <table border="1"> <tr><td>2</td></tr> </table> <p>end of dataBand2</p> <p>dataBand3:DataBand</p> <p>detail3:Detail</p> <table border="1"> <tr><td>3</td></tr> </table> <p>end of dataBand3</p>	1	2	3	<p>Document tree</p> <pre> document1 ├── page1 │   ├── sideBySide1 │   │   ├── dataBand1 │   │   │   └── detail1 │   │   └── bandContainer1 │   │       ├── dataBand2 │   │       │   ├── detail2 │   │       └── dataBand3 │   │           └── detail3 </pre>	<table border="1"> <tr><td>1</td><td>2</td><td></td></tr> <tr><td>1</td><td>2</td><td></td></tr> <tr><td>1</td><td>2</td><td></td></tr> <tr><td>1</td><td>2</td><td></td></tr> <tr><td>1</td><td></td><td>3</td></tr> <tr><td></td><td></td><td>3</td></tr> </table>	1	2		1	2		1	2		1	2		1		3			3
1																							
2																							
3																							
1	2																						
1	2																						
1	2																						
1	2																						
1		3																					
		3																					

## ピボットテーブル

ピボットテーブルは、データ出力を構築し、合計を計算するために使用されます。PivotTable はピボットテーブルの作成に使用します。DataSource プロパティは、ピボットテーブルのデータソースを設定します。

Data	
DataSource	Sales
Layout	
NestedFields	(Collection)
Tag	

Layout プロパティは処理後のピボットテーブルの行や列の値を設定します。このプロパティは、ソースフィールドで変更できます。



データソースのフィールド一覧は、左側のソースフィールドに表示されます。データキューブ フィールドは、ピボットテーブルのデータを割り当てます。ピボットテーブルの水平方向の見出しのフィールドは、X デイメンションフィールドに設定します。ピボットテーブルの垂直方向の見出しのフィールドは、Y デイメンションフィールドに設定します。ファクトフィールドには、データを持ったセルのフィールドが格納されます。プロパティグリッドは右側に表示され、セルのプロパティを設定するために使用します。

Appearance	
Caption	ProductName
Format	
ShowPercent	False
SortOrder	Ascending
TotalLabel	
Width	100
Data	
Aggregate	Sum
DisplayExpressi	
Expression	GetData("Sales.Pro

### テーブルのタイトル

TableCaption プロパティはテーブルのタイトルを設定します。

プロパティ		レポート				
StyleName	New Style					
Styles		Company				
TableCaption		Category	Product	Futterkiste	Karkki Oy	Total
ToolTip		Beverages	Chai	452.63	909.00	452.63
	Ipoh Coffee			909.00	909.00	
	Stout		420.00		420.00	





StyleName	New Style			Sales		
⊕ Styles				Company		
TableCaption	<b>Sales</b>	Category	Product	Futterkiste	Karkki Oy	Total
ToolTip		Beverages	Chai	452.625		452.625
			Ipoh Coffee		909	909
			Stout	420		420

テーブルの行タイトルや列タイトルは、ピボットテーブルの **Layout** プロパティのエディタで設定できます。  
**Caption** プロパティはそのために使用します。**TotalLabel** プロパティは合計値を持つ列や行のタイトルを設定します。このプロパティが設定されていない場合、合計値は「合計」として識別されます。

プロパティ	レポート																												
<b>Appearance</b> Caption <b>CompanyName</b> Format ShowPercent False SortOrder Ascending TotalLabel Width 100	<table border="1"> <thead> <tr> <th colspan="4">Sales</th> </tr> <tr> <th colspan="4">Company</th> </tr> <tr> <th>Product</th> <th>Futterkiste</th> <th>Karkki Oy</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>Chai</td> <td>452.63</td> <td></td> <td>452.63</td> </tr> <tr> <td>Ipoh Coffee</td> <td></td> <td>909.00</td> <td>909.00</td> </tr> <tr> <td>Stout</td> <td>420.00</td> <td></td> <td>420.00</td> </tr> <tr> <td><b>Total</b></td> <td><b>872.63</b></td> <td><b>909.00</b></td> <td><b>1781.63</b></td> </tr> </tbody> </table>	Sales				Company				Product	Futterkiste	Karkki Oy	Total	Chai	452.63		452.63	Ipoh Coffee		909.00	909.00	Stout	420.00		420.00	<b>Total</b>	<b>872.63</b>	<b>909.00</b>	<b>1781.63</b>
Sales																													
Company																													
Product	Futterkiste	Karkki Oy	Total																										
Chai	452.63		452.63																										
Ipoh Coffee		909.00	909.00																										
Stout	420.00		420.00																										
<b>Total</b>	<b>872.63</b>	<b>909.00</b>	<b>1781.63</b>																										
<b>Appearance</b> Caption <b>Company</b> Format ShowPercent False SortOrder Ascending TotalLabel <b>Sum</b> Width 100	<table border="1"> <thead> <tr> <th colspan="4">Sales</th> </tr> <tr> <th colspan="4">Company</th> </tr> <tr> <th>Product</th> <th>Futterkiste</th> <th>Karkki Oy</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>Chai</td> <td>452.63</td> <td></td> <td>452.63</td> </tr> <tr> <td>Ipoh Coffee</td> <td></td> <td>909.00</td> <td>909.00</td> </tr> <tr> <td>Stout</td> <td>420.00</td> <td></td> <td>420.00</td> </tr> <tr> <td><b>Total</b></td> <td><b>872.63</b></td> <td><b>909.00</b></td> <td><b>1781.63</b></td> </tr> </tbody> </table>	Sales				Company				Product	Futterkiste	Karkki Oy	Sum	Chai	452.63		452.63	Ipoh Coffee		909.00	909.00	Stout	420.00		420.00	<b>Total</b>	<b>872.63</b>	<b>909.00</b>	<b>1781.63</b>
Sales																													
Company																													
Product	Futterkiste	Karkki Oy	Sum																										
Chai	452.63		452.63																										
Ipoh Coffee		909.00	909.00																										
Stout	420.00		420.00																										
<b>Total</b>	<b>872.63</b>	<b>909.00</b>	<b>1781.63</b>																										

ピボットテーブルの **DimensionOnEveryPage** プロパティは、ピボットテーブルのヘッダーを各ページに出力するかを設定します。

プロパティ	レポート																																																				
<b>Appearance</b> AutoColumnWidth True AutoRowHeight <b>None</b> DimensionOnEveryPage False RowHeight <b>70</b>	<table border="1"> <thead> <tr> <th colspan="2"></th> <th colspan="3">Sales</th> </tr> <tr> <th colspan="2"></th> <th colspan="3">Company</th> </tr> <tr> <th>Category</th> <th>Product</th> <th>Futterkiste</th> <th>Karkki Oy</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td rowspan="2">Beverages</td> <td>Chai</td> <td>452.63</td> <td></td> <td>452.63</td> </tr> <tr> <td>Ipoh Coffee</td> <td></td> <td>909.00</td> <td>909.00</td> </tr> <tr> <td colspan="5"><hr/></td> </tr> <tr> <td rowspan="2">Beverages</td> <td>Stout</td> <td>420.00</td> <td></td> <td>420.00</td> </tr> <tr> <td><b>Total</b></td> <td><b>872.63</b></td> <td><b>909.00</b></td> <td><b>1781.63</b></td> </tr> <tr> <td rowspan="2">Dairy Products</td> <td>Courdavault</td> <td>890.63</td> <td></td> <td>890.63</td> </tr> <tr> <td><b>Total</b></td> <td><b>890.63</b></td> <td></td> <td><b>890.63</b></td> </tr> <tr> <td colspan="2"><b>Total</b></td> <td><b>1763.25</b></td> <td><b>909.00</b></td> <td><b>2672.25</b></td> </tr> </tbody> </table>			Sales					Company			Category	Product	Futterkiste	Karkki Oy	Total	Beverages	Chai	452.63		452.63	Ipoh Coffee		909.00	909.00	<hr/>					Beverages	Stout	420.00		420.00	<b>Total</b>	<b>872.63</b>	<b>909.00</b>	<b>1781.63</b>	Dairy Products	Courdavault	890.63		890.63	<b>Total</b>	<b>890.63</b>		<b>890.63</b>	<b>Total</b>		<b>1763.25</b>	<b>909.00</b>	<b>2672.25</b>
		Sales																																																			
		Company																																																			
Category	Product	Futterkiste	Karkki Oy	Total																																																	
Beverages	Chai	452.63		452.63																																																	
	Ipoh Coffee		909.00	909.00																																																	
<hr/>																																																					
Beverages	Stout	420.00		420.00																																																	
	<b>Total</b>	<b>872.63</b>	<b>909.00</b>	<b>1781.63</b>																																																	
Dairy Products	Courdavault	890.63		890.63																																																	
	<b>Total</b>	<b>890.63</b>		<b>890.63</b>																																																	
<b>Total</b>		<b>1763.25</b>	<b>909.00</b>	<b>2672.25</b>																																																	



<b>Appearance</b>					
AutoColumnWidth	True				
AutoRowHeight	<b>None</b>				
DimensionOnEveryPage	<b>True</b>				
RowHeight	<b>70</b>				

		Sales		
		Company		
Category	Product	Futterkiste	Karkki Oy	Total
Beverages	Chai	452.63		452.63
	Ipoh Coffee		909.00	909.00

---

		Sales		
		Company		
Category	Product	Futterkiste	Karkki Oy	Total
Beverages	Stout	420.00		420.00
	Total	872.63	909.00	1781.63

ShowSingleFact プロパティを使用して、ファクト列のタイトルを表示するかを設定できます。

プロパティ		レポート			
Shift	0 cm				
ShowRepeatText	True				
ShowSingleFact	False				
ShowTableCaption	True				

		Sales		
		Company		
Category	Product	Futterkiste	Total	
Beverages	Chai	452.63	452.63	
	Stout	420.00	420.00	

		Sales		
		Company		
Category	Product	Price	Price	
Beverages	Chai	452.63	452.63	
	Stout	420.00	420.00	

現在のページに文字列が収まり切らない場合は、次のページにテキストを反復させることができます。それには、ShowRepeatText プロパティを使用してください。

プロパティ		レポート		
RowHeight	30			
Shift	0 cm			
ShowRepeatText	True			
ShowSingleFact	False			

Dairy Products	Courdavault	890.63	
	Flotemysost	2256.00	
	Geitost	154.84	

---

Dairy Products	Total	3301.47	
----------------	-------	---------	--

Dairy Products	Courdavault	890.63	
	Flotemysost	2256.00	
	Geitost	154.84	

---

	Total	3301.47	
--	-------	---------	--

## データ形式

「ソースフィールド」エディタの Format プロパティを使用して、出力データの形式を定義できます。



プロパティ	レポート																					
<b>Appearance</b> Caption <b>Price</b> Format ShowPercent False SortOrder Ascending	<table border="1"> <tr><td>452.625</td><td></td><td>452.625</td></tr> <tr><td></td><td>909</td><td>909</td></tr> <tr><td>420</td><td></td><td>420</td></tr> <tr><td>872.625</td><td>909</td><td>1781.625</td></tr> <tr><td>890.625</td><td></td><td>890.625</td></tr> <tr><td>890.625</td><td></td><td>890.625</td></tr> <tr><td>1763.25</td><td>909</td><td>2672.25</td></tr> </table>	452.625		452.625		909	909	420		420	872.625	909	1781.625	890.625		890.625	890.625		890.625	1763.25	909	2672.25
452.625		452.625																				
	909	909																				
420		420																				
872.625	909	1781.625																				
890.625		890.625																				
890.625		890.625																				
1763.25	909	2672.25																				
<b>Appearance</b> Caption <b>Price</b> Format <b>0.00</b> ShowPercent False SortOrder Ascending	<table border="1"> <tr><td>452.63</td><td></td><td>452.63</td></tr> <tr><td></td><td>909.00</td><td>909.00</td></tr> <tr><td>420.00</td><td></td><td>420.00</td></tr> <tr><td>872.63</td><td>909.00</td><td>1781.63</td></tr> <tr><td>890.63</td><td></td><td>890.63</td></tr> <tr><td>890.63</td><td></td><td>890.63</td></tr> <tr><td>1763.25</td><td>909.00</td><td>2672.25</td></tr> </table>	452.63		452.63		909.00	909.00	420.00		420.00	872.63	909.00	1781.63	890.63		890.63	890.63		890.63	1763.25	909.00	2672.25
452.63		452.63																				
	909.00	909.00																				
420.00		420.00																				
872.63	909.00	1781.63																				
890.63		890.63																				
890.63		890.63																				
1763.25	909.00	2672.25																				
<b>Appearance</b> Caption <b>Price</b> Format <b>0.00</b> ShowPercent <b>True</b> ▼ SortOrder Ascending	<table border="1"> <tr><td>16.94%</td><td></td><td>16.94%</td></tr> <tr><td></td><td>34.02%</td><td>34.02%</td></tr> <tr><td>15.72%</td><td></td><td>15.72%</td></tr> <tr><td>32.66%</td><td>34.02%</td><td>66.67%</td></tr> <tr><td>33.33%</td><td></td><td>33.33%</td></tr> <tr><td>33.33%</td><td></td><td>33.33%</td></tr> <tr><td>65.98%</td><td>34.02%</td><td>100.00%</td></tr> </table>	16.94%		16.94%		34.02%	34.02%	15.72%		15.72%	32.66%	34.02%	66.67%	33.33%		33.33%	33.33%		33.33%	65.98%	34.02%	100.00%
16.94%		16.94%																				
	34.02%	34.02%																				
15.72%		15.72%																				
32.66%	34.02%	66.67%																				
33.33%		33.33%																				
33.33%		33.33%																				
65.98%	34.02%	100.00%																				

ShowZeroValues プロパティはレポートに NULL 値を表示します。

プロパティ	レポート									
ShowSingleFact False ShowTableCaption True ShowZeroValues False ▼ StyleName <b>Main</b>	<table border="1"> <tr><td>452.63</td><td></td><td>452.63</td></tr> <tr><td></td><td>909.00</td><td>909.00</td></tr> <tr><td>420.00</td><td></td><td>420.00</td></tr> </table>	452.63		452.63		909.00	909.00	420.00		420.00
452.63		452.63								
	909.00	909.00								
420.00		420.00								
ShowSingleFact False ShowTableCaption True ShowZeroValues <b>True</b> ▼ StyleName <b>Main</b>	<table border="1"> <tr><td>452.63</td><td>0</td><td>452.63</td></tr> <tr><td>0</td><td>909.00</td><td>909.00</td></tr> <tr><td>420.00</td><td>0</td><td>420.00</td></tr> </table>	452.63	0	452.63	0	909.00	909.00	420.00	0	420.00
452.63	0	452.63								
0	909.00	909.00								
420.00	0	420.00								

### 合計の表示/非表示

フィールドの **ShowTotal** プロパティを使用して、ピボットテーブルの「合計」列または行を非表示にすることができます。「ソースフィールド」エディタでこのプロパティを設定できます。

プロパティ	レポート																				
<b>Appearance</b> Caption <b>Category</b> Format ShowPercent False <b>ShowTotal</b> True ▼ SortOrder Ascending	<table border="1"> <thead> <tr> <th>Category</th> <th>Product</th> <th>Alfreds Futterkiste</th> </tr> </thead> <tbody> <tr> <td rowspan="4">Beverages</td> <td>Chai</td> <td>505,88</td> </tr> <tr> <td>Ipoh Coffee</td> <td></td> </tr> <tr> <td>Steeleye Stout</td> <td>525,00</td> </tr> <tr> <td>Total</td> <td>1030,88</td> </tr> <tr> <td rowspan="2">Confections</td> <td>Maxilaku</td> <td>130,00</td> </tr> <tr> <td>Total</td> <td>130,00</td> </tr> <tr> <td>Total</td> <td></td> <td>4509,22</td> </tr> </tbody> </table>	Category	Product	Alfreds Futterkiste	Beverages	Chai	505,88	Ipoh Coffee		Steeleye Stout	525,00	Total	1030,88	Confections	Maxilaku	130,00	Total	130,00	Total		4509,22
Category	Product	Alfreds Futterkiste																			
Beverages	Chai	505,88																			
	Ipoh Coffee																				
	Steeleye Stout	525,00																			
	Total	1030,88																			
Confections	Maxilaku	130,00																			
	Total	130,00																			
Total		4509,22																			

<b>Appearance</b> Caption <b>Category</b> Format ShowPercent False ShowTotal <b>False</b> ▼ SortOrder Ascending		<table border="1"> <thead> <tr> <th>Category</th> <th>Product</th> <th>Alfreds Futterkiste</th> </tr> </thead> <tbody> <tr> <td rowspan="3">Beverages</td> <td>Chai</td> <td>505,88</td> </tr> <tr> <td>Ipoh Coffee</td> <td></td> </tr> <tr> <td>Steeleye Stout</td> <td>525,00</td> </tr> <tr> <td>Confections</td> <td>Maxilaku</td> <td>130,00</td> </tr> <tr> <td colspan="2">Total</td> <td>4509,22</td> </tr> </tbody> </table>	Category	Product	Alfreds Futterkiste	Beverages	Chai	505,88	Ipoh Coffee		Steeleye Stout	525,00	Confections	Maxilaku	130,00	Total		4509,22
Category	Product	Alfreds Futterkiste																
Beverages	Chai	505,88																
	Ipoh Coffee																	
	Steeleye Stout	525,00																
Confections	Maxilaku	130,00																
Total		4509,22																

### データの並べ替え

並べ替え順は **SortOrder** プロパティで設定します。このプロパティの値には **Ascending**（昇順）と **Descending**（降順）があります。

### データのフィルタ処理

フィルタ処理は、ある特定条件を指定してデータを出力しなければならない場合に使用します。

**FilterExpression** プロパティは、その条件を設定するために使用します。式はスクリプトで設定します。この式の計算結果がピボットテーブルのデータを決めます。

プロパティ	レポート					
設定なし			Futterkiste	Karkki Oy	Shopy	Total
	Category	Product	Discount	Discount	Discount	Discount
	Beverages	Chai	0.15	0.04		0.19
		Ipoh Coffee		0.1		0.1
		Stout	0.2			0.2
		Total	0.35	0.14		0.49
FilterExpression="(double)GetData("Sales.Discount") >= 0.1"			Futterkiste	Karkki Oy	Total	
	Category	Product	Discount	Discount	Discount	
	Beverages	Chai	0.15		0.15	
		Ipoh Coffee		0.1	0.1	
		Stout	0.2		0.2	
		Total	0.35	0.1	0.45	

### テーブルのサイズ

ピボットテーブルのセルサイズは、規定ではピボットテーブルのデータの大きさに応じて自動的に決まります。このために、**AutoColumnWidth**（列の幅を設定します）と **AutoRowHeight**（行の高さを設定します）プロパティを使用します。行の高さを任意に設定するには、**AutoRowHeight** プロパティの値を「None」に選択し、**RowHeight** プロパティにピクセル単位で行の高さを設定してください。

<b>Appearance</b>	
AutoColumnWidth	True
AutoRowHeight	<b>None</b>
DimensionOnEveryPage	False
RowHeight	<b>100</b>

左側の余白は **Shift** プロパティで設定します。

### テーブルのスタイル

**StyleName** プロパティは、テーブルのスタイルを設定します。スタイルを作成するには、スタイルシートエディタを使用します。

**Styles** のプロパティ群を使用して、各セル型に個別のスタイルを設定できます。

プロパティ	レポート																																																																								
<table border="1"> <tr><td>StyleName</td><td></td></tr> <tr><td>Styles</td><td></td></tr> <tr><td>EvenRowFactStyle</td><td></td></tr> <tr><td>FieldCaptionStyle</td><td></td></tr> <tr><td>GroupTotalStyle</td><td></td></tr> <tr><td>MainTotalStyle</td><td></td></tr> <tr><td>OddRowFactStyle</td><td></td></tr> <tr><td>TableCaptionStyle</td><td></td></tr> <tr><td>TotalStyle</td><td></td></tr> <tr><td>XDimensionStyle</td><td></td></tr> <tr><td>YDimensionStyle</td><td></td></tr> </table>	StyleName		Styles		EvenRowFactStyle		FieldCaptionStyle		GroupTotalStyle		MainTotalStyle		OddRowFactStyle		TableCaptionStyle		TotalStyle		XDimensionStyle		YDimensionStyle		<table border="1"> <tr><td colspan="5">Sales</td></tr> <tr><td colspan="5">Company</td></tr> <tr><td>Category</td><td>Product</td><td>Futterkiste</td><td>Karkki Oy</td><td>Total</td></tr> <tr><td></td><td>Chai</td><td>452.63</td><td></td><td>452.63</td></tr> <tr><td>Beverages</td><td>Ipoh Coffee</td><td></td><td>909.00</td><td>909.00</td></tr> <tr><td></td><td>Stout</td><td>420.00</td><td></td><td>420.00</td></tr> <tr><td></td><td>Total</td><td>872.63</td><td>909.00</td><td>1781.63</td></tr> <tr><td>Dairy Products</td><td>Courdavault</td><td>890.63</td><td></td><td>890.63</td></tr> <tr><td></td><td>Total</td><td>890.63</td><td></td><td>890.63</td></tr> <tr><td>Total</td><td></td><td>1763.25</td><td>909.00</td><td>2672.25</td></tr> </table>	Sales					Company					Category	Product	Futterkiste	Karkki Oy	Total		Chai	452.63		452.63	Beverages	Ipoh Coffee		909.00	909.00		Stout	420.00		420.00		Total	872.63	909.00	1781.63	Dairy Products	Courdavault	890.63		890.63		Total	890.63		890.63	Total		1763.25	909.00	2672.25
StyleName																																																																									
Styles																																																																									
EvenRowFactStyle																																																																									
FieldCaptionStyle																																																																									
GroupTotalStyle																																																																									
MainTotalStyle																																																																									
OddRowFactStyle																																																																									
TableCaptionStyle																																																																									
TotalStyle																																																																									
XDimensionStyle																																																																									
YDimensionStyle																																																																									
Sales																																																																									
Company																																																																									
Category	Product	Futterkiste	Karkki Oy	Total																																																																					
	Chai	452.63		452.63																																																																					
Beverages	Ipoh Coffee		909.00	909.00																																																																					
	Stout	420.00		420.00																																																																					
	Total	872.63	909.00	1781.63																																																																					
Dairy Products	Courdavault	890.63		890.63																																																																					
	Total	890.63		890.63																																																																					
Total		1763.25	909.00	2672.25																																																																					
<table border="1"> <tr><td>StyleName</td><td>Main</td></tr> <tr><td>Styles</td><td></td></tr> <tr><td>EvenRowFactStyle</td><td></td></tr> <tr><td>FieldCaptionStyle</td><td></td></tr> <tr><td>GroupTotalStyle</td><td></td></tr> <tr><td>MainTotalStyle</td><td></td></tr> <tr><td>OddRowFactStyle</td><td></td></tr> <tr><td>TableCaptionStyle</td><td></td></tr> <tr><td>TotalStyle</td><td></td></tr> <tr><td>XDimensionStyle</td><td></td></tr> <tr><td>YDimensionStyle</td><td></td></tr> </table>	StyleName	Main	Styles		EvenRowFactStyle		FieldCaptionStyle		GroupTotalStyle		MainTotalStyle		OddRowFactStyle		TableCaptionStyle		TotalStyle		XDimensionStyle		YDimensionStyle		<table border="1"> <tr><td colspan="2"></td><td colspan="3">Sales</td></tr> <tr><td colspan="2"></td><td colspan="3">Company</td></tr> <tr><td>Category</td><td>Product</td><td>Futterkiste</td><td>Karkki Oy</td><td>Total</td></tr> <tr><td></td><td>Chai</td><td>452.63</td><td></td><td>452.63</td></tr> <tr><td>Beverages</td><td>Ipoh Coffee</td><td></td><td>909.00</td><td>909.00</td></tr> <tr><td></td><td>Stout</td><td>420.00</td><td></td><td>420.00</td></tr> <tr><td></td><td>Total</td><td>872.63</td><td>909.00</td><td>1781.63</td></tr> <tr><td>Dairy Products</td><td>Courdavault</td><td>890.63</td><td></td><td>890.63</td></tr> <tr><td></td><td>Total</td><td>890.63</td><td></td><td>890.63</td></tr> <tr><td>Total</td><td></td><td>1763.25</td><td>909.00</td><td>2672.25</td></tr> </table>			Sales					Company			Category	Product	Futterkiste	Karkki Oy	Total		Chai	452.63		452.63	Beverages	Ipoh Coffee		909.00	909.00		Stout	420.00		420.00		Total	872.63	909.00	1781.63	Dairy Products	Courdavault	890.63		890.63		Total	890.63		890.63	Total		1763.25	909.00	2672.25
StyleName	Main																																																																								
Styles																																																																									
EvenRowFactStyle																																																																									
FieldCaptionStyle																																																																									
GroupTotalStyle																																																																									
MainTotalStyle																																																																									
OddRowFactStyle																																																																									
TableCaptionStyle																																																																									
TotalStyle																																																																									
XDimensionStyle																																																																									
YDimensionStyle																																																																									
		Sales																																																																							
		Company																																																																							
Category	Product	Futterkiste	Karkki Oy	Total																																																																					
	Chai	452.63		452.63																																																																					
Beverages	Ipoh Coffee		909.00	909.00																																																																					
	Stout	420.00		420.00																																																																					
	Total	872.63	909.00	1781.63																																																																					
Dairy Products	Courdavault	890.63		890.63																																																																					
	Total	890.63		890.63																																																																					
Total		1763.25	909.00	2672.25																																																																					
<table border="1"> <tr><td>StyleName</td><td>Main</td></tr> <tr><td>Styles</td><td></td></tr> <tr><td>EvenRowFactStyle</td><td>DataStyle</td></tr> <tr><td>FieldCaptionStyle</td><td>CaptionStyle</td></tr> <tr><td>GroupTotalStyle</td><td></td></tr> <tr><td>MainTotalStyle</td><td>Total Style</td></tr> <tr><td>OddRowFactStyle</td><td>DataStyle</td></tr> <tr><td>TableCaptionStyle</td><td>CaptionStyle</td></tr> <tr><td>TotalStyle</td><td>Total Style</td></tr> <tr><td>XDimensionStyle</td><td></td></tr> <tr><td>YDimensionStyle</td><td></td></tr> </table>	StyleName	Main	Styles		EvenRowFactStyle	DataStyle	FieldCaptionStyle	CaptionStyle	GroupTotalStyle		MainTotalStyle	Total Style	OddRowFactStyle	DataStyle	TableCaptionStyle	CaptionStyle	TotalStyle	Total Style	XDimensionStyle		YDimensionStyle		<table border="1"> <tr><td colspan="2"></td><td colspan="3">Sales</td></tr> <tr><td colspan="2"></td><td colspan="3">Company</td></tr> <tr><td>Category</td><td>Product</td><td>Futterkiste</td><td>Karkki Oy</td><td>Total</td></tr> <tr><td></td><td>Chai</td><td>452.63</td><td></td><td>452.63</td></tr> <tr><td>Beverages</td><td>Ipoh Coffee</td><td></td><td>909.00</td><td>909.00</td></tr> <tr><td></td><td>Stout</td><td>420.00</td><td></td><td>420.00</td></tr> <tr><td></td><td>Total</td><td>872.63</td><td>909.00</td><td>1781.63</td></tr> <tr><td>Dairy Products</td><td>Courdavault</td><td>890.63</td><td></td><td>890.63</td></tr> <tr><td></td><td>Total</td><td>890.63</td><td></td><td>890.63</td></tr> <tr><td>Total</td><td></td><td>1763.25</td><td>909.00</td><td>2672.25</td></tr> </table>			Sales					Company			Category	Product	Futterkiste	Karkki Oy	Total		Chai	452.63		452.63	Beverages	Ipoh Coffee		909.00	909.00		Stout	420.00		420.00		Total	872.63	909.00	1781.63	Dairy Products	Courdavault	890.63		890.63		Total	890.63		890.63	Total		1763.25	909.00	2672.25
StyleName	Main																																																																								
Styles																																																																									
EvenRowFactStyle	DataStyle																																																																								
FieldCaptionStyle	CaptionStyle																																																																								
GroupTotalStyle																																																																									
MainTotalStyle	Total Style																																																																								
OddRowFactStyle	DataStyle																																																																								
TableCaptionStyle	CaptionStyle																																																																								
TotalStyle	Total Style																																																																								
XDimensionStyle																																																																									
YDimensionStyle																																																																									
		Sales																																																																							
		Company																																																																							
Category	Product	Futterkiste	Karkki Oy	Total																																																																					
	Chai	452.63		452.63																																																																					
Beverages	Ipoh Coffee		909.00	909.00																																																																					
	Stout	420.00		420.00																																																																					
	Total	872.63	909.00	1781.63																																																																					
Dairy Products	Courdavault	890.63		890.63																																																																					
	Total	890.63		890.63																																																																					
Total		1763.25	909.00	2672.25																																																																					

### ピボットテーブルの保存

**Layout** プロパティはピボットテーブルのフィールドの式や位置を設定します。後で使用するためにレイアウトをファイルに保存して、最初のピボットテーブルのフォームに瞬時に戻ることができます。**Layout** プロパティのエディタの[保存]/[開く]ボタンを使用して、ピボットテーブルのパラメータの保存/読込ができます。ファイルは XML 形式で、拡張子は .dcl です。

### イメージ

**Picture** オブジェクトは、イメージを表示するために使用します。

**Image** プロパティはイメージを設定します。ファイルを選択するには、**Image** プロパティのボタンを押すか、**Picture** オブジェクトを左マウスボタンでダブルクリックしてください。

アップロード可能なイメージ形式 : \*.bmp、\*.gif、\*.jpg、\*.jpeg、\*.png、\*.ico、\*.emf、\*.wmf

### イメージのサイズとスケーリング

**Size** プロパティは **Picture** オブジェクトのサイズを設定します。出力時にイメージのサイズが **Picture** オブジェクトのサイズに収まりきらない場合、**SizeMode** プロパティを使用してサイズを変更できます。

イメージの引き伸ばし - オブジェクトのサイズに応じてイメージの比率を変更します。

プロパティ	レポート								
<table border="1"> <tr><td>Margins</td><td>0; 0; 0; 0 cm</td></tr> <tr><td>SizeMode</td><td>Stretch</td></tr> <tr><td>StyleName</td><td></td></tr> <tr><td>ToolTip</td><td></td></tr> </table>	Margins	0; 0; 0; 0 cm	SizeMode	Stretch	StyleName		ToolTip		
Margins	0; 0; 0; 0 cm								
SizeMode	Stretch								
StyleName									
ToolTip									
イメージの切り取り - イメージは実サイズのままで、オブジェクトのサイズに応じて切り取られます。									
<table border="1"> <tr><td>Margins</td><td>0; 0; 0; 0 cm</td></tr> <tr><td>SizeMode</td><td>Normal</td></tr> <tr><td>StyleName</td><td></td></tr> <tr><td>ToolTip</td><td></td></tr> </table>	Margins	0; 0; 0; 0 cm	SizeMode	Normal	StyleName		ToolTip		
Margins	0; 0; 0; 0 cm								
SizeMode	Normal								
StyleName									
ToolTip									
自動サイズ - オブジェクトのサイズはイメージサイズに応じて変更され、編集はされません。									
<table border="1"> <tr><td>Margins</td><td>0; 0; 0; 0 cm</td></tr> <tr><td>SizeMode</td><td>AutoSize</td></tr> <tr><td>StyleName</td><td></td></tr> <tr><td>ToolTip</td><td></td></tr> </table>	Margins	0; 0; 0; 0 cm	SizeMode	AutoSize	StyleName		ToolTip		
Margins	0; 0; 0; 0 cm								
SizeMode	AutoSize								
StyleName									
ToolTip									
均一 - イメージはオブジェクトの領域内の最大サイズにリサイズされ、イメージのアスペクト比は維持されます。									
<table border="1"> <tr><td>Margins</td><td>0; 0; 0; 0 cm</td></tr> <tr><td>SizeMode</td><td>Uniform</td></tr> <tr><td>StyleName</td><td></td></tr> <tr><td>ToolTip</td><td></td></tr> </table>	Margins	0; 0; 0; 0 cm	SizeMode	Uniform	StyleName		ToolTip		
Margins	0; 0; 0; 0 cm								
SizeMode	Uniform								
StyleName									
ToolTip									

### イメージの位置合わせ



SizeMode プロパティを Normal か Uniform に設定した場合、ImageAlign プロパティは Picture オブジェクトの領域内のイメージの位置を定義します。

プロパティ	レポート										
<table border="1"> <tr><td>Fill</td><td>[None]</td></tr> <tr><td>Image</td><td> System.Drawing.Bitmap</td></tr> <tr><td>ImageAlign</td><td>TopLeft</td></tr> <tr><td>Margins</td><td>0; 0; 0; 0 cm</td></tr> <tr><td>SizeMode</td><td>Normal</td></tr> </table>	Fill	[None]	Image	System.Drawing.Bitmap	ImageAlign	TopLeft	Margins	0; 0; 0; 0 cm	SizeMode	Normal	
Fill	[None]										
Image	System.Drawing.Bitmap										
ImageAlign	TopLeft										
Margins	0; 0; 0; 0 cm										
SizeMode	Normal										
<table border="1"> <tr><td>Image</td><td> System.Drawing.Bitmap</td></tr> <tr><td>ImageAlign</td><td>MiddleCenter</td></tr> <tr><td>Margins</td><td>0; 0; 0; 0 cm</td></tr> <tr><td>SizeMode</td><td>Normal</td></tr> <tr><td>StyleName</td><td></td></tr> </table>	Image	System.Drawing.Bitmap	ImageAlign	MiddleCenter	Margins	0; 0; 0; 0 cm	SizeMode	Normal	StyleName		
Image	System.Drawing.Bitmap										
ImageAlign	MiddleCenter										
Margins	0; 0; 0; 0 cm										
SizeMode	Normal										
StyleName											
<table border="1"> <tr><td>Image</td><td> System.Drawing.Bitmap</td></tr> <tr><td>ImageAlign</td><td>BottomRight</td></tr> <tr><td>Margins</td><td>0; 0; 0; 0 cm</td></tr> <tr><td>SizeMode</td><td>Normal</td></tr> <tr><td>StyleName</td><td></td></tr> </table>	Image	System.Drawing.Bitmap	ImageAlign	BottomRight	Margins	0; 0; 0; 0 cm	SizeMode	Normal	StyleName		
Image	System.Drawing.Bitmap										
ImageAlign	BottomRight										
Margins	0; 0; 0; 0 cm										
SizeMode	Normal										
StyleName											

### イメージの動的な読込

レポートの生成中にイメージを読み込まなければならない場合、バインドのプロパティを開いて Image プロパティを使用できます。ファイルからイメージを読み込む場合、次のスクリプトを使用します。



```
System.Drawing.Image.FromFile("<path to the file>")
```

レポートテンプレート	レポート
<pre>dataBand:DataBand DataSource = Employees detail:Detail &lt;dataBand["EmployeeName"]&gt; end of dataBand</pre>	<p><u>Nancy Davolio</u></p> 
<pre>dataBand.DataSource = Employees; textBox.Value = dataBand["EmployeeName"]; picture.Image = System.Drawing.Image.FromFile ("C:\\Data\\Pictures\\"+dataBand["EmployeeName"]. ToString()+".png"); picture.SizeMode = AutoSize;</pre>	<p><u>Andrew Fuller</u></p> 

### データソースのイメージ

データソースからイメージを出力するには、データソースが設定されている **DataBand** 内の **Detail** に **Picture** オブジェクトを配置し、「バインド」タブで **Image** プロパティを設定します。この場合、イメージはデータソースにバイト型で格納されます。



レポートテンプレート	レポート
<pre>dataBand:DataBand DataSource = Employee detail1:Detail &lt;dataBand["EmployeeName"]&gt; end of dataBand</pre>	<p>Nancy Davolio</p>  <p>Andrew Fuller</p> 
<pre>dataBand.DataSource = Employees; textBox.Value = dataBand["EmployeeName"]; picture.Image = dataBand["Picture"]; picture.SizeMode = AutoSize;</pre>	

## シェープの使用

Shape コンポーネントを使用して、様々なグラフィカルな図形を出力できます。

### シェープの種類

ShapeStyle プロパティは図形の種類を定義します。次の種類を使用できます。

シェープのスタイル	レポート														
<p>RectangleShape - 四角形を描画します</p> <table border="1"> <tr><td colspan="2">Appearance</td></tr> <tr><td>Fill</td><td>HatchFill</td></tr> <tr><td>Line</td><td>2 Solid DodgerBlue</td></tr> <tr><td>Shadow</td><td>Shadow</td></tr> <tr><td>ShapeStyle</td><td>RectangleShape</td></tr> </table>	Appearance		Fill	HatchFill	Line	2 Solid DodgerBlue	Shadow	Shadow	ShapeStyle	RectangleShape					
Appearance															
Fill	HatchFill														
Line	2 Solid DodgerBlue														
Shadow	Shadow														
ShapeStyle	RectangleShape														
<p>TriangleShape - 二等辺三角形を描画します。Direction プロパティが三角形の向きを設定します。</p> <table border="1"> <tr><td colspan="2">Appearance</td></tr> <tr><td>Fill</td><td>HatchFill</td></tr> <tr><td>Line</td><td>2 Solid DeepPink</td></tr> <tr><td>Shadow</td><td>Shadow</td></tr> <tr><td>ShapeStyle</td><td>TriangleShape</td></tr> <tr><td>Direction</td><td>Right</td></tr> <tr><td>StyleName</td><td></td></tr> </table>	Appearance		Fill	HatchFill	Line	2 Solid DeepPink	Shadow	Shadow	ShapeStyle	TriangleShape	Direction	Right	StyleName		
Appearance															
Fill	HatchFill														
Line	2 Solid DeepPink														
Shadow	Shadow														
ShapeStyle	TriangleShape														
Direction	Right														
StyleName															

<p><b>CrossShape</b> – 十字を描画します</p> <table border="1"> <tr><td colspan="2"><b>Appearance</b></td></tr> <tr><td>Fill</td><td>HatchFill</td></tr> <tr><td>Line</td><td>2 Solid DarkTurquoise</td></tr> <tr><td>Shadow</td><td>Shadow</td></tr> <tr><td>ShapeStyle</td><td>CrossShape</td></tr> </table>		<b>Appearance</b>		Fill	HatchFill	Line	2 Solid DarkTurquoise	Shadow	Shadow	ShapeStyle	CrossShape									
<b>Appearance</b>																				
Fill	HatchFill																			
Line	2 Solid DarkTurquoise																			
Shadow	Shadow																			
ShapeStyle	CrossShape																			
<p><b>DiamondShape</b> – ひし形を描画します</p> <table border="1"> <tr><td colspan="2"><b>Appearance</b></td></tr> <tr><td>Fill</td><td>HatchFill</td></tr> <tr><td>Line</td><td>2 Solid DarkViolet</td></tr> <tr><td>Shadow</td><td>Shadow</td></tr> <tr><td>ShapeStyle</td><td>DiamondShape</td></tr> </table>		<b>Appearance</b>		Fill	HatchFill	Line	2 Solid DarkViolet	Shadow	Shadow	ShapeStyle	DiamondShape									
<b>Appearance</b>																				
Fill	HatchFill																			
Line	2 Solid DarkViolet																			
Shadow	Shadow																			
ShapeStyle	DiamondShape																			
<p><b>StarShape</b> – 星型を描画します。InternalRadius プロパティは内接円の半径を設定します。KeepRoundForm プロパティは星の形（楕円または円）を定義します。Points プロパティは放射線の数を定義します。StartAngle プロパティは星の回転角度を設定します。</p>																				
<table border="1"> <tr><td colspan="2"><b>Appearance</b></td></tr> <tr><td>Fill</td><td>HatchFill</td></tr> <tr><td>Line</td><td>2 Solid Salmon</td></tr> <tr><td>Shadow</td><td>Shadow</td></tr> <tr><td>ShapeStyle</td><td>StarShape</td></tr> <tr><td>InternalRadius</td><td>0,37999999523162842</td></tr> <tr><td>KeepRoundForm</td><td>False</td></tr> <tr><td>Points</td><td>5</td></tr> <tr><td>StartAngle</td><td>0</td></tr> </table>		<b>Appearance</b>		Fill	HatchFill	Line	2 Solid Salmon	Shadow	Shadow	ShapeStyle	StarShape	InternalRadius	0,37999999523162842	KeepRoundForm	False	Points	5	StartAngle	0	
<b>Appearance</b>																				
Fill	HatchFill																			
Line	2 Solid Salmon																			
Shadow	Shadow																			
ShapeStyle	StarShape																			
InternalRadius	0,37999999523162842																			
KeepRoundForm	False																			
Points	5																			
StartAngle	0																			
<p><b>ArrowShape</b> – 矢印を描画します。Direction プロパティが矢印の向きを設定します。</p>																				
<table border="1"> <tr><td colspan="2"><b>Appearance</b></td></tr> <tr><td>Fill</td><td>HatchFill</td></tr> <tr><td>Line</td><td>2 Solid SpringGreen</td></tr> <tr><td>Shadow</td><td>Shadow</td></tr> <tr><td>ShapeStyle</td><td>ArrowShape</td></tr> <tr><td>Direction</td><td>Left</td></tr> </table>		<b>Appearance</b>		Fill	HatchFill	Line	2 Solid SpringGreen	Shadow	Shadow	ShapeStyle	ArrowShape	Direction	Left							
<b>Appearance</b>																				
Fill	HatchFill																			
Line	2 Solid SpringGreen																			
Shadow	Shadow																			
ShapeStyle	ArrowShape																			
Direction	Left																			
<p><b>RectTriangleShape</b> – 5角形を描画します。Direction プロパティが図形の向きを設定します。</p>																				
<table border="1"> <tr><td colspan="2"><b>Appearance</b></td></tr> <tr><td>Fill</td><td>HatchFill</td></tr> <tr><td>Line</td><td>2 Solid DarkOrange</td></tr> <tr><td>Shadow</td><td>Shadow</td></tr> <tr><td>ShapeStyle</td><td>RectTriangleShape</td></tr> <tr><td>Direction</td><td>Up</td></tr> </table>		<b>Appearance</b>		Fill	HatchFill	Line	2 Solid DarkOrange	Shadow	Shadow	ShapeStyle	RectTriangleShape	Direction	Up							
<b>Appearance</b>																				
Fill	HatchFill																			
Line	2 Solid DarkOrange																			
Shadow	Shadow																			
ShapeStyle	RectTriangleShape																			
Direction	Up																			
<p><b>EllipseShape</b> – 楕円を描画します。</p>																				
<table border="1"> <tr><td colspan="2"><b>Appearance</b></td></tr> <tr><td>Fill</td><td>HatchFill</td></tr> <tr><td>Line</td><td>2 Solid DarkCyan</td></tr> <tr><td>Shadow</td><td>Shadow</td></tr> <tr><td>ShapeStyle</td><td>EllipseShape</td></tr> </table>		<b>Appearance</b>		Fill	HatchFill	Line	2 Solid DarkCyan	Shadow	Shadow	ShapeStyle	EllipseShape									
<b>Appearance</b>																				
Fill	HatchFill																			
Line	2 Solid DarkCyan																			
Shadow	Shadow																			
ShapeStyle	EllipseShape																			
<p><b>RoundRectangleShape</b> – 角丸四角形を描画します。Round プロパティは角丸の半径を設定します。</p>																				
<table border="1"> <tr><td colspan="2"><b>Appearance</b></td></tr> <tr><td>Fill</td><td>HatchFill</td></tr> <tr><td>Line</td><td>2 Solid DarkSlateBlue</td></tr> <tr><td>Shadow</td><td>Shadow</td></tr> <tr><td>ShapeStyle</td><td>RoundRectangleShape</td></tr> <tr><td>Round</td><td>0,4</td></tr> </table>		<b>Appearance</b>		Fill	HatchFill	Line	2 Solid DarkSlateBlue	Shadow	Shadow	ShapeStyle	RoundRectangleShape	Round	0,4							
<b>Appearance</b>																				
Fill	HatchFill																			
Line	2 Solid DarkSlateBlue																			
Shadow	Shadow																			
ShapeStyle	RoundRectangleShape																			
Round	0,4																			
<p><b>ParallelogramShape</b> – 平行四辺形を描画します。Angle プロパティは平行四辺形の回転角度を設定します。</p>																				

<b>Appearance</b> Fill HatchFill Line 2 Solid DarkMagenta Shadow Shadow ShapeStyle ParallelogramShape Angle 45		
LineShape –線を描画します。LineKind プロパティは線の回転角度を設定します。		
<b>Appearance</b> Fill [None] Line 2 Solid Red Shadow Shadow ShapeStyle LineShape LineKind BackSlash		

### シェープの影

シェープの影は **Shadow** プロパティで設定します。DX 値と DY 値は影のオフセットを設定します。Fill プロパティは塗りつぶしを設定します。影を追加するとオブジェクトの大きさは同じですが、シェープの大きさは小さくなります。

シェープの影の使用例：



### シェープの枠

シェープの枠は、**Line** プロパティで定義します。**Color** プロパティは線の色を設定します。**Style** プロパティは線のスタイルを設定します。**Width** プロパティは線の幅を設定します。

枠線の例：



### バーコード

**BarCode** コンポーネントはバーコードを作成するために使用します。バーコードのコードは **Code** プロパティに設定します。大半のバーコードに数字を使用できます。一部の高度なバーコードにラテン記号を使用できます。

プロパティ	レポート
CalcCheckSumm False Code <b>123456</b> CodeType Code39	 123456
CalcCheckSumm False Code <b>Code 123456</b> CodeType <b>Code39Extended</b>	 Code 123456

バーコードの比率は **BarRatio** プロパティに設定します。

### バーコードの種類

バーコードの種類は **CodeType** プロパティに設定します。利用可能なバーコード型は次の通りです。  
 Code\_2\_5\_interleaved, Code\_2\_5\_industrial, Code\_2\_5\_matrix, Code39 (default value), Code39Extended, Code128A, Code128B, Code128C, Code93, Code93Extended, CodeMSI,



CodePostNet, CodeCodabar, CodeEAN8, CodeJAN8, CodeEAN13, CodeJAN13, CodeUPC\_A, CodeUPC\_E0, CodeUPC\_E1, CodeUPC\_Supp2, CodeUPC\_Supp5, CodeEAN128A, CodeEAN128B, CodeEAN128C.

2次元バーコードを使用するには次のような付加的な設定が必要です。

PDF417、PDF417Compact の場合：Pdf417Settings のプロパティ群の設定

CodeType	PDF417
Fill	[None]
Font	Arial; 12pt
LabelPosition	BottomCenter
Margins	0; 0; 0; 0 cm
MicroQRCodeSettings	PerpetuumSoft.Framework.Drawir
Pdf417Settings	3; 0; 5; TextMode
ColumnsCount	3
ErrorLevel	0
RowsCount	5
Type	TextMode

QRCode の場合：QRCodeSettings のプロパティ群の設定

CodeType	QRCode
Fill	[None]
Font	Arial; 12pt
LabelPosition	BottomCenter
Margins	0; 0; 0; 0 cm
MicroQRCodeSettings	PerpetuumSoft.Framework.Drawir
Pdf417Settings	3; 0; 5; TextMode
QRCodeSettings	PerpetuumSoft.Framework.Drawi
ErrorCorrectionLevel	L
Mask	Mask000
MirrorImaging	False
ReflectanceReversal	False
Version	4

MicroQRCode の場合：MicroQRCodeSettings のプロパティ群の設定

CodeType	MicroQRCode
Fill	[None]
Font	Arial; 12pt
LabelPosition	BottomCenter
Margins	0; 0; 0; 0 cm
MicroQRCodeSettings	PerpetuumSoft.Framework.Drawi
ErrorCorrectionLevel	L
Mask	Mask00
MirrorImaging	False
ReflectanceReversal	False
Version	4

郵便カスタマーバーコードを利用するには、Barcode オブジェクトの CodeType プロパティを「Japan\_Postal」に設定します。このバーコードでは、BarPoint プロパティにバーコードのポイント数（日本郵便では 8 ポイントから 11.5 ポイントを推奨、デフォルト 10 ポイント）を指定することでデザイン時および実行時に自動的に Barcode オブジェクトの大きさが決まるようになっています。なお、BarPoint プロパティは CodeType プロパティが Japan\_Postal の場合のみ有効なプロパティです。



プロパティ		レポート
BarPoint	10	
BarRatio	2	
Border		
CalcCheckSumm	False	
Code	123456	
CodeType	Japan_Postal	

### バーコードのスタイル

バーコードの背景色を作成するには、BackColor プロパティを使用します。バーコード自体の色は BarColor プロパティで設定できます。

プロパティ		レポート
BackColor	<input type="checkbox"/> Transparent	
BarColor	<input type="checkbox"/> Black	
BackColor	<input checked="" type="checkbox"/> DodgerBlue	
BarColor	<input checked="" type="checkbox"/> DeepPink	

ShowLabel プロパティはバーコードのラベル表示を定義します。ラベルのフォント (Font プロパティ)、色 (TextColor プロパティ)、位置 (LabelPosition プロパティ) を設定できます。

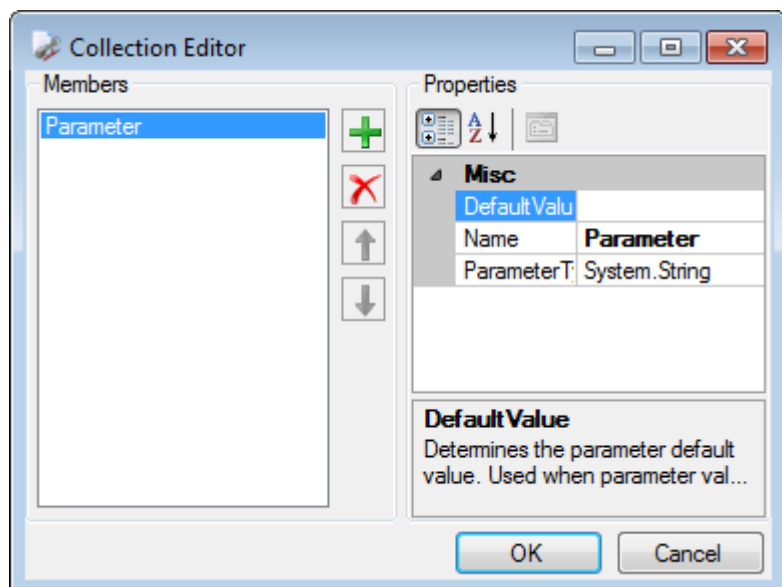
プロパティ		レポート
Font	Calibri; 9pt; Bold=Off; Italic=Off; U	
LabelPosition	TopLeft	
Margins	0; 0; 0; 0 cm	
ShowCheckDigit	False	
ShowLabel	False	
StyleName		

### CalcCheckSumm と ShowCheckDigit プロパティ

未定

### Parameters

Parameters プロパティは、レポートパラメータの設定に使用します。コレクションエディタは、このプロパティのエディタです。



**defaultValue** プロパティはパラメータ値を設定します。**Name** プロパティはサブレポートのパラメータ値を表す名前を設定します。**ParameterType** プロパティはパラメータ型を設定します（規定ではすべてのパラメータに **string** 型が使用されます）。

サブレポートのパラメータ値を取得する式：

```
GetParameter("<ParameterName>")
```

未使用

## ハイパーリンクとレポートのナビゲーション

ハイパーリンクを設定するには、オブジェクトの **Hyperlink** プロパティを使用します。

### 外部アプリケーションの呼び出し

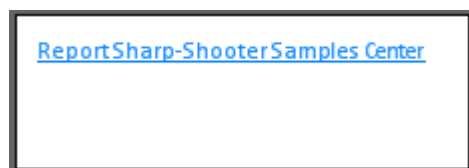
**Hyperlink** プロパティを使用して、外部のアプリケーションを呼び出すことができます。

**TextBox** オブジェクトの **Hyperlink** プロパティを使用してアプリケーションを呼び出す例：

```
textBox.Hyperlink プロパティ = "C:\Program Files\Perpetuum Software\SharpShooter Reports\Bin\SharpShooterReporsSamplesCenter.exe"
```

Navigation	
Bookmark	
Hyperlink	C:\Program Files\Perpetuum

レポート：



**TextBox** をクリックすると **SharpShooter Reports Samples Center** が起動します。

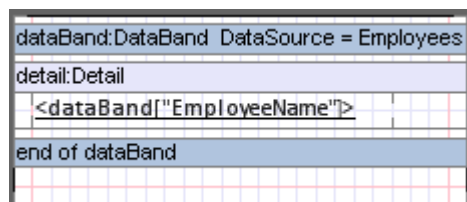
### ファイルを開く

**Hyperlink** プロパティに設定されたファイルパスは、ファイルの種類に合ったプログラムを使用して表示されます。

ファイルパスの設定例：

**Hyperlink** プロパティは、レポートの生成中にファイルアドレスのスクリプトで設定されます。

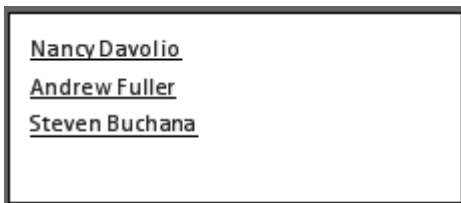
レポートテンプレート：



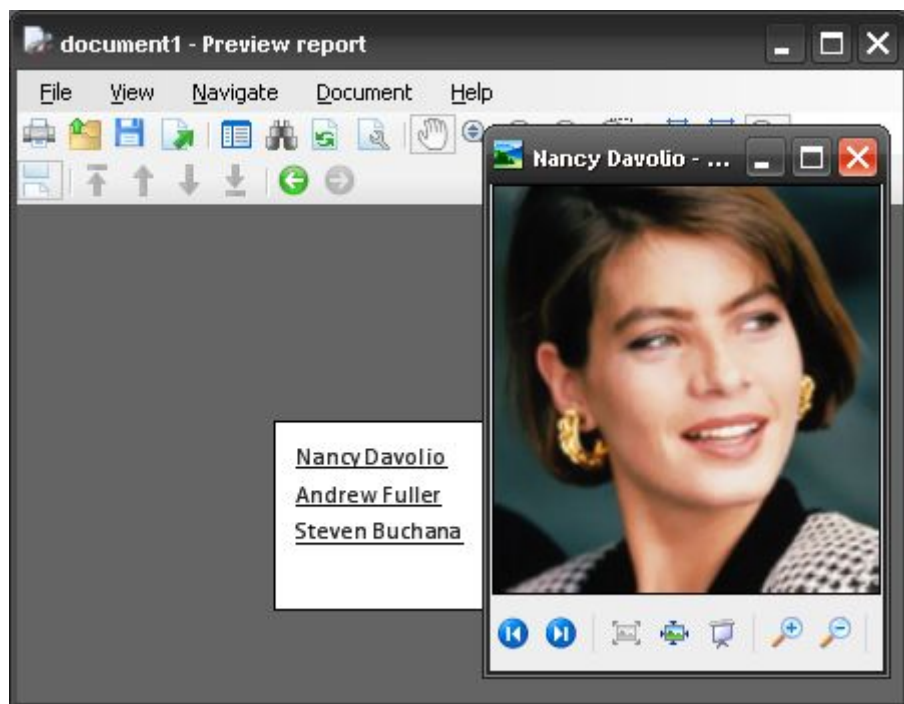
```
dataBand.DataSource = Employees;
```

```
textBox.Value = dataBand["EmployeeName"]; textBox.Hyperlink = "C:\\Data\\Pictures\\"+dataBand["EmployeeName"].ToString()+".png"
```

レポート :



TextBox オブジェクトの領域をクリックすると、イメージのビューアを使って対象のイメージが表示されます。



### インターネット リソースや Emails のハイパーリンク

Hyperlink プロパティにインターネットリソースのハイパーリンクを取得するには、URL を設定する必要があります。

例 : `textBox.Hyperlink = "http://www.perpetuumsoft.com/"`

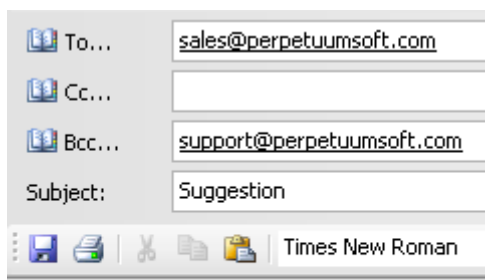
Navigation	
Bookmark	
Hyperlink	<a href="http://www.perpetuumsoft.com/">http://www.perpetuumsoft.com/</a>

最終ドキュメントの TextBox オブジェクトをクリックすると、ブラウザにハイパーリンクが表示されます。

リンクをクリックすると電子メールクライアントが開くようにするには、Hyperlink プロパティに `mailto` ハイパーリンク (例 : `textBox.Hyperlink = "mailto:sales@perpetuumsoft.com"`) を設定します。そしてそのハイパーリンクをクリックすると、電子メールクライアントが開いて、設定されたアドレスにメッセージを送ることができます。件名、本文、Cc、Bcc などのパラメータも同じです。件名、本文、Cc/Bcc を送信するための Email の設定が可能です。例えば、ハイパーリンク :

<mailto:sales@perpetuumsoft.com?subject=Suggestion&Body=Hello!%0DI%20have%20some%20ideas%20for%20you.&bcc=support@perpetuumsoft.com> は次のようなデータの電子メールクライアントを開きます。



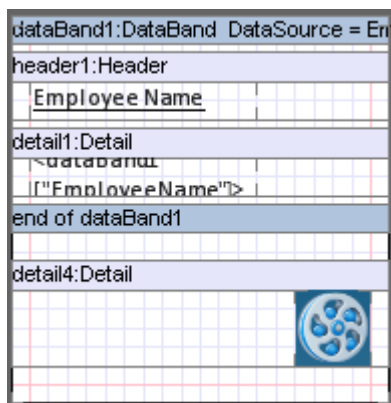


Hello!  
I have some ideas for you. |

## ブックマーク

レポートを簡単に移動するには、**Bookmark** プロパティを使用できます。このプロパティは、レポートビューアのコンテンツツリーを使ってレポートの別の場所から対象のオブジェクトに移動できるブックマークを作成します。ブックマーク名は、# 記号で始めなければなりません。そうしないと **Bookmark** プロパティの値は無視されます。

レポートテンプレート :

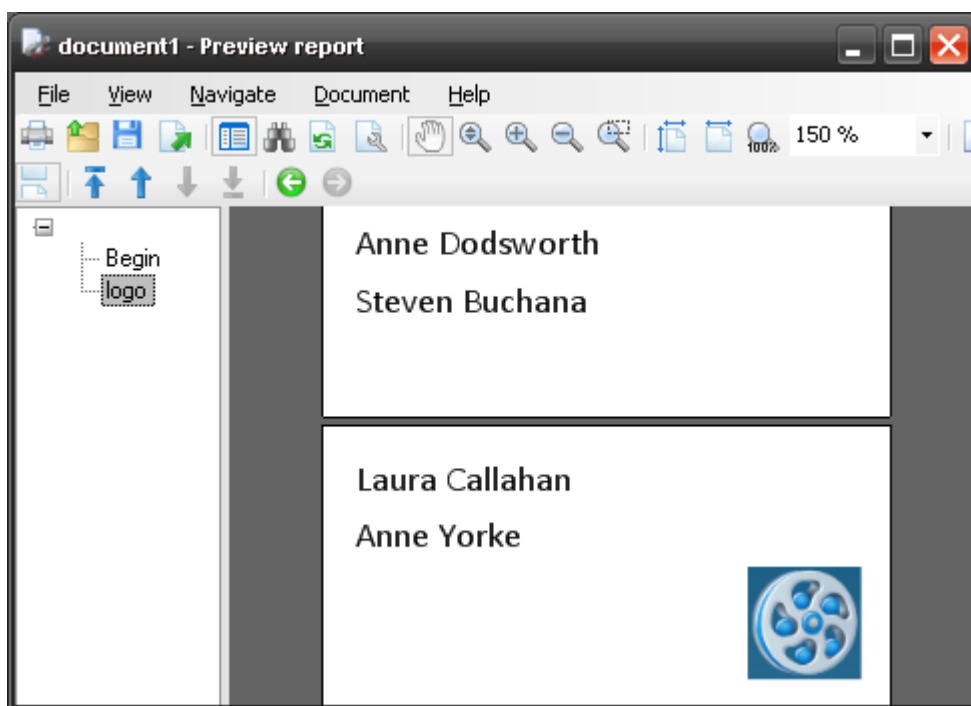


```
textBoxHeader.Bookmark = #Begin;
```

```
picture.Bookmark = #logo;
```



レポートビューアのレポート :

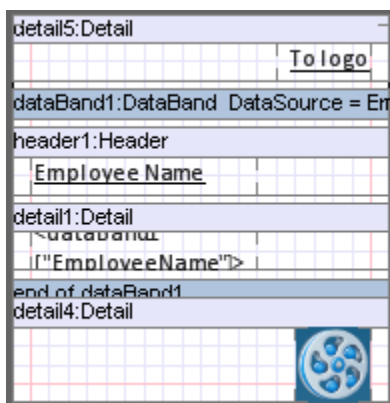


ブックマークは左側にツリー表示されます。「コンテンツ表示」ボタンはこのツリーを表示/非表示にするために使用します。コンテンツからブックマークが設定されている項目を選択すると、ブックマークの設定先のレポート領域が表示されます。

### ハイパーリンクとブックマークを使ったレポートの移動

ハイパーリンクを使ってブックマークに移動するには、**Bookmark** プロパティを使用できます。それには、**Hyperlink** プロパティにブックマークの名前を設定する必要があります。

レポートテンプレート :

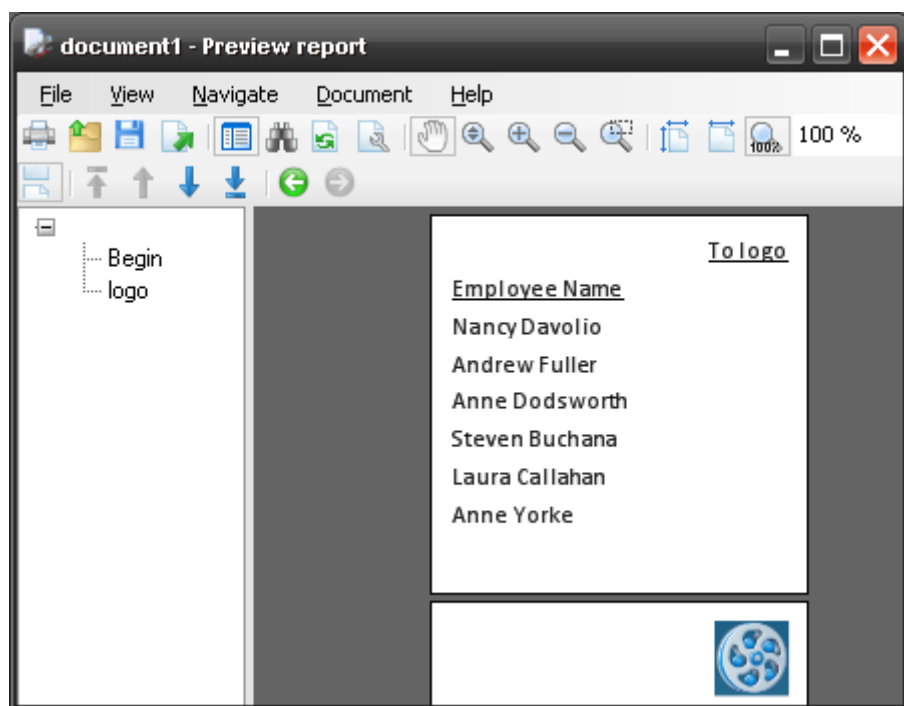


picture.Bookmark = #logo;

textBoxToLogo.Hyperlink = #logo

Navigation	
Bookmark	
Hyperlink	#logo

レポートビューアのレポート :



「To logo」というテキストが設定された **TextBox** をクリックすると、ロゴの入ったレポート部分が開きます。前のレポート領域に戻るには、「Back」というハイパーリンクを設定して戻ることができます。

## サブレポートの使用

複数のレポートをひとつにまとめなければならない場合にサブレポートの使用メカニズムが使われます。サブレポートはそのために使用され、テンプレートに入れるサブレポートとその位置を定義します。サブレポートのテンプレートは、現在のテンプレートが含まれている **ReportManager** に入れてください。

サブレポートのテンプレートを設定するには、サブレポートの **TemplateName** プロパティを使用します。このプロパティのコンボボックスには、現在の **ReportManager** に含まれているレポート名が表示されます（**ReportManager** に **ReportSlot** の **ReportName** プロパティが設定されていない場合、テンプレート名はリストには表示されません）。

<b>Misc</b>	
Parameters	(Collection)
Shift	0 cm
TemplateName	
<b>Navigation</b>	
Bookmark	List
Hyperlink	Properties
	Report1
	Report2
<b>Render</b>	
CanGrow	
CanShrink	
NewColumnAfter	False

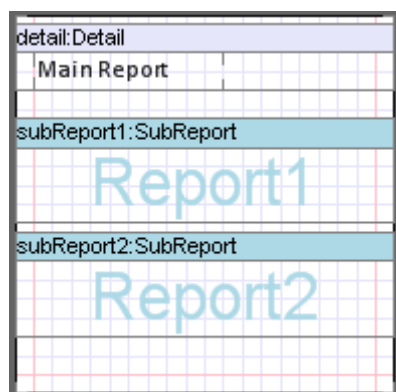


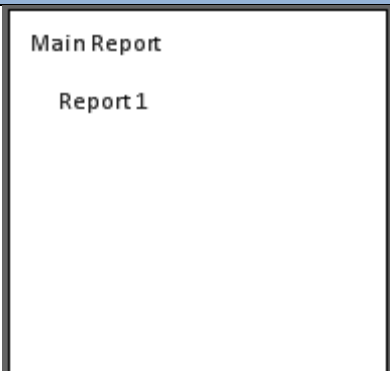
マスターレポートテンプレート	サブレポートテンプレート	レポート
 <p>プロパティ： subReport.TemplateName = Report1</p>		

(DataBand、Detail などの) バンドに格納されたオブジェクトだけが最終レポートに含まれます。サブレポートのオブジェクトにスタイルが設定されていない場合、マスターテンプレートまたはサブレポートコントロールのスタイルが適用されます。

Report1 の左側の余白を移動させるには、サブレポートの **Shift** プロパティを使用します。

レポートテンプレート：



プロパティ	レポート						
<p>☐ <b>Misc</b></p> <table border="1"> <tr> <td>Parameters</td> <td>(Collection)</td> </tr> <tr> <td>Shift</td> <td>0 cm</td> </tr> <tr> <td>TemplateName</td> <td><b>Report1</b></td> </tr> </table>	Parameters	(Collection)	Shift	0 cm	TemplateName	<b>Report1</b>	
Parameters	(Collection)						
Shift	0 cm						
TemplateName	<b>Report1</b>						



<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #e0e0e0; padding: 2px;"><b>Misc</b></div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Parameters</td> <td>(Collection)</td> </tr> <tr> <td>Shift</td> <td>2 cm</td> </tr> <tr> <td>TemplateName</td> <td>Report1</td> </tr> </table> </div>		Parameters	(Collection)	Shift	2 cm	TemplateName	Report1	<div style="border: 1px solid black; padding: 10px; min-height: 150px;"> <p style="margin: 0;">Main Report</p> <p style="margin: 20px 0 0 40px;">Report 1</p> </div>
Parameters	(Collection)							
Shift	2 cm							
TemplateName	Report1							

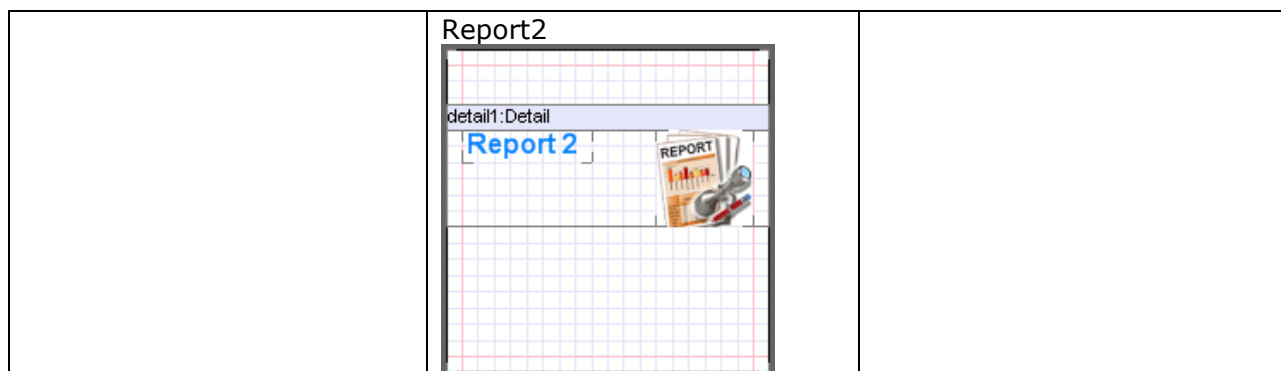
### 複雑なレポート

1つのレポートにサブレポートを好きなだけ追加できます。

マスターレポートテンプレート	サブレポートテンプレート	レポート										
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <table border="1" style="width: 100%; border-collapse: collapse; background-color: #f0f0f0;"> <tr><td style="background-color: #e0e0e0;">detail:Detail</td></tr> <tr><td>Main Report</td></tr> <tr style="background-color: #e0e0e0;"><td>subReport1:SubReport</td></tr> <tr><td style="text-align: center; color: #4a7ebb; font-size: 2em;">Report1</td></tr> <tr style="background-color: #e0e0e0;"><td>subReport2:SubReport</td></tr> <tr><td style="text-align: center; color: #4a7ebb; font-size: 2em;">Report2</td></tr> </table> </div> <p>プロパティ:  subReport1.TemplateName = Report1  subReport2.TemplateName = Report2</p>	detail:Detail	Main Report	subReport1:SubReport	Report1	subReport2:SubReport	Report2	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="margin: 0;"><b>Report1</b></p> <table border="1" style="width: 100%; border-collapse: collapse; background-color: #f0f0f0;"> <tr><td style="background-color: #e0e0e0;">detail:Detail</td></tr> <tr><td style="text-align: center;">Report 1</td></tr> </table> </div> <div style="border: 1px solid black; padding: 5px;"> <p style="margin: 0;"><b>Report2</b></p> <table border="1" style="width: 100%; border-collapse: collapse; background-color: #f0f0f0;"> <tr><td style="background-color: #e0e0e0;">detail1:Detail</td></tr> <tr><td style="text-align: center;">Report 2 </td></tr> </table> </div>	detail:Detail	Report 1	detail1:Detail	Report 2	<div style="border: 1px solid black; padding: 10px; min-height: 150px;"> <p style="margin: 0;">Main Report</p> <p style="margin: 10px 0 0 40px;">Report 1</p> <p style="margin: 0 0 0 40px; color: #4a7ebb; font-size: 1.2em;">Report 2</p> </div>
detail:Detail												
Main Report												
subReport1:SubReport												
Report1												
subReport2:SubReport												
Report2												
detail:Detail												
Report 1												
detail1:Detail												
Report 2												

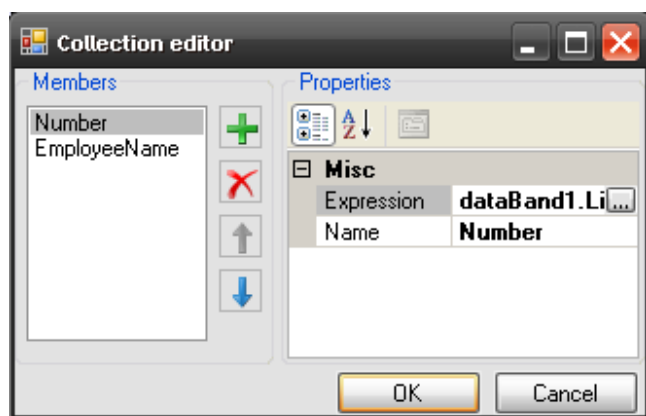
入れ子のレポートを使用できます。

マスターレポートテンプレート	サブレポートテンプレート	レポート								
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <table border="1" style="width: 100%; border-collapse: collapse; background-color: #f0f0f0;"> <tr><td style="background-color: #e0e0e0;">detail:Detail</td></tr> <tr><td>Main Report</td></tr> <tr style="background-color: #e0e0e0;"><td>subReport1:SubReport</td></tr> <tr><td style="text-align: center; color: #4a7ebb; font-size: 2em;">Report1</td></tr> </table> </div> <p>プロパティ:  subReport1.TemplateName = Report1</p>	detail:Detail	Main Report	subReport1:SubReport	Report1	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="margin: 0;"><b>Report1</b></p> <table border="1" style="width: 100%; border-collapse: collapse; background-color: #f0f0f0;"> <tr><td style="background-color: #e0e0e0;">detail:Detail</td></tr> <tr><td style="text-align: center;">Report 1</td></tr> </table> </div> <div style="border: 1px solid black; padding: 5px;"> <p style="margin: 0;"><b>Report2</b></p> <table border="1" style="width: 100%; border-collapse: collapse; background-color: #f0f0f0;"> <tr><td style="background-color: #e0e0e0;">subReport1:SubReport</td></tr> <tr><td style="text-align: center; color: #4a7ebb; font-size: 2em;">Report2</td></tr> </table> </div> <p>プロパティ:  subReport1.TemplateName = Report2</p>	detail:Detail	Report 1	subReport1:SubReport	Report2	<div style="border: 1px solid black; padding: 10px; min-height: 150px;"> <p style="margin: 0;">Main Report</p> <p style="margin: 10px 0 0 40px;">Report 1</p> <p style="margin: 0 0 0 40px; color: #4a7ebb; font-size: 1.2em;">Report 2</p> </div>
detail:Detail										
Main Report										
subReport1:SubReport										
Report1										
detail:Detail										
Report 1										
subReport1:SubReport										
Report2										



### サブレポートにパラメータを渡す

サブレポートの **Parameters** プロパティを使用してマスターレポートからサブレポートにパラメータを渡すことができます。コレクションエディタはこのプロパティのエディタです。

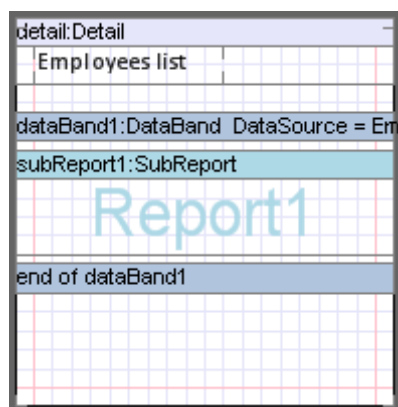


**Expression** プロパティには、計算結果をパラメータとして渡す式を設定します。**Name** プロパティはサブレポートのパラメータ値を表す名前を設定します。

サブレポートのパラメータ値を取得するには、式 : `GetParameter("<ParameterName>")` を使用します。

下図は社員一覧を表示するテンプレートの例です。名前と番号（連番）がサブレポートに表示されます。

マスターレポートテンプレート :

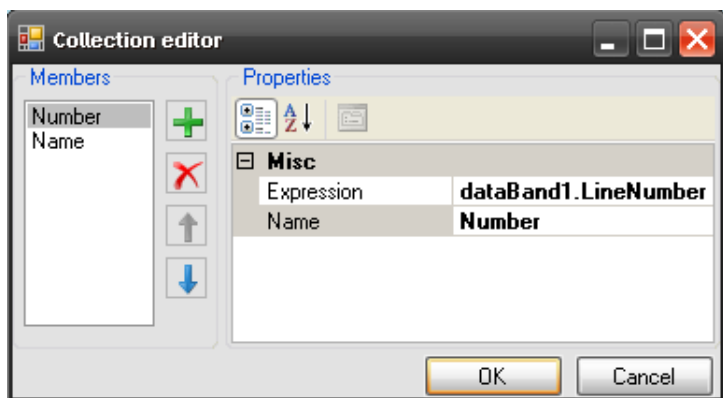


プロパティ : `subReport1.TemplateName = Report1`

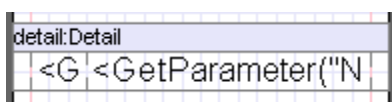
`subReport1.Parameters` プロパティのコレクションにパラメータが 2 つ追加されています。

1 つ目のパラメータ : `Expression = dataBand1.LineNumber, Name = Number`

2 つ目のパラメータ : `Expression = dataBand1["EmployeeName"], Name = Name`



サブレポートテンプレート :



```
textBox1.Value = GetParameter("Number");
```

```
textBox2.Value = GetParameter("Name")
```

レポート :

Employees list	
1	Nancy Davolio
2	Andrew Fuller
3	Anne Dodsworth
4	Steven Buchana
5	Laura Callahan
6	Anne Yorke

## Zip コード

Zip コードを表示するには、ZipCode コンポーネントを使用します。コードは Code プロパティに設定します。

Appearance	
Angle	0
Border	
Code	22550
Color	<span style="background-color: black; color: black;">Black</span>

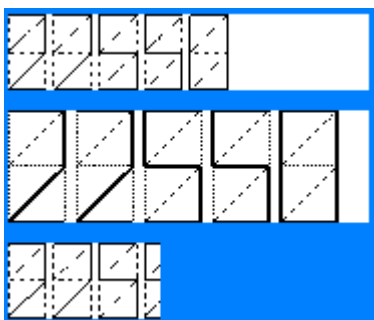
コードの色は Color プロパティに設定します。

Appearance	
Angle	0
Border	
Code	22550
Color	<span style="background-color: blue; color: blue;">DodgerBlue</span> ▼
Fill	[None]

テンプレートの Zip コード :



ZipCode コンポーネントのサイズは **Size** プロパティで設定します。Zip コードの割合が一定であることを考慮する必要があります。Zip コードのラベルの高さは ZipCode コンポーネントの高さに依存します。ラベルが ZipCode コンポーネントの幅に収まらない場合、Zip コードは途中で切れてしまいます。



## マスターレポート

マスターレポートを使用できます。マスターレポートを使用してドキュメントを生成する場合、最終ドキュメントにはマスターテンプレートとドキュメントテンプレートの生成結果が含まれます。これにより、均一のレポートを作成できます。

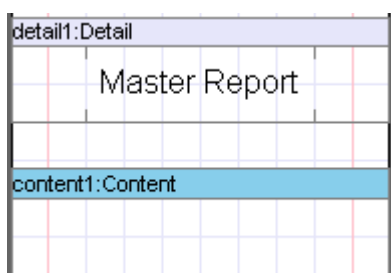
マスターレポートを設定するには、**Document.MasterReport** プロパティにマスターレポート名を設定する必要があります。



指名されたテンプレートは生成時に **IResolveSubReport** インターフェイスを介して取得されます。例えば、指名されたマスターテンプレートを、最初のテンプレートが追加されているのと同じ **ReportManager** に追加できます。

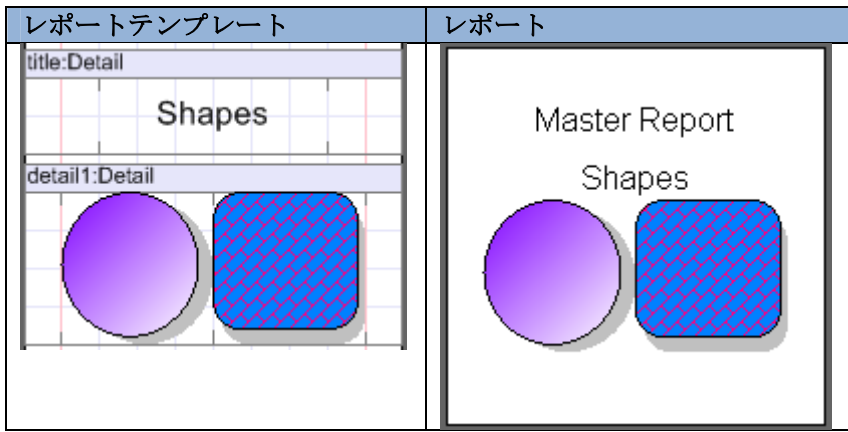
マスターとして使用するテンプレートを作成する場合、そのテンプレートに **Content** オブジェクトを追加する必要があります。

マスターレポートテンプレート :



マスターテンプレートを使用したドキュメントの生成は次のように行われます。生成されたレポートにマスターテンプレートの生成結果が入れられ、**Content** オブジェクトは最初のレポート結果に置き換えられます。





## ウィザードを使ったレポートの作成

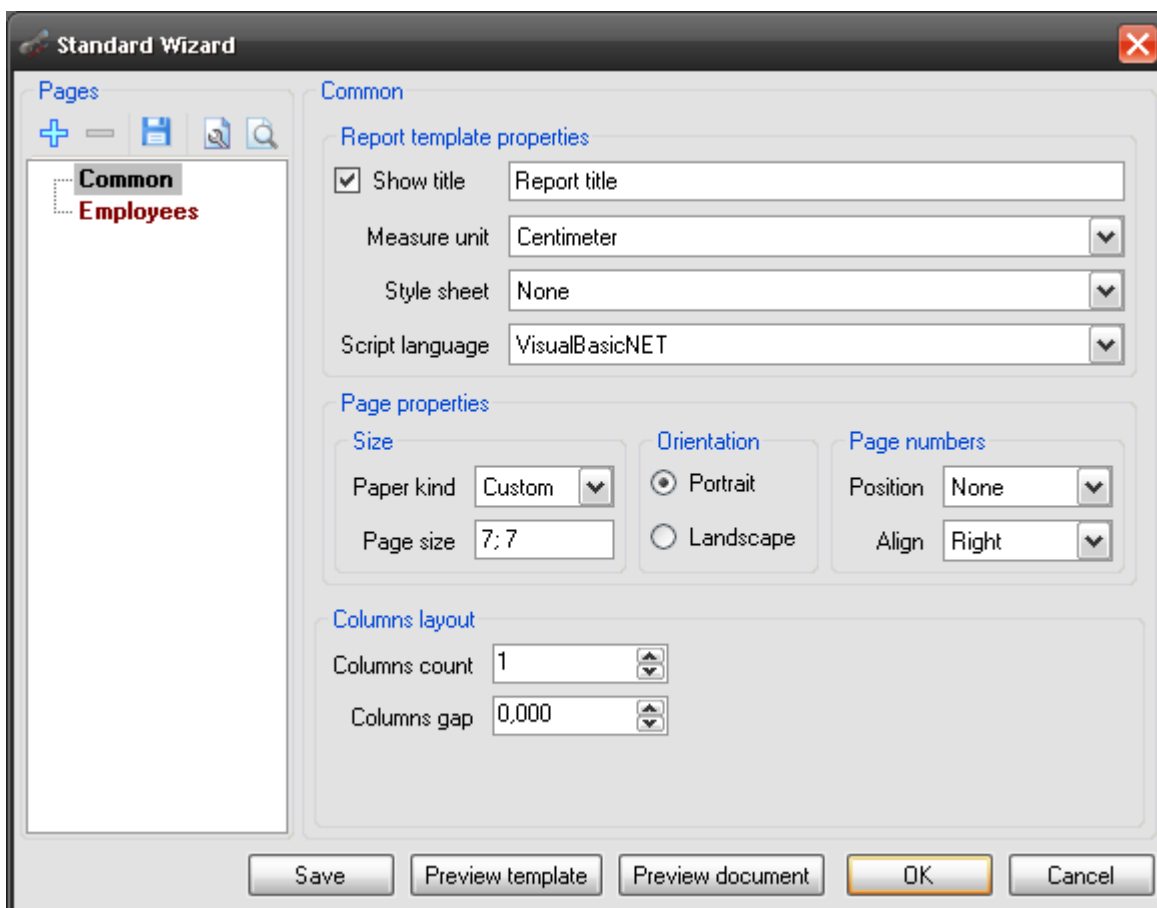
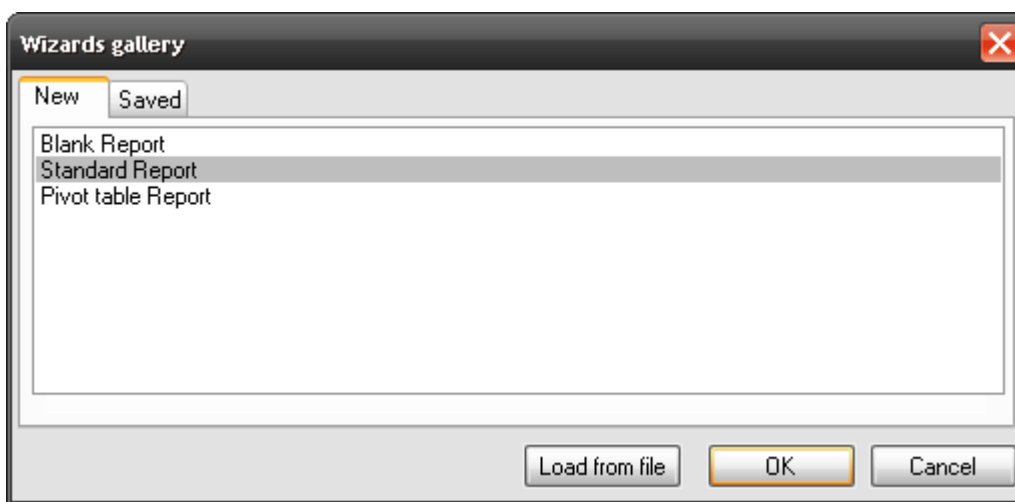
シンプルなレポートテンプレートを作成する簡単な方法は、「ウィザード」を使うことです。見た目やデータソースなどのレポートパラメータをウィザードで設定できます。そして、自動的にパラメータを設定してレポートテンプレートを作成できます。

ウィザードの設定はすべて xml 形式で .rsw ファイルに保存できます。

### シンプルレポートの作成 (スタンダードウィザード)

スタンダードウィザードを使用してレポートテンプレートを作成できます。スタンダードウィザードでデータソースのデータをリストまたはカードとして出力するかを定義します。

ウィザードギャラリーダイアログからスタンダードウィザードを開くことができます。



### スタンダードウィザードのレポートのプロパティ

スタンダードウィザードの「共通」セクションを使って、レポートの見た目を定義するテンプレートに共通するプロパティを設定できます。

(Document.Title プロパティを使用して「タイトル」フィールドに設定した) レポート名はレポートヘッダーに表示されます。

タイトル	レポートテンプレート	レポート				
<p>Report template properties</p> <p><input checked="" type="checkbox"/> Show title Employees</p> <p>Measure unit Centimeter</p>		<p>Employees</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Phone</th> </tr> </thead> <tbody> <tr> <td>Nancy</td> <td>(206)555</td> </tr> </tbody> </table>	Name	Phone	Nancy	(206)555
Name	Phone					
Nancy	(206)555					

レポートの計測単位は、「計測単位」フィールド (Document.MetricUnit プロパティ) で設定します。

レポートのスタイルは、「スタイルシート」フィールド (Document.StyleSheet プロパティ) で設定できます。

スタイルシート	プロパティ	レポート														
<p>Measure unit Centimeter</p> <p>Style sheet Modern1</p> <p>Script language VisualBasic.NET</p>	<p><input checked="" type="checkbox"/> Render</p> <p>StyleSheet Modern1</p> <p><input checked="" type="checkbox"/> Script</p>	<p>Employees</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Phone</th> </tr> </thead> <tbody> <tr> <td>Nancy Davolio</td> <td>(206)555-9857</td> </tr> <tr> <td>Andrew Fuller</td> <td>(206) 555-9482</td> </tr> <tr> <td>Anne Dodsworth</td> <td>(71) 555-4444</td> </tr> <tr> <td>Steven Buchana</td> <td>(71) 555-4444</td> </tr> <tr> <td>Laura Callahan</td> <td>(71) 555-4444</td> </tr> <tr> <td>Anne Yorke</td> <td>(71) 555-4444</td> </tr> </tbody> </table>	Name	Phone	Nancy Davolio	(206)555-9857	Andrew Fuller	(206) 555-9482	Anne Dodsworth	(71) 555-4444	Steven Buchana	(71) 555-4444	Laura Callahan	(71) 555-4444	Anne Yorke	(71) 555-4444
Name	Phone															
Nancy Davolio	(206)555-9857															
Andrew Fuller	(206) 555-9482															
Anne Dodsworth	(71) 555-4444															
Steven Buchana	(71) 555-4444															
Laura Callahan	(71) 555-4444															
Anne Yorke	(71) 555-4444															
<p>Measure unit Centimeter</p> <p>Style sheet Modern</p> <p>Script language VisualBasic.NET</p>	<p><input checked="" type="checkbox"/> Render</p> <p>StyleSheet Modern</p> <p><input checked="" type="checkbox"/> Script</p>	<p>Employees</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Phone</th> </tr> </thead> <tbody> <tr> <td>Nancy Davolio</td> <td>(206)555-9857</td> </tr> <tr> <td>Andrew Fuller</td> <td>(206) 555-9482</td> </tr> <tr> <td>Anne Dodsworth</td> <td>(71) 555-4444</td> </tr> <tr> <td>Steven Buchana</td> <td>(71) 555-4444</td> </tr> <tr> <td>Laura Callahan</td> <td>(71) 555-4444</td> </tr> <tr> <td>Anne Yorke</td> <td>(71) 555-4444</td> </tr> </tbody> </table>	Name	Phone	Nancy Davolio	(206)555-9857	Andrew Fuller	(206) 555-9482	Anne Dodsworth	(71) 555-4444	Steven Buchana	(71) 555-4444	Laura Callahan	(71) 555-4444	Anne Yorke	(71) 555-4444
Name	Phone															
Nancy Davolio	(206)555-9857															
Andrew Fuller	(206) 555-9482															
Anne Dodsworth	(71) 555-4444															
Steven Buchana	(71) 555-4444															
Laura Callahan	(71) 555-4444															
Anne Yorke	(71) 555-4444															

「スクリプト言語」フィールドでスクリプト言語を選択できます。スクリプト言語は Document.ScriptLanguage プロパティを使用して設定されます。

### スタンダードウィザードのページのプロパティ

スタンダードウィザードのページのプロパティはページのプロパティを設定できます。「用紙サイズ」のリストから用紙サイズが選択され、Page.PaperKind プロパティで設定されます。カスタムな用紙サイズを設定するには、「用紙サイズ」コンボボックスから「ユーザー定義用紙」を選択し、「ページサイズ」フィールドに値を設定してください。

Size

Paper kind Custom

Page size 7:7

用紙の向き (縦または横) は「向き」のセクションで設定できます。

Orientation

Portrait

Landscape

ページ番号を出力するには見出しを使用します。「ページ番号」セクションで見出しの表示モードやテキストの位置合わせを設定できます。

Page numbers

Position

Align

レポート :

Employees	
Name	Phone
Nancy Davolio	(206)555-9857
Andrew Fuller	(206) 555-9482
Anne Dodsworth	(71) 555-4444
Steven Buchana	(71) 555-4444
Laura Callahan	(71) 555-4444

スタンダードウィザードを使用して段組レポートを作成できます。それには、「列のレイアウト」を設定します。「列数」は列の数 (`DataBand.ColumnsCount` プロパティ) を定義し、「列の間隔」は列の間隔 (`DataBand.ColumnsGap` プロパティ) を定義します。

Columns layout

Columns count

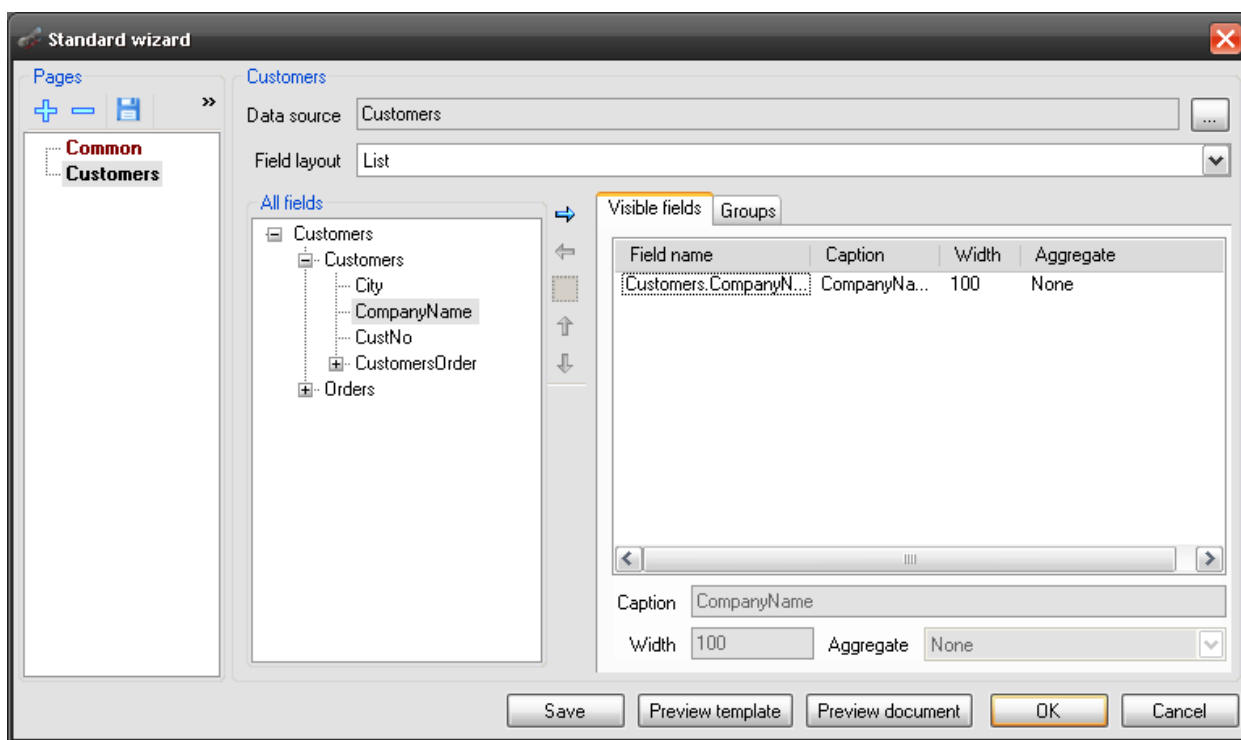
Columns gap

レポート :

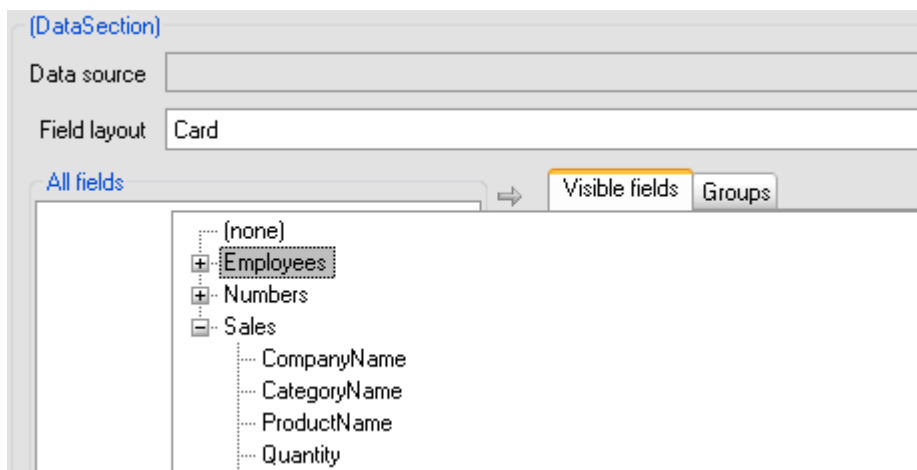
Employees	
Nancy Davolio	Laura Callahan
Andrew Fuller	Anne Yorke
Anne Dodsworth	
Steven Buchana	

### シンプルリストのレポートデータ

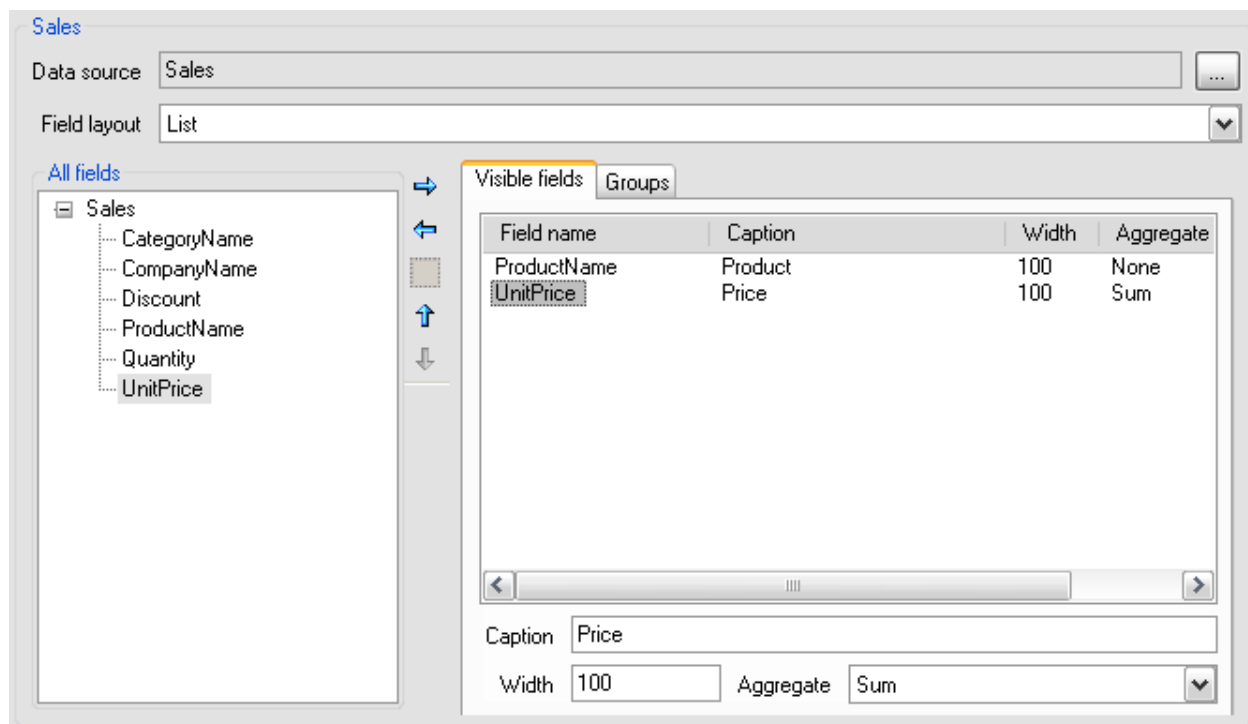
レポートを作成するためのデータは、レポートデザイナーを呼び出すアプリケーションで用意してください。データを追加するには、「データの追加」ボタンを押してください。この後、テンプレートのツリー構造に新しいレコードが表示され、データソースを設定したりデータを表示するためのセクションが開きます。



「データソース」フィールドでデータソースを設定します。



「全フィールド」の一覧に、「データソース」フィールドで選択したデータソースフィールドが表示されます。コントロールボタンを使用して「全フィールド」から「表示フィールド」にフィールドを追加できます。各フィールドにタイトルや幅の編集、集計関数の設定が行えます。



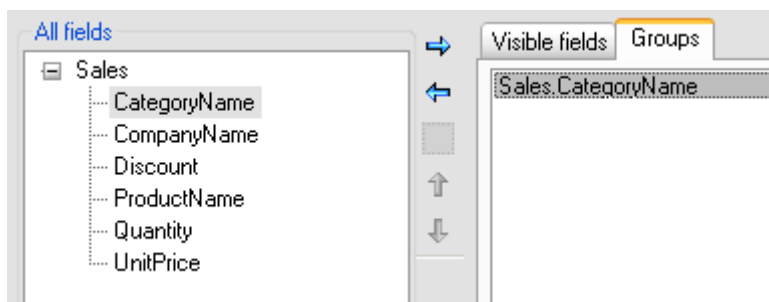
レポートテンプレート	レポート
dataBand1:DataBand DataSource = Sales	
header1:Header	<b>Product</b> <b>Price</b>
detail1:Detail	Chai              35,5
<GetData !["Sales.ProductName"]>    !["Sales.UnitPri	Stout             105
footer1:Footer	Maxilaku        2,6
end of dataBand1	Geitost          15,8
	Flotemysost     240
	Courdavault     62,5
	Chai              35,5
	Ipoh Coffee     50,5
	Filo Mix          14
	Filo Mix          14
	<b>575,4</b>

「フィールドのレイアウト」でレポートのデータ表示（リストまたはカード）を選択できます。

フィールドのレイアウト	レポートテンプレート	レポート
リスト： Field layout List	dataBand1:DataBand DataSource = Employees header1:Header <b>Employee</b> <b>Phone</b> detail1:Detail <GetData      <GetData !["Employees.EmployeeName"]>    !["Employees.HomeP end of dataBand1	<b>Employee</b> <b>Phone</b> Nancy Davolio    (206)555-9857 Andrew Fuller    (206) 555-9482 Anne Dodsworth   (71) 555-4444 Steven Buchana   (71) 555-4444 Laura Callahan   (71) 555-4444 Anne Yorke        (71) 555-4444
カード： Field layout Card	dataBand1:DataBand DataSource = Employees detail1_Employees_EmployeeName:Detail Employee      <GetData !["Employees.EmployeeName"]>    <GetData detail1_Employees_HomePhone:Detail Phone          <GetData !["Employees.HomePh end of dataBand1	Employee      Nancy Davolio Phone          (206)555-9857 Employee      Andrew Fuller Phone          (206) 555-9482 Employee      Anne Dodsworth Phone          (71) 555-4444 Employee      Steven Buchana Phone          (71) 555-4444 Employee      Laura Callahan Phone          (71) 555-4444

### データのグループ化

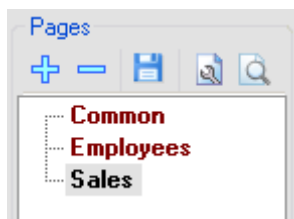
レポートをグループ化して作成するには、グループ化するフィールドを「全フィールド」のツリーから「グループ」タブのテーブルに移動します。



レポートテンプレート	レポート																																								
dataBand1:DataBand DataSource = Sales																																									
header1:Header																																									
<table border="1"><tr><th>Product</th><th>Price</th></tr></table>	Product	Price	<table border="1"><tr><th>Product</th><th>Price</th></tr></table>	Product	Price																																				
Product	Price																																								
Product	Price																																								
groupBand1:GroupBand Group = GetData("Sales.CategoryName")																																									
header11:Header																																									
<table border="1"><tr><td>&lt;GetData("Sales.CategoryName")&gt;</td><td></td></tr></table>	<GetData("Sales.CategoryName")>																																								
<GetData("Sales.CategoryName")>																																									
detail1:Detail																																									
<table border="1"><tr><td>&lt;GetData("Sales.ProductName")&gt;</td><td>&lt;GetData("Sales.UnitPrice")&gt;</td></tr></table>	<GetData("Sales.ProductName")>	<GetData("Sales.UnitPrice")>																																							
<GetData("Sales.ProductName")>	<GetData("Sales.UnitPrice")>																																								
footer1:Footer																																									
<table border="1"><tr><td>&lt;GetTotal("Sales.UnitPrice")&gt;</td></tr></table>	<GetTotal("Sales.UnitPrice")>																																								
<GetTotal("Sales.UnitPrice")>																																									
end of groupBand1																																									
footer11:Footer																																									
<table border="1"><tr><td>&lt;GetTotal("Sales.UnitPrice")&gt;</td></tr></table>	<GetTotal("Sales.UnitPrice")>																																								
<GetTotal("Sales.UnitPrice")>																																									
end of dataBand1																																									
	<table border="1"><tr><th>Product</th><th>Price</th></tr><tr><td><b>Beverages</b></td><td></td></tr><tr><td>Chai</td><td>35,5</td></tr><tr><td>Stout</td><td>105</td></tr><tr><td>Chai</td><td>35,5</td></tr><tr><td>Ipoh Coffee</td><td>50,5</td></tr><tr><td></td><td><b>226,5</b></td></tr><tr><td><b>Confections</b></td><td></td></tr><tr><td>Maxilaku</td><td>2,6</td></tr><tr><td></td><td><b>2,6</b></td></tr><tr><td><b>Dairy Products</b></td><td></td></tr><tr><td>Geitost</td><td>15,8</td></tr><tr><td>Flotemysost</td><td>240</td></tr><tr><td>Courdavault</td><td>62,5</td></tr><tr><td></td><td><b>318,3</b></td></tr><tr><td><b>Grains/Cereals</b></td><td></td></tr><tr><td>Filo Mix</td><td>14</td></tr><tr><td>Filo Mix</td><td>14</td></tr><tr><td></td><td><b>28</b></td></tr><tr><td></td><td><b>575,4</b></td></tr></table>	Product	Price	<b>Beverages</b>		Chai	35,5	Stout	105	Chai	35,5	Ipoh Coffee	50,5		<b>226,5</b>	<b>Confections</b>		Maxilaku	2,6		<b>2,6</b>	<b>Dairy Products</b>		Geitost	15,8	Flotemysost	240	Courdavault	62,5		<b>318,3</b>	<b>Grains/Cereals</b>		Filo Mix	14	Filo Mix	14		<b>28</b>		<b>575,4</b>
Product	Price																																								
<b>Beverages</b>																																									
Chai	35,5																																								
Stout	105																																								
Chai	35,5																																								
Ipoh Coffee	50,5																																								
	<b>226,5</b>																																								
<b>Confections</b>																																									
Maxilaku	2,6																																								
	<b>2,6</b>																																								
<b>Dairy Products</b>																																									
Geitost	15,8																																								
Flotemysost	240																																								
Courdavault	62,5																																								
	<b>318,3</b>																																								
<b>Grains/Cereals</b>																																									
Filo Mix	14																																								
Filo Mix	14																																								
	<b>28</b>																																								
	<b>575,4</b>																																								

### 複数のデータソース

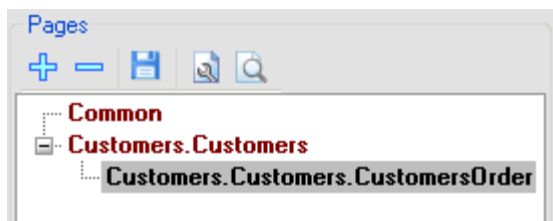
複数のデータソースのデータをレポートに続けて出力できます。これを行うには、テンプレートのツリー構造にデータソースを追加し、必要なフィールドを「表示フィールド」に移動してください。



レポートテンプレート	レポート																						
dataBand1:DataBand DataSource = Employees																							
header1:Header																							
<table border="1"><tr><th>Employee</th><th>Phone</th></tr></table>	Employee	Phone	<table border="1"><tr><th>Employee</th><th>Phone</th></tr></table>	Employee	Phone																		
Employee	Phone																						
Employee	Phone																						
detail1:Detail																							
<table border="1"><tr><td>&lt;GetData("Employees.EmployeeID")&gt;</td><td>&lt;GetData("Employees.HomePhone")&gt;</td></tr></table>	<GetData("Employees.EmployeeID")>	<GetData("Employees.HomePhone")>																					
<GetData("Employees.EmployeeID")>	<GetData("Employees.HomePhone")>																						
end of dataBand1																							
dataBand2:DataBand DataSource = Sales																							
header2:Header																							
<table border="1"><tr><th>Company</th><th>Product</th></tr></table>	Company	Product	<table border="1"><tr><th>Company</th><th>Product</th></tr></table>	Company	Product																		
Company	Product																						
Company	Product																						
detail2:Detail																							
<table border="1"><tr><td>&lt;GetData("Sales.CompanyName")&gt;</td><td>&lt;GetData("Sales.ProductName")&gt;</td></tr></table>	<GetData("Sales.CompanyName")>	<GetData("Sales.ProductName")>																					
<GetData("Sales.CompanyName")>	<GetData("Sales.ProductName")>																						
end of dataBand2																							
	<table border="1"><tr><td>Nancy Davolio</td><td>(206)555-9857</td></tr><tr><td>Andrew Fuller</td><td>(206) 555-9482</td></tr><tr><td>Anne Dodsworth</td><td>(71) 555-4444</td></tr><tr><td>Steven Buchana</td><td>(71) 555-4444</td></tr><tr><td>Laura Callahan</td><td>(71) 555-4444</td></tr><tr><td>Anne Yorke</td><td>(71) 555-4444</td></tr><tr><td><b>Company</b></td><td><b>Product</b></td></tr><tr><td>Futterkiste</td><td>Chai</td></tr><tr><td>Futterkiste</td><td>Stout</td></tr><tr><td>Futterkiste</td><td>Maxilaku</td></tr><tr><td>Futterkiste</td><td>Geitost</td></tr></table>	Nancy Davolio	(206)555-9857	Andrew Fuller	(206) 555-9482	Anne Dodsworth	(71) 555-4444	Steven Buchana	(71) 555-4444	Laura Callahan	(71) 555-4444	Anne Yorke	(71) 555-4444	<b>Company</b>	<b>Product</b>	Futterkiste	Chai	Futterkiste	Stout	Futterkiste	Maxilaku	Futterkiste	Geitost
Nancy Davolio	(206)555-9857																						
Andrew Fuller	(206) 555-9482																						
Anne Dodsworth	(71) 555-4444																						
Steven Buchana	(71) 555-4444																						
Laura Callahan	(71) 555-4444																						
Anne Yorke	(71) 555-4444																						
<b>Company</b>	<b>Product</b>																						
Futterkiste	Chai																						
Futterkiste	Stout																						
Futterkiste	Maxilaku																						
Futterkiste	Geitost																						

## マスター詳細レポート

マスター詳細レポートを作成するには、テンプレートのツリー構造にあるデータソースにもう 1 つデータソースと詳細を追加します。

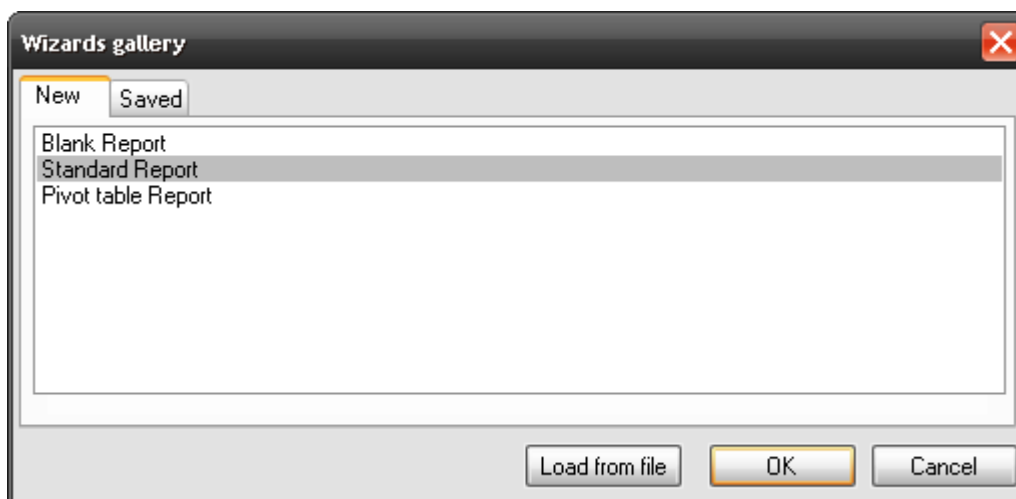


レポートテンプレート	レポート
dataBand1:DataBand DataSource = Customers.Customers	Company Bon App'
detail1_Customers_Customers_CompanyName:Detail	Date Sum
Company <GetData <GetData <GetData <GetData !["Customers.Custome ]	21.03.10 50.00
	15.02.10 14.50
dataBand1_1:DataBand DataSource = Customers.Customers.Customers	64,5
header1_1:Header	Company Chop-suey Chinese
Date Sum	Date Sum
	03.04.10 134.00
detail1_1:Detail	15.01.10 45.45
<GetData <GetData <GetData <GetData !["Customers_C_!["Customers.Custome ]	01.02.10 500.00
	30.12.09 6.03
footer1_1:Footer	685,48
<Get total <Get total !["Customers.Custome ]	Company Maison Dewey
	Date Sum
end of dataBand1_1	06.06.10 66.66
end of dataBand1	66,66

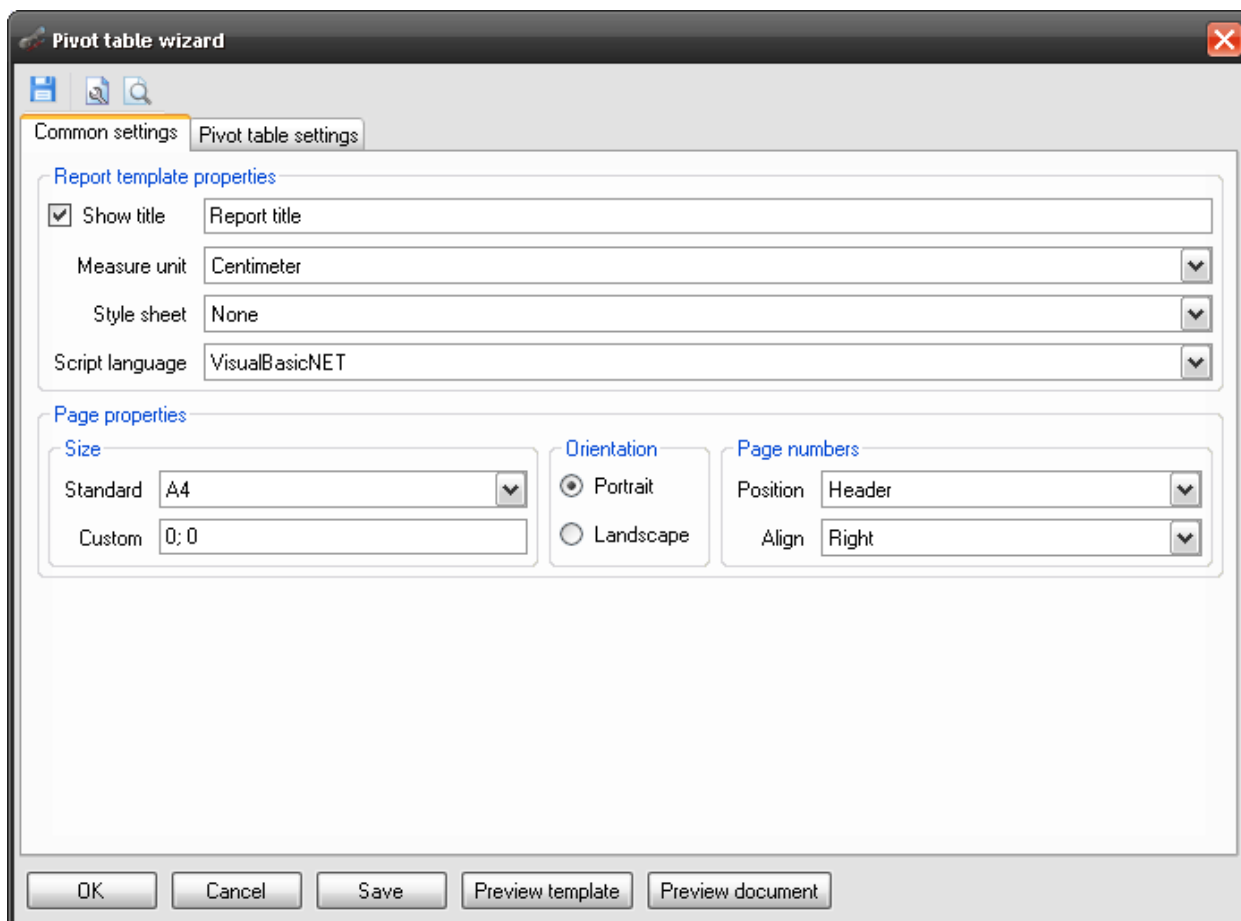
## ピボットテーブル (ピボットテーブル ウィザード) を使ったレポートの作成

ピボットテーブルウィザードは、データソースのデータをピボットテーブルに出力するレポートテンプレートを作成できます。

ウィザードギャラリーダイアログからピボットテーブルウィザードを開くことができます。







### ピボットテーブルウィザードのレポートのプロパティ

ピボットテーブルウィザードの「共通」タブで、レポートの見た目を定義するテンプレートに共通するプロパティを設定できます。

**Document.Title** プロパティを使って「タイトル」フィールドに設定したレポート名が、レポートヘッダーに表示されます。

タイトル	レポートテンプレート	レポート
Report template properties <input checked="" type="checkbox"/> Show title Sales Measure unit Centimeter	reportTitle:Detail <Document.Title>	Sales

レポートの計測単位は、「計測単位」フィールド (Document.MetricUnit プロパティ) で設定します。

レポートのスタイルは、「スタイル」フィールドで設定できます。

スタイルシート	レポートテンプレート	レポート																																																																	
Style sheet Modern1	reportTitle:Detail <Document.Title> pivotTable1:PivotTable	<table border="1"> <thead> <tr> <th colspan="2"></th> <th colspan="3">Sales</th> </tr> <tr> <th colspan="2"></th> <th colspan="3">Company</th> </tr> <tr> <th colspan="2"></th> <th>Futterkiste</th> <th>Karkki Oy</th> <th>Shopy</th> </tr> <tr> <th>Category</th> <th>Product</th> <th>Unit</th> <th>Unit</th> <th>Unit</th> </tr> </thead> <tbody> <tr> <td rowspan="3">Beverages</td> <td>Chai</td> <td>35.5</td> <td>35.5</td> <td></td> </tr> <tr> <td>Ipoh Coffee</td> <td></td> <td>50.5</td> <td></td> </tr> <tr> <td>Stout</td> <td>105</td> <td></td> <td></td> </tr> <tr> <td></td> <td>Total</td> <td>140.5</td> <td>86</td> <td></td> </tr> <tr> <td rowspan="2">Confections</td> <td>Maxilaku</td> <td>2.6</td> <td></td> <td></td> </tr> <tr> <td>Total</td> <td>2.6</td> <td></td> <td></td> </tr> <tr> <td rowspan="3">Dairy Products</td> <td>Courdavault</td> <td>62.5</td> <td></td> <td></td> </tr> <tr> <td>Flotemysost</td> <td>240</td> <td></td> <td></td> </tr> <tr> <td>Geifost</td> <td>15.8</td> <td></td> <td></td> </tr> <tr> <td></td> <td>Total</td> <td>318.3</td> <td></td> <td></td> </tr> </tbody> </table>			Sales					Company					Futterkiste	Karkki Oy	Shopy	Category	Product	Unit	Unit	Unit	Beverages	Chai	35.5	35.5		Ipoh Coffee		50.5		Stout	105				Total	140.5	86		Confections	Maxilaku	2.6			Total	2.6			Dairy Products	Courdavault	62.5			Flotemysost	240			Geifost	15.8				Total	318.3		
		Sales																																																																	
		Company																																																																	
		Futterkiste	Karkki Oy	Shopy																																																															
Category	Product	Unit	Unit	Unit																																																															
Beverages	Chai	35.5	35.5																																																																
	Ipoh Coffee		50.5																																																																
	Stout	105																																																																	
	Total	140.5	86																																																																
Confections	Maxilaku	2.6																																																																	
	Total	2.6																																																																	
Dairy Products	Courdavault	62.5																																																																	
	Flotemysost	240																																																																	
	Geifost	15.8																																																																	
	Total	318.3																																																																	

「スクリプト言語」フィールドは、`Document.ScriptLanguage` プロパティを使用してスクリプト言語を選択できます。

### ピボットテーブル ウィザードのページのプロパティ

ピボットテーブルウィザードのページのプロパティで、ページのプロパティを設定できます。「用紙サイズ」の一覧から用紙サイズが選択され、`Page.PaperKind` プロパティで設定されます。カスタムサイズを設定するには、「用紙サイズ」コンボボックスから「ユーザー定義用紙」を選択し、「ページサイズ」フィールドに値を設定してください。

Size

Paper kind A4

Page size 0;0

ページの向きは「向き」セクションで縦または横に設定できます。

Orientation

Portrait

Landscape

ページ番号を出力するには見出しを使用します。「ページ番号」セクションで見出しの表示モードやテキストの位置合わせを設定できます。

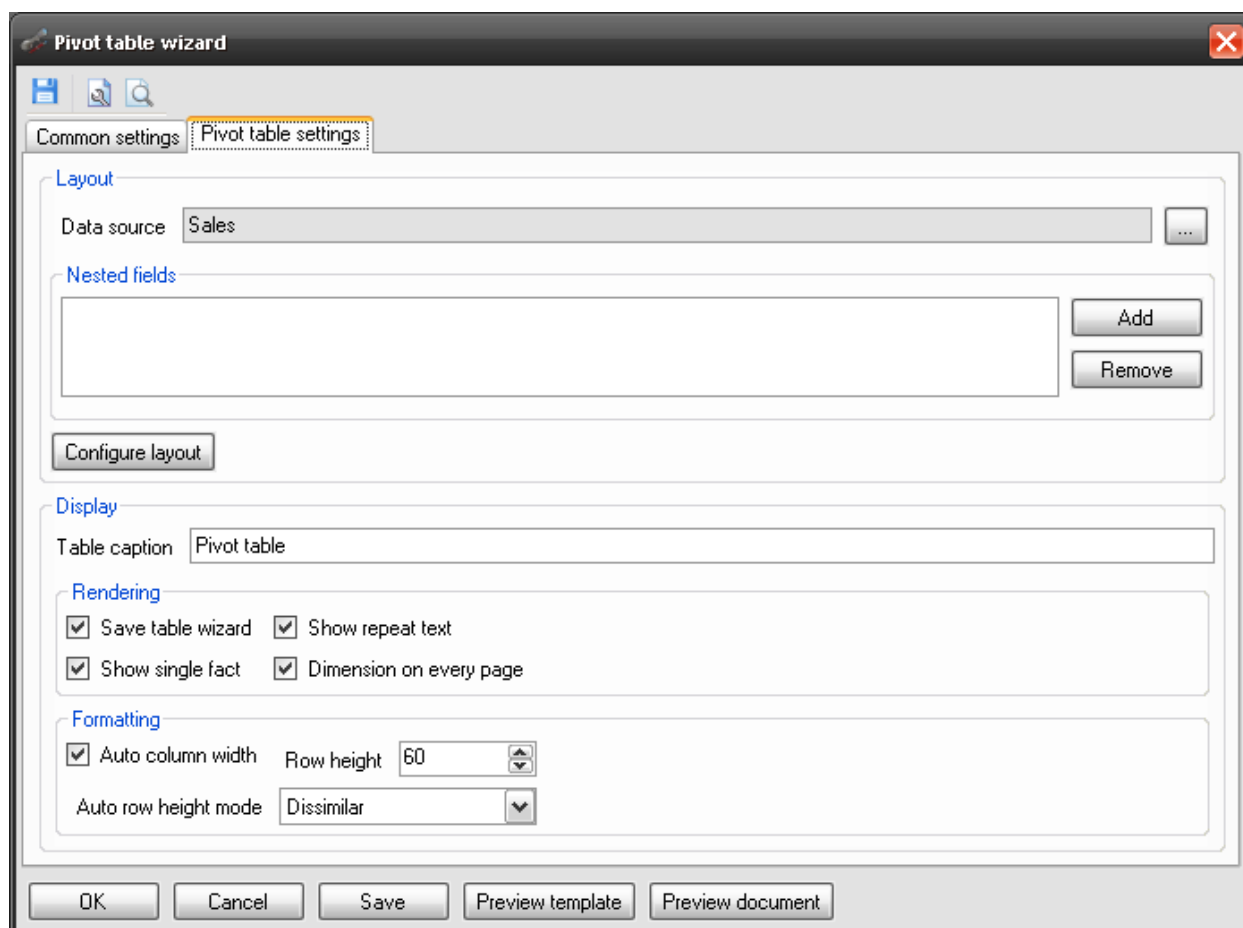
Page numbers

Position Header

Align Right

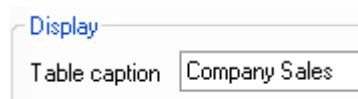
### ピボットテーブルのデータ

ピボットテーブルのデータや表の見た目は、「ピボットテーブル設定」タブの設定で定義されます。



「データソース」フィールドでデータソースを設定します。ピボットテーブルのセルを調節するには、「レイアウトの構成」ボタンでソースフィールドエディタを開いてください（参照：ピボットテーブルのデータ）。

「テーブルのタイトル」フィールドでピボットテーブルのタイトルを設定できます。



Company Sales					
		Company			
		Futterkiste	Karkki Oy	Shopy	Total
Category	Product	Price	Price	Price	Price
Beverages	Chai	35.5	35.5		71
	Ipoh Coffee		50.5		50.5
	Stout	105			105
	Total	140.5	86		226.5
Confections	Maxilaku	2.6			2.6
	Total	2.6			2.6
Dairy Products	Courdavault	62.5			62.5
	Flotemysost	240			240
	Geitost	15.8			15.8
	Total	318.3			318.3

「生成」のプロパティ群で、ピボットテーブルのプロパティ（ShowRepeatText、ShowSingleFact、DementionOnEveryPage プロパティ）を設定できます。



#### Rendering

- Save table wizard
- Show repeat text
- Show single fact
- Dimension on every page

ピボットテーブルのセルサイズは、「形式」セクションで `PivotTable.AutoColumnWidth`、`PivotTable.RowHeight`、`PivotTable.AutoRowHeight` プロパティを設定できます。

#### Formatting

- Auto column width
- Row height
- Auto row height mode



## 本製品の特徴を使用する

### スクリプト

#### 目的とその働き

スクリプトは、ドキュメントのテンプレートに接続された .NET 型の抽象宣言子で、この型は「DocumentScripts」型と呼ばれています。

DocumentScript 型の一部の特定メソッドはレポート生成時に呼び出されます。大半のイベントは生成時に特定のメソッドを呼び出します。メソッドのロジックはスクリプトで定義されます。

わかりやすく言うと、スクリプトはレポートまたはそれぞれのオブジェクトの生成時やデータ処理時に初期化されるイベントハンドラのセットと考えることができます。また、スクリプトは補助的なロジック（例：Windows フォームでのマウスクリック処理）にも使えます。

スクリプトの主な使用目的は、レポートの生成に必要なデータを提供し、そのデータに基づいて決定を下し、生成プロセスを操作することです。

DocumentScripts クラスのメソッドは以下にアクセスします。

- ドキュメントのテンプレートのデータソースのデータ
- ドキュメントのテンプレートのオブジェクトモデル（ドキュメント、ページ、オブジェクト）
- ドキュメントの生成エンジン（現在の生成状態を表すプロパティやその先の生成プロセスに影響を与えるメソッドを持った、生成プロセスを表す特殊なオブジェクト）
- すべてのパブリックデータやアプリケーションのメソッド、アプリケーションが参照するアセンブリ

スクリプトを使うことによって、柔軟にレポートを生成することができます。スクリプトはすべての .NET の特徴や標準/他社のライブラリを使用できる、.NET コードです。

C#、Visual Basic.Net または .NET 環境をサポートするその他の任意の言語をスクリプト言語として使用できます。

注意：スクリプトはすべてソースコードとして保存されるので、レポートを生成するクライアントコンピュータに対象のコンパイラをインストールしておくのはそのためです。C#、Visual Basic. NET や Jscript.NET のコンパイラは Microsoft .NET Framework に含まれています。

#### スクリプトの背景

レポートジェネレータで使用するスクリプトの内部メカニズムです。すべてのスクリプトや式は次の構造のクラスになります。

```
public class Script : PerpetuumSoft.Reporting.Rendering.ReportScriptBase
{
    private PerpetuumSoft.Reporting.DOM.Page page1;
    private PerpetuumSoft.Reporting.DOM.TextBox textBox1;
    ...
    <Document.CommonScript>
    public Script(PerpetuumSoft.Reporting.DOM.Document document,
PerpetuumSoft.Reporting.Components.ObjectPointerCollection dataObjects,
PerpetuumSoft.Reporting.Rendering.RenderEngine engine) : base(document, dataObjects, engine)
    {
        this.page1 = ((PerpetuumSoft.Reporting.DOM.Page)(document.ControlByName("page1")));
        this.textBox1 = ((PerpetuumSoft.Reporting.DOM.TextBox)
(document.ControlByName("textBox1")));
        ...
        this.page1.ManualBuild += new System.EventHandler(this.page1_ManualBuild);
        this.textBox1.Generate += new System.EventHandler(this.textBox1_Generate);
        ...
    }
}
```



```
}  
  
private void page1_ManualBuild(object sender, System.EventArgs e)  
{  
    ...  
}  
  
private void textBox1_Generate(object sender, System.EventArgs e)  
{  
    this.textBox1.Value = <binding expression for Value property of the textBox1>  
    ...  
    <textBox1.GenerateScript>  
}  
    ...  
}
```

下記は、累計計算の 2 番目のレポートで生成されるクラスの内部メカニズムです。

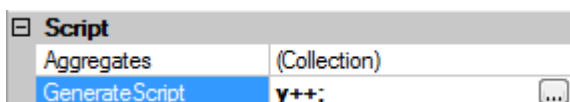
```
namespace PerpetuumSoft.Reporting.ReportScript  
{  
    #line 1 "Document$ImportsString"  
        using System;  
    #line default  
    #line hidden  
    #line 1 "Document$ImportsString"  
        using PerpetuumSoft.Reporting.DOM;  
    #line default  
    #line hidden  
    #line 1 "Document$ImportsString"  
        using PerpetuumSoft.Framework.Drawing;  
    #line default  
    #line hidden  
    public class Script : PerpetuumSoft.Reporting.Rendering.ReportScriptBase  
    {  
        private PerpetuumSoft.Reporting.DOM.TextBox textBox2;  
        private PerpetuumSoft.Reporting.DOM.TextBox textBox3;  
        private PerpetuumSoft.Reporting.DOM.TextBox textBox1;  
        private PerpetuumSoft.Reporting.DOM.Header header1;  
        private PerpetuumSoft.Reporting.DOM.Detail detail1;  
        private PerpetuumSoft.Reporting.DOM.TextBox textBox4;  
        private PerpetuumSoft.Reporting.DOM.DataBand dataBand1;  
        private PerpetuumSoft.Reporting.DOM.Page page1;  
    #line 1 "Document$Common"  
        int sum1 = 0;  
        int sum2 = 0;  
    #line default  
    #line hidden  
        public Script(PerpetuumSoft.Reporting.DOM.Document document,  
PerpetuumSoft.Reporting.Components.ObjectPointerCollection dataObjects,  
PerpetuumSoft.Reporting.Rendering.RenderEngine engine)  
            : base(document, dataObjects, engine)  
        {  
            this.textBox2 =  
((PerpetuumSoft.Reporting.DOM.TextBox)(document.ControlByName("textBox2")));  
            this.textBox3 =  
((PerpetuumSoft.Reporting.DOM.TextBox)(document.ControlByName("textBox3")));  
            this.textBox1 =  
((PerpetuumSoft.Reporting.DOM.TextBox)(document.ControlByName("textBox1")));  
            this.header1 =  
((PerpetuumSoft.Reporting.DOM.Header)(document.ControlByName("header1")));  
            this.detail1 =  
((PerpetuumSoft.Reporting.DOM.Detail)(document.ControlByName("detail1")));  
            this.textBox4 =  
((PerpetuumSoft.Reporting.DOM.TextBox)(document.ControlByName("textBox4")));  
        }  
    }  
}
```

```
        this.dataBand1 =
((PerpetuumSoft.Reporting.DOM.DataBand)(document.ControlByName("dataBand1")));
        this.page1 =
((PerpetuumSoft.Reporting.DOM.Page)(document.ControlByName("page1")));
        this.textBox2.Generate += new System.EventHandler(this.textBox2_Generate);
        this.textBox1.Generate += new System.EventHandler(this.textBox1_Generate);
        this.dataBand1.Generate += new System.EventHandler(this.dataBand1_Generate);
    }
    private void textBox2_Generate(object sender, System.EventArgs e)
    {
#line 1 "textBox2$Value"
        this.textBox2.Value = sum1 - sum2 + (int)dataBand1.DataItem;
#line default
#line hidden
#line 1 "textBox2$Generate"
#line default
#line hidden
    }
    private void textBox1_Generate(object sender, System.EventArgs e)
    {
#line 1 "textBox1$Value"
        this.textBox1.Value = dataBand1.DataItem;
#line default
#line hidden
#line 1 "textBox1$Generate"
#line default
#line hidden
    }
    private void dataBand1_Generate(object sender, System.EventArgs e)
    {
#line 1 "dataBand1$Generate"
        if (!Engine.IsDoublePass)
            sum1 += (int)dataBand1.DataItem;
        else
            sum2 += (int)dataBand1.DataItem;
#line default
#line hidden
    }
}
}
```

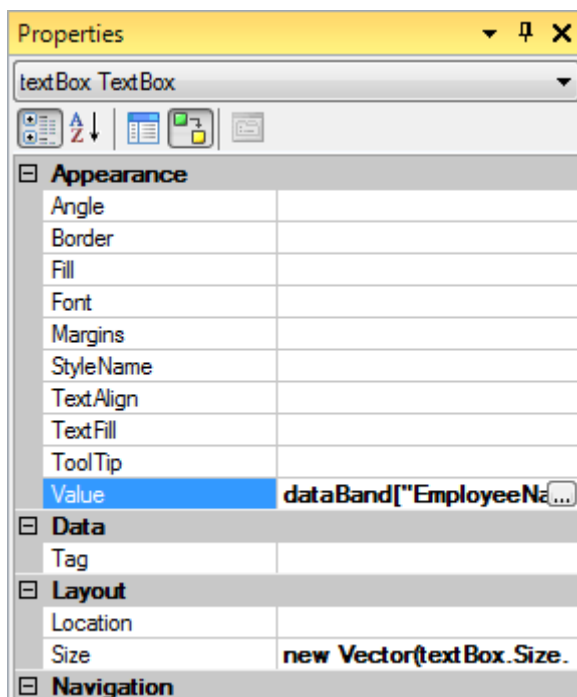
## スクリプトの設定方法

スクリプトはテンプレートのデザイナー（レポートデザイナー）で設定/編集され、テンプレートと一緒に保存されます。スクリプトはオブジェクトの文字列のプロパティにテキストとして保存され、プロパティグリッドで取得できます。

テンプレートのすべてのオブジェクトには「Script」のプロパティ群があり、そこで色々な種類のスクリプトを設定できます。



特殊な型「バインド」のスクリプトは、プロパティグリッドの「バインド」タブで設定されます。



「バインド」に素早くアクセスするには、バインド バーの専用の要素を使用できます。

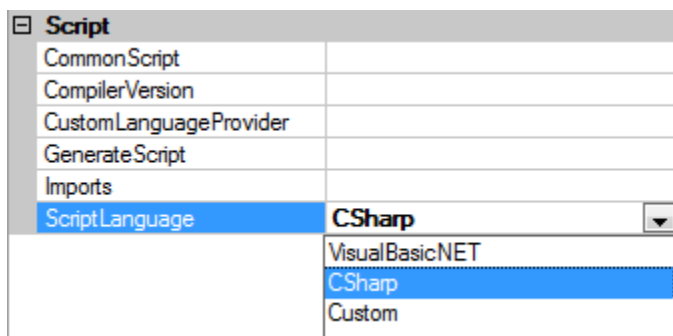


スクリプトを編集するためにデザインされた、構文の強調表示やコードコンプリートが付いた、特別な「スクリプトエディタ」があります。

### スクリプト言語とコンパイラバージョンの設定

スクリプトの記述には、VB.NET、C# またはその他の任意の .NET プログラミング言語（J#、マネージド c++ やその他）を選択できます。

言語は、Document.ScriptLanguage プロパティで設定します。VisualBasic.NET が規定の言語です。



その他の任意の言語を使用するには、そのコンパイラの位置を指定する必要があります（例：System.CodeDom.Compiler.CodeDomProvider 派生型の名前）。

必要に応じて、Document.CompilerVersion プロパティを使用して、コンパイラバージョンを設定できます。例えば、C# v. 3.5 を使用するには、Document.CompilerVersion = "3.5" に設定してください。

注意：スクリプトはレポート生成時にコンパイルされます。このテンプレートを使ってレポートを生成するすべてのコンピュータに、対応バージョンのコンパイラをインストールするのはそのためです。

### スクリプトの種類

スクリプトは、3つのグループに分類できます。



## イベントハンドラ

イベントハンドラは、イベント発生時に実行される一連の演算子を表したスクリプトです。この種のスクリプトはすべて、プロパティグリッドの「Script」のプロパティ群にあります。

Script	
Aggregates	(Collection)
GenerateScript	
GetDataScript	

プロパティ	オブジェクト	説明
GenerateScript	すべてのオブジェクト	オブジェクトの生成前に実行されるコード
ManualBuildScript	Page	ページ内容を生成するコード。スクリプトが設定された場合、標準の生成メカニズムを利用することはできません。
CommonScript	Document	式の主要部分のコード。ここに、グローバル変数、関数、プロシージャを指定できます。
GetDataScript	DataBand、CrossBand	データソースのデータを取得する時に実行される式のコード。

## プロパティのバインド

プロパティのバインドは、オブジェクトのプロパティに設定する式を示したスクリプトで、その計算結果がそのプロパティの値として設定されます。式は、プロパティグリッドの「バインド」タブの大半のオブジェクトのプロパティに設定できます。式は **GenerateScript** プロパティを呼び出す前で、オブジェクトが生成される前に計算されます。式の計算順は決められません。異なるプロパティに、計算順に基づいて式のロジックを記述しないようお勧めするのはそのためです。

例：

```
textBox.Value = a + b
```

```
textBox.Style = (textBox.Value == 0) ? "Zero" : "Normal"
```

記述に間違いはありませんが、**textBox.Style** プロパティの式の計算結果が無効になってしまう場合があるので次のように設定します。

```
textBox.Style = (a + b == 0) ? "Zero" : "Normal"
```

## 各種 Expressions プロパティ

各種 Expressions プロパティは、特定の型の値を取得する特別なプロパティです。各種 Expressions プロパティはプロパティのバインドに似ていますが、どのプロパティとも合致しません。式で取得した結果が使用されますが、プロパティには格納されません。

各種 Expressions プロパティはプロパティグリッドで設定され、スクリプトエディタを使用して設定できます。各種 Expressions プロパティのコレクションを示したプロパティがあります。

各種 Expressions プロパティ：

**GroupBand.GroupExpression** プロパティ

グループ順を設定します。式はグループキーを取得します（キーが等しい場合、行は同じグループになります）。計算された値はプロパティに設定されませんが、内部の生成アルゴリズムで使用されます。

**GroupBand.SortExpression** プロパティ

ソート順を設定します。

**DataBand.FilterExpression** プロパティ

データのフィルタ処理のルールを設定します。

**DataBand.Sort** プロパティ

ソート条件のコレクション。各ソート条件は、**Expression** プロパティ（データを並べ替えるキーを取得するための式を設定します）を持つオブジェクトになります。

**DataBand.Totals** プロパティ

集計関数の式のコレクション。

これらの式は生成の内部ロジックとして計算されるので、式を呼び出す時期や計算順は設定されません。仮の計算順/計算時期に基づいてロジックを作成しないでください。原則としてこのような式はソースデータの値が基になります。もちろん、ドキュメント全体や **DocumentScripts** スクリプトクラスの初期化のために設定したスクリプトが最初に実行されるだろうと考えることはできます。

### ドキュメントオブジェクト モデル

テンプレートのすべてのオブジェクトやドキュメントそのものを **Name** プロパティに設定された名前やドキュメントの名前を使用して参照できます。

すべてのパブリックメソッドや要素の型のプロパティを使用できます（型の詳細はクラスリファレンスを参照してください）。それは、テンプレートに追加された要素の型や名前と一致するフィールドが **DocumentScripts** スクリプトクラスに追加されるため使用可能となります。これらのフィールドはレポート生成プロセスで使用されるオブジェクトのインスタンスによって初期化されます。例えば、テンプレートに「**textBox1**」というテキストボックスを追加した場合、プライベート型のテキストボックスの **textBox1** フィールドが **DocumentScripts** クラスに追加されます。そして生成開始前に、**TextBox** オブジェクトの特定のインスタンスや生成プロセスに使用されるまさしくインスタンスによって初期化されます。任意のスクリプトでこの要素を参照し、例えば **textBox1.Value = 1** を設定できます。このメカニズムはビジュアルデザイナーに追加された各要素に適用されます。ビジュアルデザイナーに要素を追加することにより、対象のフィールドの記述をコードに追加します。

### メソッドとプロパティ

テンプレートのデザイン時やスクリプトの記述時に作成される「**DocumentScripts**」クラスは、特別なクラス：**PerpetuumSoft.Reporting.Rendering.ReportScriptBase** から派生されます。この型には、スクリプトで使用可能な便利なメソッドやプロパティが多数あります。上述したように、スクリプトは **Rendering.ReportScriptBase** から継承されたクラスのコードの一部となります。（クラスリファレンスの型の記述に関する詳細はクラスリファレンスを参照してください。）

### レポート生成時のスクリプトの動作

未定

### 型、基本型およびそのメソッドへのアクセス

#### 基本クラスへのアクセス

スクリプトの基本クラスの主要なプロパティやメソッドは次の通りです。

プロパティまたはメソッド	説明
Now	データと生成開始時間を取得します
PageCount	ドキュメントのページ数を取得します
PageNumber	現在のページ番号を取得します
ColumnNumber	現在の列番号を取得します
DataObjects	関連付けられたデータソースのコレクションを取得します
Document	現在生成したドキュメントのテンプレートを取得します
Engine	関連付けられた <b>RenderEngine</b> を取得します
Group	データをグループ化した時の現在のグループを取得します
GetData	指定したデータメンバのデータを返します
GetParameter	指定した名前のパラメータを返します
GetTotal	フィールドごとの合計を取得します。参照の位置に応じて、グループの合計/入力データの総計を取得できます。
GetRunningTotal	その名前の累計を取得します

**PerpetuumSoft.Reporting.Rendering.RenderEngine** 型の **Engine** プロパティは、レポート生成エンジンを表す複雑なオブジェクトなので少し説明します。

**RenderEngine** 型の主要なプロパティやメソッドは次の通りです。

プロパティまたはメソッド	説明
ColumnsCount	列数を取得します
LineNumber	現在の行番号を取得します
PageCount	ドキュメントのページ数を取得します
PageNumber	現在のページ番号を取得します
現在生成したページに設定されているデータ：	
PageRenderArea	そのページで利用可能な余白スペースを定義する四角形を取得および設定します
Caret	生成したページの現在位置を取得および設定します
UsedHeight	使用したページの高さを取得します
UsedWidth	使用したページ幅を取得します
FreeHeight	余白ページの高さを取得します
FreeWidth	余白ページの幅を取得します
データソースおよびパラメータ：	
Objects	データソースのコレクションを取得および設定します
Parameters	パラメータのコレクションを取得します
生成したドキュメントおよび現在のページ：	
ProductionDocument	最終ドキュメントを取得します
ProductionPage	最終ドキュメントの現在のページを取得します
使用したドキュメントのテンプレートや現在使われているテンプレートページのデータ：	
TemplateDocument	ドキュメントのテンプレートを取得します
TemplatePage	テンプレートの現在のページを取得します

生成プロセスの現状を知るだけでなく、対象の値を変更することで生成プロセスを操作することができます。

メソッド	説明
生成プロセスを操作するために使用する <b>RenderEngine</b> のメソッド	
NewColumn	新しい列を作成します
NewPage	新しいページを作成します
CancelRender	生成をキャンセルします

### データへのアクセス

スクリプトは、レポートコントロールと特定のデータとの関連付けによく使われます。さまざまなプロパティの設定は、ソースとなるデータを使ってプロパティのバインドで行われます。ソースフィールドの現在値を取得するには、**GetData** メソッドを使用します。“.”記号で区切られたデータのパスをパラメータとして指定します。

データバンドに追加されたコントロールは、**DataBand.DataSource** プロパティに設定したのと同じソースフィールドを頻繁に表示します。この場合、**DataBand** のインデクサを使った方が便利です。

**GetData**, **DataBand[]** は現在行の値を取得します。データバンドはレポート生成時にカーソルを行から行へと移動します。スクリプトエディタのデータソースツリーからフィールドをドラッグ&ドロップすることで、そのようなスクリプトを設定できます。

### 生成エンジンへのアクセス

未定

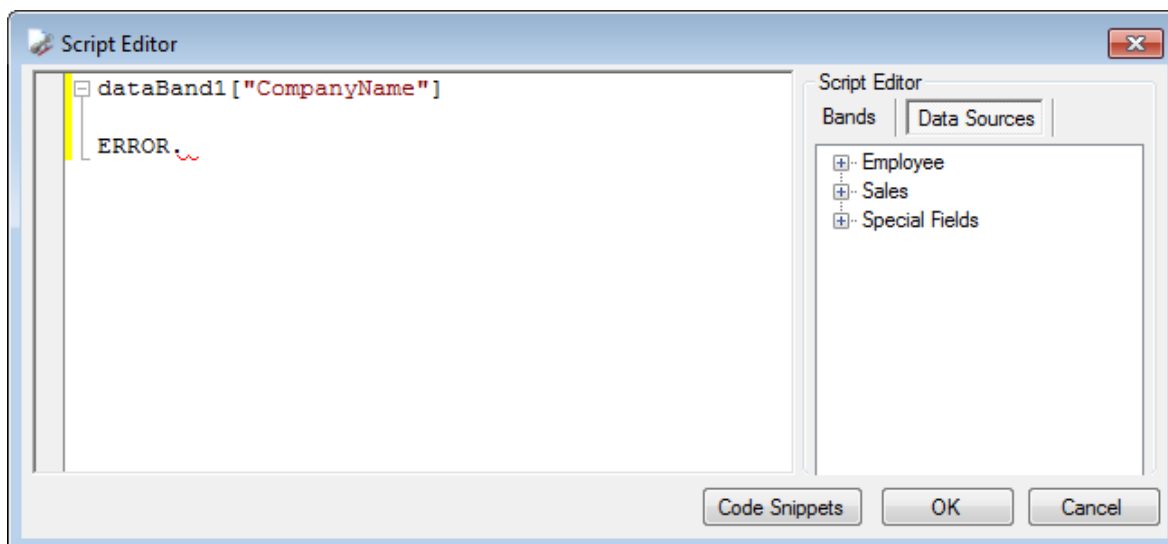
### その他のアセンブリへのアクセス

未定

### エラー

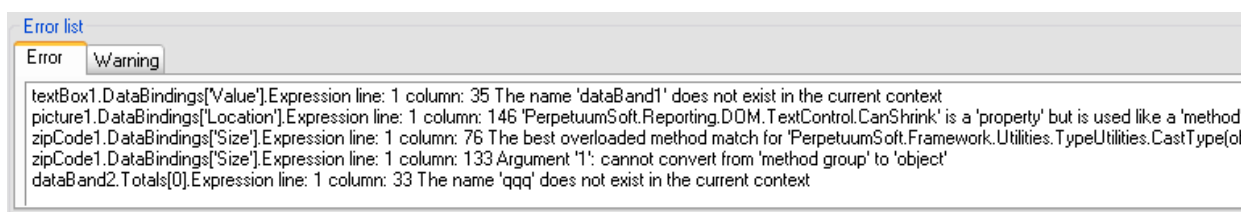
スクリプトの記述時に、構文エラー、セマンティックエラー、スクリプトのロジック内のエラーなどさまざまなエラーを起こしてしまう場合があります。

スクリプトエディタでの構文エラーは、構文が強調表示されてエラーに下線が付くのですぐわかります。



ドキュメントを生成しようとしてスクリプトをチェックする時に、構文エラーやセマンティックエラーが特定のエラー一覧にリストアップされます。

([レポート]->[スクリプトのチェック]メニューまたはツールバーの「スクリプトのチェック」ボタンをクリックしてスクリプトを確認することができます)



エラー一覧にはエラーが含まれているバンド名とその説明が記述されます。リスト項目をダブルクリックすると、スクリプト/式を編集するスクリプトエディタが表示されます。

スクリプトをデバッグするには、。例えば、**System.Windows.Forms.MessageBox(var1.ToString())** ダイアログを使用して、コントロールポイントに変数の値を表示できます。

スクリプトに高度なロジックは使わずに基本的なロジックだけを使用することを強くお勧めします。スクリプトが複雑すぎる場合は、レポートテンプレートのコンセプトを見直す必要があります。

高度な計算ロジックをデバッグしなければならない場合の回避策としては、そのようなロジックをすべて関数内に分離することです。これらの関数はアプリケーションに入れられ、スクリプトはアプリケーションを参照して必要な計算を行います。使用している開発環境のデバッガをアプリケーションで使用できます。テンプレートをアプリケーションから独立させなければならない場合、ロジックをデバッグした後で関数をスクリプトコードに戻すことができます。

## スクリプトの代表的な役割と解決法

### データ処理

未定

### 表示/非表示の操作

現在の行番号を表示するサンプル。奇数行だけが表示されます。

レポートテンプレート :

dataBand1:DataBand
detail1:Detail
<dataBand1.LineN
end of dataBand1

`dataBand1.InstanceCount = 100` (このプロパティは、レポートにバンド内容を表示する回数を設定します)

`textBox1.Value = dataBand1.LineNumber` (現在のページ番号を出力)

`detail1.Visible` (動的なプロパティのバインド) = `"dataBand1.LineNumber % 2 == 0 ? false : true"`

つまり、現在のページが奇数の場合に **Detail** オブジェクトが表示されます。

レポート :

1
3
5
7
9
11
13

(Detail ではなく) **TextBox** オブジェクトを表示/非表示にした場合、奇数行の間に空白行 (非表示の **Textbox** オブジェクトを持つ **Detail** オブジェクト) が表示されます。

### 位置の操作

オブジェクトの位置を示すサンプル

レポートテンプレート :

dataBand1:DataBand
detail1:Detail
<dataBand1.LineN
data
end of dataBand1

`dataBand1.InstanceCount = 100` (このプロパティはレポートにバンド内容を表示する回数を設定します)

`textBox1.Value = dataBand1.LineNumber` (現在の行番号を出力します)

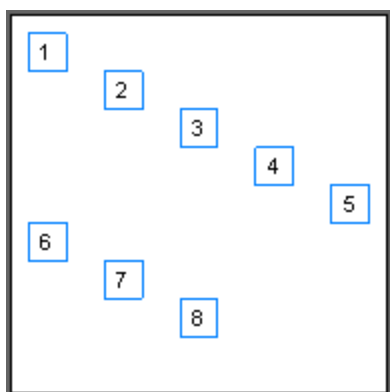
`textBox1.Location` (動的なプロパティのバインド) =

`"PerpetuumSoft.Framework.Drawing.Vector(0.25f+1f*((dataBand1.LineNumber - 1) % 5),0f).ConvertUnits(Unit.Centimeter,Unit.InternalUnit)"`

**TextBox** オブジェクトを 5 つづつ右に 1 cm 移動させて表示します。

`ConvertUnits` メソッドは計測単位を変換します。

レポート :



### アプリケーション ロジックへのアクセス

アプリケーションのアセンブリのすべてのオブジェクトはスクリプトからアクセスできます。アプリケーションでのメソッドの使い方を示したサンプルは **HostingApplicationExample** フォルダにあります。「Hello from HostingApplicationExample.exe」という文字列を取得する **GetString()** メソッドがメインフォームのクラスで設定されています。

```
public static string GetString()  
{  
    return "Hello from HostingApplicationExample.exe";  
}
```

フォーム クラスにアクセスするには、フォームが置かれている名前空間を **Document.Imports** プロパティに追加するだけです。

Script	
CommonScript	
CompilerVersion	
CustomLanguageProvider	
GenerateScript	
Imports	HostingApplicationExample ...
ScriptLanguage	CSharp

textBox1 オブジェクトの Value プロパティに、**GetString()** メソッドが呼び出されます。

```
<Form1.GetString()>
```

### データソース

#### データソースのクラス

SharpShooter Reports のデータソースとして使用可能なクラスは次の通りです。

ADO.NET のオブジェクト - **System.Data.DataSet**、**System.Data.DataView**、**System.Data.DataTable**

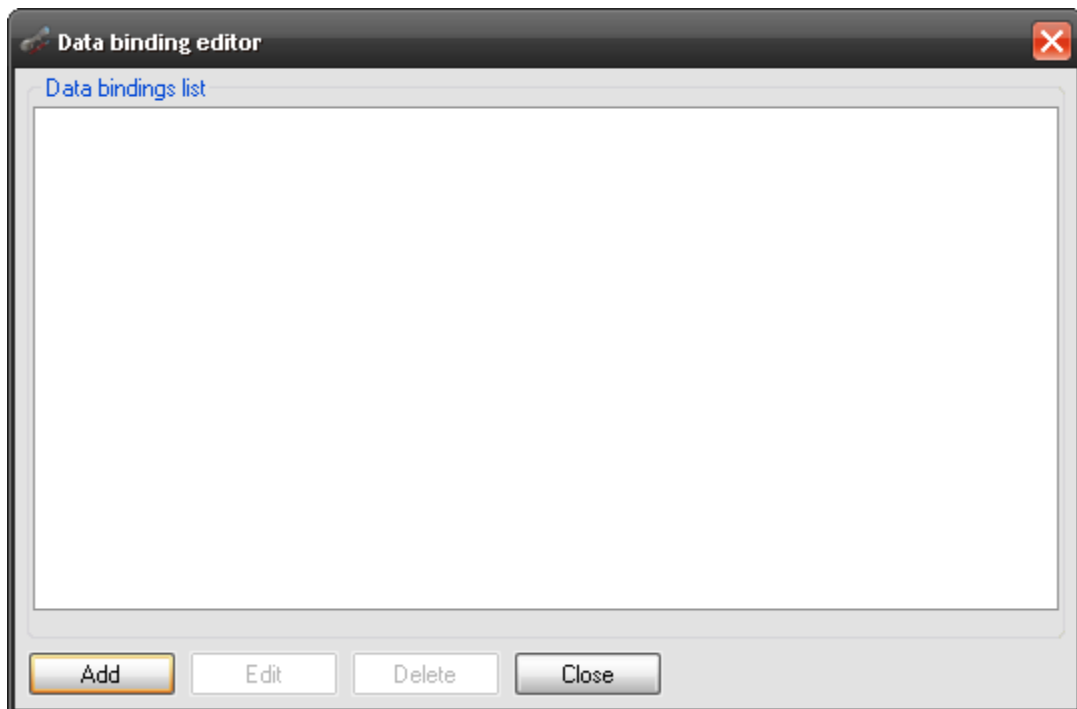
**System.ComponentModel.IListSource** または **System.Collections** インターフェイスを実装するカスタムクラス（ビジネスオブジェクト）。**IEnumerable** (**IEnumerable** インターフェイスは多くの標準クラスで実装されます（例：**System.Array**、**System.Collections.ArrayList**、**System.Collections.CollectionBase** やその他）

上記の一覧に含まれていないその他のカスタムクラス（ビジネスオブジェクト）やこのオブジェクトのすべてのプロパティをデータとして利用できます。

#### データソースの設定

レポートのデータソースは **ReportManager** クラスの **DataSources** プロパティに設定します。このプロパティには **PerpetuumSoft.Reporting.Components.ObjectPointerCollection** 型があります。このクラスは文字列キーにバインドされたオブジェクトのコレクションです。

**DataSources** プロパティのエディタを使ってレポートのデータソースを設定するには、**ReportManager** オブジェクトを選択し、プロパティグリッドの **DataSources** プロパティのボタンをクリックすると、下図のようなデータバインドエディタが表示されます。



「編集」ボタンは、データバインドリストで選択したデータソースのエディタを開きます。「削除」ボタンは、選択したデータソースを削除します。「閉じる」ボタンはデータバインドエディタを閉じます。

新しいデータソースを追加するには、「追加」ボタンをクリックして下図のエディタを開きます。



「オブジェクト名」フィールドは、レポートに使用するデータソース名を入力するために使用します。「値」フィールドは、アプリケーションで使用するオブジェクト名（データソースそのもの）を選択するために使用します。その結果、下記コードに **InitializeComponent()** メソッドが追加されます。

(C#)

```
this.reportManager.DataSources =
    new PerpetuumSoft.Reporting.Components.ObjectPointerCollection(
        new string[] {"AccountsDataSet", "CustomersByCity"},
        new object[] {this.accountsDataSet, this.customersByCity});
```

(VB)

```
Me.reportManager.DataSources =
    New PerpetuumSoft.Reporting.Components.ObjectPointerCollection(
        New String() {"AccountsDataSet", "CustomersByCity"},
        New Object() {Me.accountsDataSet, Me.customersByCity })
```

「値」リストにデータソースを表示させるには、**System.ComponentModel.Component** クラスから継承していなければなりません。



コードからデータソースを追加するには、`ObjectPointerCollection.Add` メソッドまたは `ObjectPointerCollection.Item` プロパティを使用できます。

To add data source from code, you can use Add method of the ObjectPointerCollection class of the Item property:

(C#)

```
reportManager.DataSources.Add("DataSourceName", dataSource);
```

(VB)

```
reportManager.DataSources.Add("DataSourceName", dataSource)
```

または

(C#)

```
reportManager.DataSources["DataSourceName"] = dataSource;
```

(VB)

```
reportManager.DataSources("DataSourceName ") = dataSource
```

`Add` メソッドは 2 つのパラメータ (レポートで使用可能なデータソース名とデータソースそのもの) を受け取ります。 `Item` プロパティを使う場合も同様にデータソース名とデータソースそのものを設定できます。

レポートのデータソースは、直接テンプレートでレポートを使うことによって読み込めます。

### ADO.NET オブジェクトの使用

ADO.NET をデータソースとして使用したサンプルについて説明します。「**Authors**」と「**Books**」という 2 つのテーブルを持つデータソースがあります。「**Authors**」には作者名、「**Books**」には本のタイトルが格納されています。「**Authors**」テーブルには、**AuthorID** (主キー) と **Name** (作者名) の 2 つのフィールドがあります。「**Books**」テーブルには、**BookID** (主キー)、**Name** (本のタイトル)、**AuthorID** (作者キー)、**Price** (値段) があります。このサンプルは、**ADODataSource** フォルダにあります。

### ビジネスオブジェクトの使用

上述したように、カスタムなデータソースでデータを表すには、**System.Collections.IEnumerable** または **System.ComponentModel.IListSource** インターフェイスを実装してください。( **IEnumerable** インターフェイスは **System.Array**、**System.Collections.ArrayList**、**System.Collections.CollectionBase** やその他の多数の標準クラスで実装されています)。

このサンプルは **UserDataSource** フォルダにあります。データソースの設定やデータソースを **ReportManager** クラスの **DataSources** プロパティのコレクションに追加するのは **Init()** メソッドで行われています。

**IListSource** インターフェイスは 2 つのメンバを定義します。**IlistSource** インターフェイスには **SharpShooter Reports** の動作に関係のない **ContainsListCollection {get;}** プロパティと、**GetList()** メソッドがあります。**GetList()** メソッドは、順次に **IEnumerable** インターフェイスを実装する **System.Collections.Ilist** インターフェイスの参照を取得します。

つまり、カスタムなデータの処理は **System.Collections.IEnumerator** インターフェイスの参照を取得する **IEnumerable.Getenumerator()** メソッドで行われます。このインターフェイスを使用してコレクションの項目をすべて表示できます。**IEnumerator** インターフェイスは **Current** プロパティ、**IEnumerator.MoveNext()** メソッド、**IEnumerator.Reset()** メソッドを定義します。**Current** プロパティは、コレクションの現在のオブジェクトを表します。**MoveNext()** メソッドは、コレクションの次のオブジェクトに移動し、成功した場合は **True** を、現在のオブジェクトがコレクションの末尾の場合は **False** を返します。**Reset()** メソッドは、列挙子を初期位置、つまりコレクションの最初のオブジェクトの前に設定します。





プロパティはすべて、カスタムなデータ配列に格納されたオブジェクトのデータフィールドとして利用できません。**IListSource** または **IEnumerable** インターフェイスを実装する型のプロパティの場合はカスタムなデータ配列と見なされ、カスタムなデータソースを使用してマスター詳細のリレーションを持った複雑なレポートを作成できます。

データソースが **IListSource** も **IEnumerable** インターフェイスも実装しない場合、プロパティはすべてデータとして取得できるので、テーブルからレコードを 1 つ実装できます。

では、「**ADO.NET** オブジェクトの使用」で取り上げた作者と本のデータソースのためにカスタムなデータソースを作成します。

一冊の本に関する情報を格納するために、次のクラスを作成します。

```
public class Book
{
    public Book()
    {
    }
    private string name = string.Empty;
    public string Name
    {
        get
        {
            return name;
        }
        set
        {
            name = value;
        }
    }
    private decimal price;
    public decimal Price
    {
        get
        {
            return price;
        }
        set
        {
            price = value;
        }
    }
}
```

このクラスには、レポート作成時に取得可能なプロパティが 2 つあります。

では、**BookCollection** にこのクラスを追加して、**IEnumerable** インターフェイスを実装する **Book** クラスのインスタンスを格納します。データを **ArrayList** クラスのインスタンスに格納します。更に、2 つのメソッド: **Add()** メソッド (項目を **BookCollection** に追加します) と **GetEnumerator** メソッドの実装、そして **IEnumerable** インターフェイスの実装をこのように行います。このクラスのコードは次の通りです。

```
public class BookCollection : IEnumerable
{
    private ArrayList list = new ArrayList();

    public BookCollection()
    {
    }
    public void Add(Book b)
    {
        list.Add(b);
    }
}
```



```
    }  
    public IEnumerator GetEnumerator()  
    {  
        return list.GetEnumerator();  
    }  
}
```

更に、1 作者の情報を格納するクラスを作成する必要があります。作者の本を多数格納するために、本が格納されるプロパティはコレクションでなければなりません。従って、カスタムなデータソースにマスター詳細のリレーションを実現します。

```
public class Author  
{  
    public Author()  
    {  
    }  
    private string name = string.Empty;  
    public string Name  
    {  
        get  
        {  
            return name;  
        }  
        set  
        {  
            name = value;  
        }  
    }  
    private BookCollection books = new BookCollection();  
    public BookCollection Books  
    {  
        get  
        {  
            return books;  
        }  
    }  
}
```

作者を格納するコレクションを作成しなければなりません。例えば、このクラスは **System.Collections.CollectionBase**（厳密に型指定されたコレクションの基本クラスで、**IEnumerable** インターフェイスを実装します）から継承することができます。

```
public class AuthorCollection : CollectionBase  
{  
    public Author Add(Author value)  
    {  
        base.List.Add(value as object);  
        return value;  
    }  
    public void Remove(Author value)  
    {  
        base.List.Remove(value as object);  
    }  
    public void Insert(int index, Author value)  
    {  
        base.List.Insert(index, value as object);  
    }  
    public Author this[int index]  
    {  
        get  
        {  
            return (base.List[index] as Author );  
        }  
    }  
}
```

```

    }
    set
    {
        base.List[index] = value;
    }
}
}

```

このサンプルでは、このような機能があることを実装するためだけに **BookCollection** クラスが **IEnumerable** インターフェイスを実装しています。もちろんコレクションを配列や **ArrayList** クラスのインスタンスに格納することはできますが、厳密に型指定されたコレクション（**CollectionBase** から継承されたクラス）を使用することをお勧めいたします。

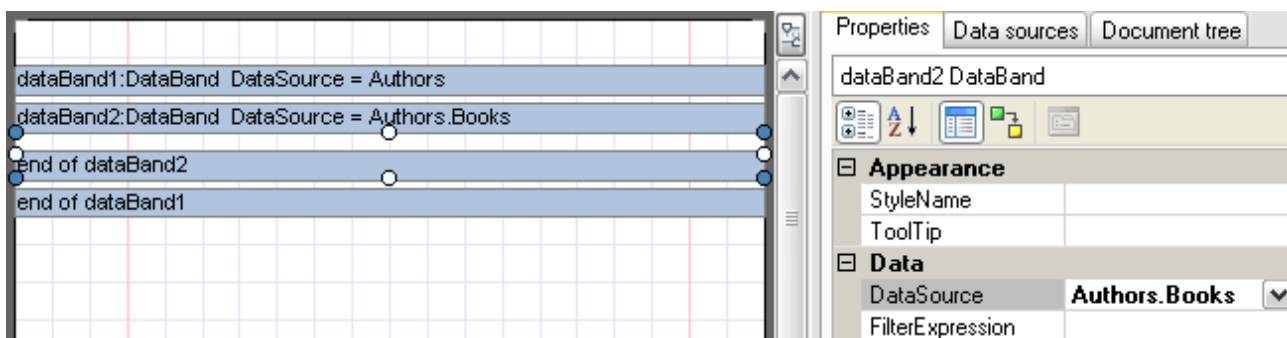
最後に、データソースを設定するメソッドを記述し、それを **ReportManager** の **DataSouces** のコレクションに追加する必要があります。

```

private void Init()
{
    Author a = new Author();
    a.Name = "Kent Beck";
    Book b = new Book();
    b.Name = "Extreme Programming";
    b.Price = 7.45M;
    a.Books.Add(b);
    b = new Book();
    b.Name = "Extreme Programming Explained";
    b.Price = 11.33M;
    a.Books.Add(b);
    authorDS.Add(a);
    a = new Author();
    a.Name = "Craig Larman";
    b = new Book();
    b.Name = "Applying UML and Patterns";
    b.Price = 15.80m;
    a.Books.Add(b);
    authorDS.Add(a);
    reportManager1.DataSources.Add("Authors", authorDS);
}

```

マスター詳細レポートテンプレートを作成するには、レポートデザイナーで **DataBand** 内に別のデータバンドを 1 つ格納し、**dataBand1.DataSource** プロパティを「**Authors**」に、**dataBand2.DataSource** プロパティを「**Authors.Books**」に設定します。



データソースのフィールドにアクセスするには、**textBox.Value = <DataBand name>["<Field name>"]** を設定します。

dataBand1:DataBand DataSource = Authors	
detail2:Detail	
	<dataBand1["Name"]>
dataBand2:DataBand DataSource = Authors.Books	
detail1:Detail	
	<dataBand2["Name"]   <dataBand2["Price"]>
end of dataBand2	
end of dataBand1	

このサンプルは、UserDataSource フォルダにあります。

### ビジネスオブジェクトの一般的ではない使い方

(オブジェクトのプロパティではなく) 動的に計算されたプロパティをフィールドとして取得できるように、カスタムなデータソース処理の標準メカニズムを変更しなければならない場合、データソースの 1 レコードを表すオブジェクトには **ICustomTypeDescriptor** インターフェイスを、これらのオブジェクトを格納するコレクションには **ITypedList** インターフェイスを実装しなければなりません。

**CustomTypeDescriptorExample** サンプルを例にあげて説明します。世界中に部品を販売する企業向けのプログラムを開発する場合、基準となる通貨を各国の通貨に換算して価格表示しなければなりません。

**Currency** クラスは、あらゆる通貨情報を格納するために使用します。このクラスには名前と為替レートを設定する 2 つのプロパティがあります。**SystemCurrencies** という静的な配列がこのクラスで設定されています。この配列は、このシステムで使用されるすべての通貨を格納するために使用します。このサンプルでは、配列には限定的な値が設定されていますが、実際のアプリケーションではこの配列をデータベースから読み込みます。

**Path** クラスは、部品のレコードを 1 つ格納するために使用します。このクラスは **ICustomTypeDescriptor** インターフェイスを実装します。このクラスの動的プロパティのコレクションを生成する静的メソッドのコードは次の通りです。

```
public static PropertyDescriptorCollection GetPartProperties()
{
    PropertyDescriptorCollection props = new PropertyDescriptorCollection(null);
    foreach(Currency c in Currency.SystemCurrencies)
    {
        props.Add(new CurrencyPropertyDescriptor(c));
    }
    props.Add(TypeDescriptor.CreateProperty(typeof(Part), "Name", typeof(string)));
    return props;
}
```

動的なプロパティのコレクションを取得するために、まさしくこのメソッドはインターフェイスのメソッドによって呼び出されます。

```
ICustomTypeDescriptor.GetProperties(Attribute[] attributes)
ICustomTypeDescriptor.GetProperties()
```

コードからお分かりになるように **CurrencyPropertyDescriptor** クラスの冒頭で、**Currency.SystemCurrencies** の配列からすべての通貨型のインスタンスが作成されてから部品名が追加されます。

**CurrencyPropertyDescriptor** クラスは、クラスのプロパティ定義の抽象化を提供する **PropertyDescriptor** クラスから継承されます。**CurrencyPropertyDescriptor** クラスの **PropertyType** プロパティと **ComponentType** プロパティの実装に注目してください。**PropertyType/ComponentType** プロパティは、プロパティ値を取得/設定する **GetValue/SetValue** メソッドと同じように、このクラスを実装するプロパティ型/クラス型を取得します。

最後に、部品のコレクションを格納するために使用される、**PartCollection** クラスについて説明します。このクラスは **ITypedList** インターフェイスを実装します。このインターフェイスには 2 つのメソッドがあります。**GetListName** メソッドはリスト名を取得し、**GetItemProperties** メソッドはリストのオブジェクトの動的なプロパティを定義する **PropertyDescriptor** オブジェクトの配列を取得します。このメソッドは **Part.GetPartProperties()** 静的メソッドの処理結果を取得するだけです。

従って、各国の通貨で価格を表示するレポートテンプレートを作成する必要があります。

### XML ファイルをデータソースとして使用する

XML ドキュメントをデータソースとして使用するには、次の手順を行ってください (XML ファイルのデータを使用したサンプルが **XmlDataSourceExample** フォルダにあります)。

1. **PerpetuumSoft.Reporting.Data.XmlDataSource** オブジェクトを作成する
2. **DataFile** プロパティに XML ファイルのパスを設定する
3. **XPath** プロパティに、XML データに適用する最初のパス (XPath 言語の式) を設定する
4. 作成したオブジェクトを **ReportManager** のデータソースとして設定する

これは、**ReportManager.DataSources.Add("DataSourceName", "DataSource")** メソッドを使用して行えます。

他のデータソースと同じように、テンプレートでこのデータソースを使用することができます。

データソースは、最初のパス (XPath プロパティ) を設定することにより、最初の XML ドキュメントのレコードと同じレコード数を格納します。

レポートテンプレートのデータを取得するには、**GetData("SourcePath")** メソッドを使用してください。

この **SourcePath** は、データソース名を指定する文字列で、その次に XML ドキュメントの構造に従ってカンマで区切られた要素のパスとなります。

**SourcePath** が XML のタグ属性のパスを設定する場合、**GetData** メソッドはこの属性の文字列の値を取得します。XML タグ内のテキストを取得するには、**SourcePath** プロパティにタグのパス、その次にカンマで区切られた **"GetText()"** を設定してください。

サンプルについて説明します。

本の情報を格納している XML ドキュメントをデータソースとして使用しています。



```
- <bookstore xmlns:bk="urn:samples">
- <book genre="novel" publicationdate="1997" bk:ISBN="1-861001-57-8">
  <title>Pride And Prejudice</title>
  - <author>
    <first-name>Jane</first-name>
    <last-name>Austen</last-name>
  </author>
  <price>24.95</price>
</book>
- <book genre="novel" publicationdate="1992" bk:ISBN="1-861002-30-1">
  <title>The Handmaid's Tale</title>
  - <author>
    <first-name>Margaret</first-name>
    <last-name>Atwood</last-name>
  </author>
  <price>29.95</price>
</book>
- <book genre="novel" publicationdate="1991" bk:ISBN="1-861001-57-6">
  <title>Emma</title>
  - <author>
    <first-name>Jane</first-name>
    <last-name>Austen</last-name>
  </author>
  <price>19.99</price>
</book>
</bookstore>
```

すべての本の情報を表示するレポートを作成しなければなりません。以下はその手順です。

1. XmlDataSource データソースを作成し、DataFile プロパティにこの XML ファイルを、XPath プロパティに book のパス (XPath='bookstore/book') を設定します。
2. ReportManager.DataSources.Add メソッドを使用して、データソースのコレクション ("XmlSource") に、作成したデータソースを追加します。
3. レポートテンプレートを作成し、DataBand を貼り付けて、DataBand.DataSource = "XmlSource" を設定します。
4. すべての本 ("book") の情報を表示するために、DataBand に Detail を追加します。
5. 本のさまざまなデータを表示するために、Detail に TextBox オブジェクトを追加します。例えば、本のタイトルを表示するには、TextBox.Value (プロパティのバインド) = 'GetData("XmlSource.title.GetText()")' を設定します。作者名を表示するには、'GetData("XmSource.author.first-name.GetText()")' を設定します。XML の属性、例えば 'publicationdate' のデータを表示するには、'GetData("XmSource. publicationdate ")' を設定します。

レポートテンプレートは下図のようになるはずですが。

dataBand1:DataBand DataSource = XmlSource				
header2:Header				
Title	First name	Last name	Date	Price
detail1:Detail				
<GetData ("XmlSource.title.GetText()")	<GetData ("XmlSource.author.first-na	<GetData ("XmlSource.author.last-na	<GetData ("XmlSourc	<GetData ("XmlSourc
end of dataBand1				

実行例 :

Title	First name	Last name	Date	Price
Pride And Prejudice	Jane	Austen	1997	\$24.95
The Handmaid's Tale	Margaret	Atwood	1992	\$29.95
Emma	Jane	Austen	1991	\$19.99

場合によっては、テンプレートのデザイナーで直接 XML ドキュメントをデータソースとして設定しなければならない場合があります。その際は、「データベースに直接アクセスする」を参照してください。

### リレーショナル データベース管理システム

未定

#### データベースに直接アクセスする

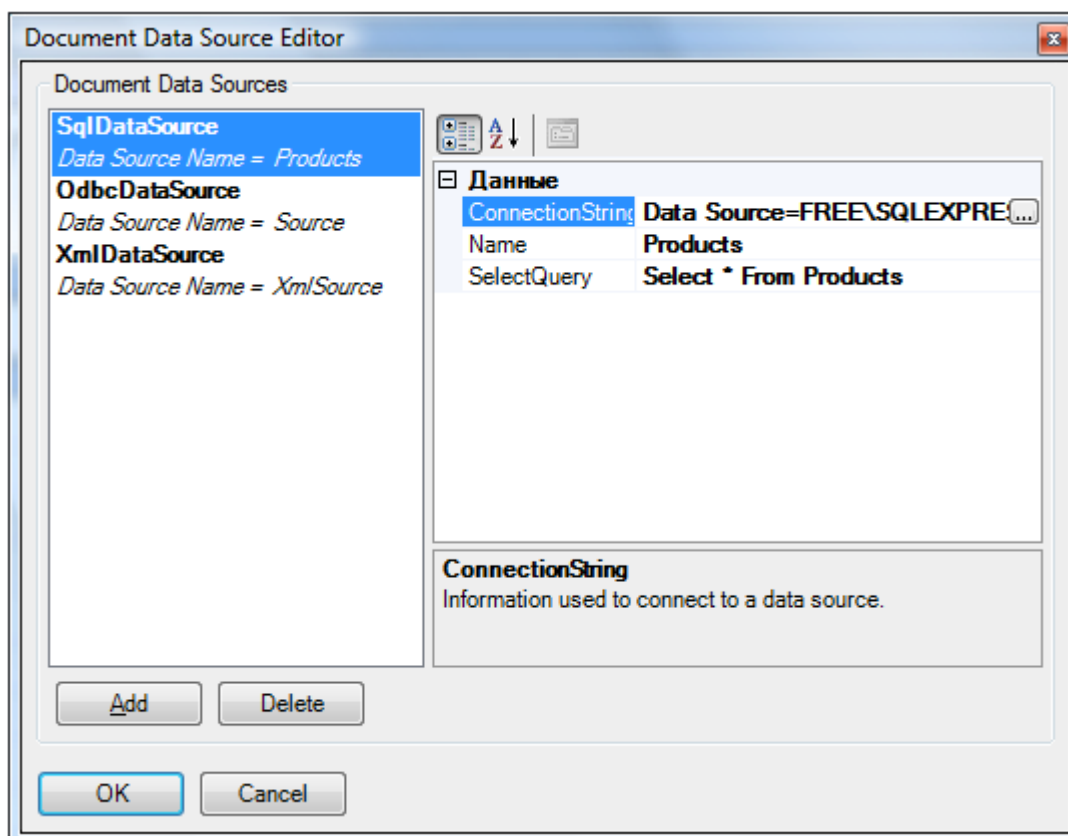
SharpShooter Reports のレポートジェネレータは「アプリケーションでレポートのデータを提供する」というコンセプトを実現します。これは任意のデータソースを使ったり特定のデータソースとは関係ないデータ（レポートテンプレート）を提供するためのルールを作成できます。

しかし、レポートテンプレートを特定のデータソースにバインドしたり、エンドユーザがレポートテンプレートを作成/編集したり独自のデータソースを追加できるようにしなければならない場合があります。

ドキュメントのテンプレートに組み込まれたデータソースのコレクション、つまり直接テンプレートに指定されたデータソース（**Document.DataSources** プロパティ）があるのはそのためです。レポートデザイナーで組み込み式のデータソースを追加したり編集したりすることができます。

組み込み式のドキュメントのデータソースエディタ :





このエディタは **Document.DataSources** プロパティを設定する時やツールバーの「データソース」ボタンをクリックすると表示されます。

このエディタで、次のデータソース型を追加または編集できます。

- 1) **OdbcDataSource** - ODBC インターフェイスを使用してさまざまな DBMS からデータを取得します
- 2) **SqlDataSource** - MS SQL からデータを取得します
- 3) **XmlDataSource** - ドキュメントのデータを取得します

ドキュメントのテンプレートに組み込まれたデータソースとアプリケーションから接続したデータソースの使い方に違いはありません。組み込まれたデータソースとその構造は、(テンプレートのデザイン時にこのデータソースからデータを取得できれば) テンプレートのデザイナーのデータソースツリーから取得できます。

テンプレートのデータを直接データベースから取得するには、**OdbcDataSource** か **SqlDataSource** を使用できます。どちらも機能やプロパティは同じですが、DBSM への接続方法が違います。MS SQL を使用する場合は **SqlDataSource** の使用をお勧めします。

**OdbcDataSource** または **SQL SqlDataSource** を設定するには、次のプロパティを設定しなければなりません。

1. **Name** プロパティ - テンプレートからデータソースを参照するために使用するデータソース名を設定します
2. **ConnectionString** プロパティ - データソースへの接続文字列を設定します
3. **SelectQuery** プロパティ - データを取得するために SQL クエリーを設定します

XML ドキュメントから直接テンプレートにデータを取得するには **XmlDataSource** を追加し、次のプロパティを設定しなければなりません。





1. **Name** プロパティ - テンプレートからデータソースを参照するために使用するデータソース名を設定します
2. **DataFile** プロパティ - XML ドキュメントのパスを設定します
3. **XPath** プロパティ - XML データに適用する最初のパス (XPath 言語での式) を設定します

**XmlDataSource** の使い方に関する詳細は、「XML ファイルをデータソースとして使用する」をご覧ください。

### データなしのレポート

データソースを設定しなくてもレポートは作成できます。**DataBand** オブジェクトには、レポートにオブジェクトの内容を何回出力するかを定義する **InstanceCount** プロパティがあります。データソースなしのレポートサンプルは「**WithoutDataExample**」フォルダにあります。このサンプルは乗算テーブルを出力します。レポートテンプレートには 2 つの **DataBand** があります。**DataBand2** は **dataBand1** に含まれています。どちらの **InstanceCount** プロパティも 10 に設定されています。**textBox1** オブジェクトの **Value** プロパティには次の値が設定されています。

```
dataBand1.LineNumber.ToString() + " * " + dataBand2.LineNumber.ToString() + " = " +  
(dataBand1.LineNumber * dataBand2.LineNumber).ToString()
```

**DataBand.LineNumber** プロパティは現在の行番号を示すので、1 から 10 までの乗算テーブルを表示します。

### 一般的ではないデータソース

未定

### エクスポート

**SharpShooter Reports** は最終ドキュメントを一般的な形式にエクスポートできます。このためにエクスポートフィルタが使用され、**Export** オーバーロードメソッドがあります。

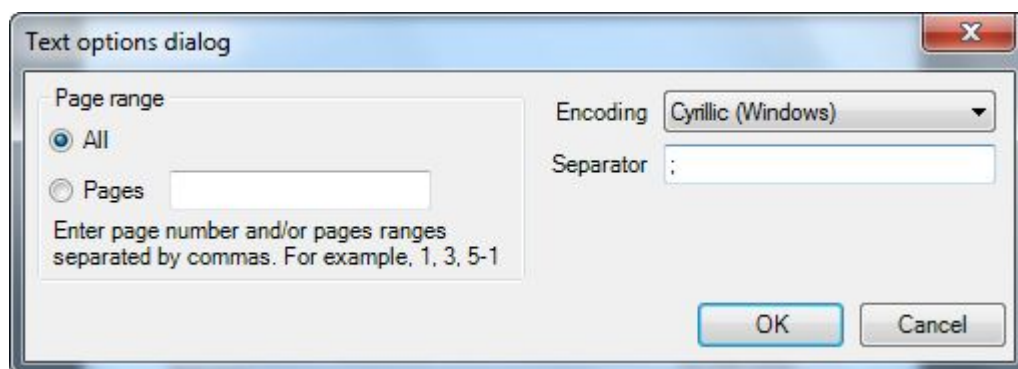
```
public void Export(Document document, string fileName )  
public virtual void Export(Document document, string fileName, bool showDialog),
```

上記の **document** は最終レポート、**filename** はエクスポートするドキュメントの保存先のファイル名を表し、**showDialog** はエクスポート設定ウィンドウの表示/非表示を設定します。最初の **Export** メソッドを使用する場合、エクスポートフィルタ設定のダイアログウィンドウが表示されます。**PerpetuumSoft.Reporting** アセンブリに位置付けられているフィルタ形式へのエクスポートは常にレポートビューアから行えます。他のエクスポートフィルタを使用するには、対象アセンブリへの接続と、少なくともそのフィルタクラスのインスタンスを 1 つ作成する必要があります。最も簡単な方法は、必要とするフィルタをフォームに追加することです。更に、**PerpetuumSoft.Reporting.Export.RegisterExportFilter(ExportFilterFactory factory)** 静的メソッドや **PerpetuumSoft.Reporting.Export.ExportFilters** 静的コレクションにアクセスすることで利用可能なエクスポートフィルタを操作できます。

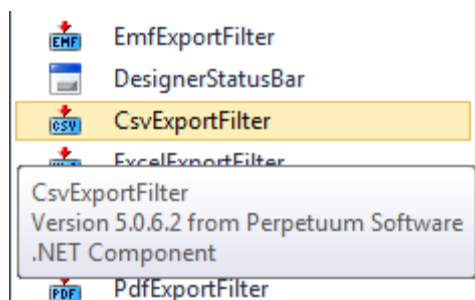
どの形式も標準の設定があります。ページ範囲はエクスポートするレポートページの範囲を設定します。符号化はテキストコードを設定します。解像度はイメージの解像度を設定します。

### CSV エクスポート

CSV エクスポートフィルタは **PerpetuumSoft.Reporting.Export.Text** アセンブリに位置付けられています。テキスト情報だけ CSV にエクスポートされます。**TextBox** や **AdvancedText** オブジェクトのコンテンツは CSV にエクスポートされます。ファイルの行はすべてテーブルの行と一致します。各列の値は（「区切り」フィールドで設定することができる）区切り記号で区切られます。



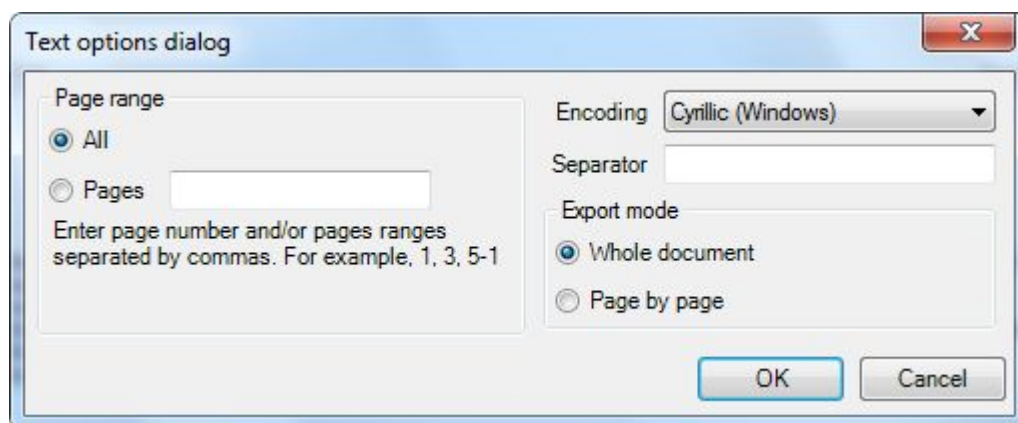
Visual Studio のツールボックスを使用して、プロジェクトに CSV エクスポートフィルタを追加できます。



### TXT エクスポート

TXT へのエクスポートは、PerpetuumSoft.Reporting.Export.Text アセンブリに位置付けられています。TextBox や AdvancedText オブジェクトのコンテンツは TXT にエクスポートされます。エクスポートモードの設定で生成モードを選択できます。「ページ毎」は改ページ記号を追加します。「ドキュメント全体」の場合は改ページ記号の表示はありません。

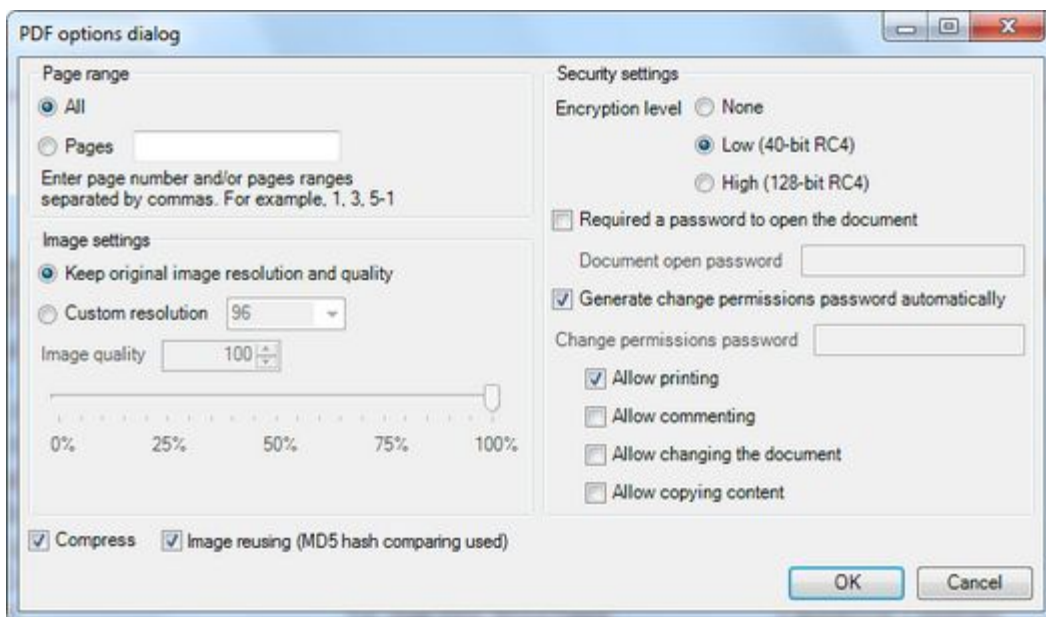
テキストオプションダイアログ



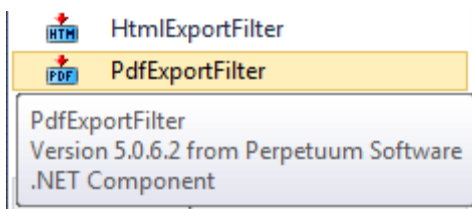
### PDF エクスポート

PDF エクスポートフィルタは、PerpetuumSoft.Reporting.Export.Pdf と PerpetuumSoft.Writers.Pdf アセンブリに位置付けられています。TextBox や AdvancedText オブジェクトのコンテンツは PDF にエクスポートされ、スタイルが保存されます。Picture、ZipCide、BarCode、RichText、WinFormsControl、Chart、Widget オブジェクトはイメージとしてエクスポートされます。ConicalFill や HatchFill は SolidFill としてエクスポートされます。PDF ドキュメントにセキュリティ設定やイメージ品質の設定を行えます。エクスポートしたレポートに反復イメージが含まれている場合、「イメージ再利用」オプションで最終ドキュメントのサイズを最適化することができます。

PDF オプションダイアログ



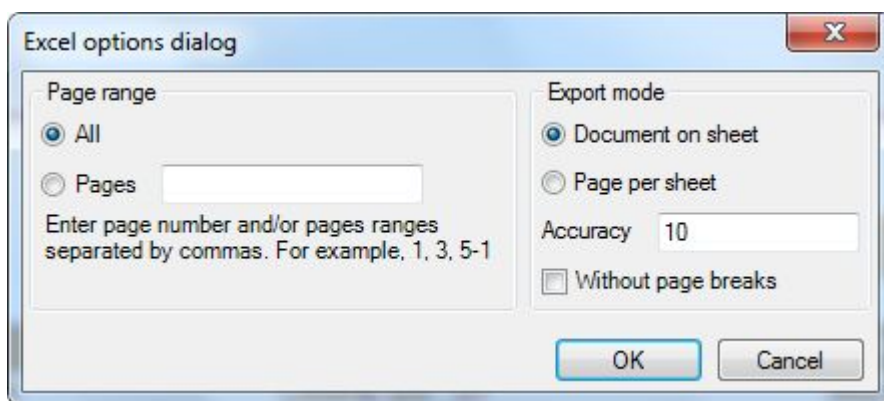
Visual Studio のツールボックスを使用して、プロジェクトに PDF エクスポートフィルタを追加できます。



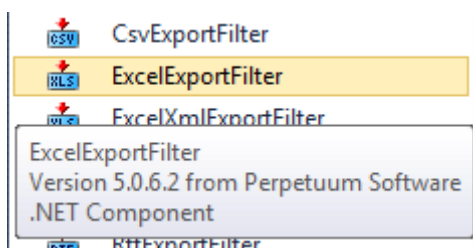
### Excel エクスポート

Excel エクスポートフィルタは、PerpetuumSoft.Reporting.Export.Excel と PerpetuumSoft.Writers.Excel アセンブリに位置付けられています。Shape オブジェクト同様、TextBox や AdvancedText オブジェクトのテキスト情報は Excel にエクスポートされます。Fill と TextFill のすべての塗りつぶし型が SolidFill としてエクスポートされます。エクスポートモードで、全ページを 1 ページに出力するか、1 ページを 1 シートで出力するかを選択できます。

Excel オプションダイアログ



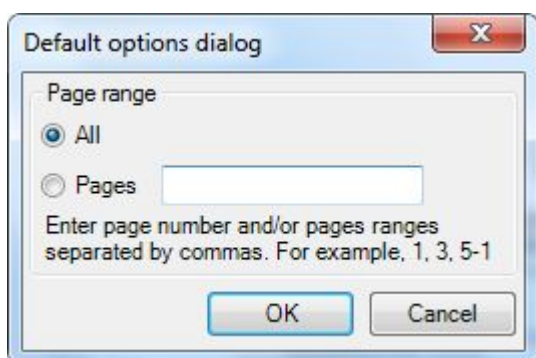
Visual Studio のツールボックスを使用して、プロジェクトに Excel エクスポートフィルタを追加できます。



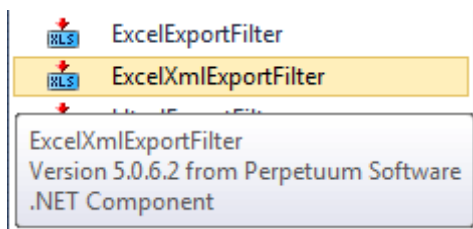
### Excel (XML) エクスポート

Excel XML エクスポートフィルタは、PerpetuumSoft.Reporting.Export.ExcelXML と PerpetuumSoft.Writers.Excel アセンブリに位置付けられています。TextBox や AdvancedText オブジェクトのコンテンツは Excel (XML) にエクスポートされ、スタイルが保存され、Fill と TextFill のすべての塗りつぶし型が SolidFill としてエクスポートされます。

デフォルトオプションダイアログ



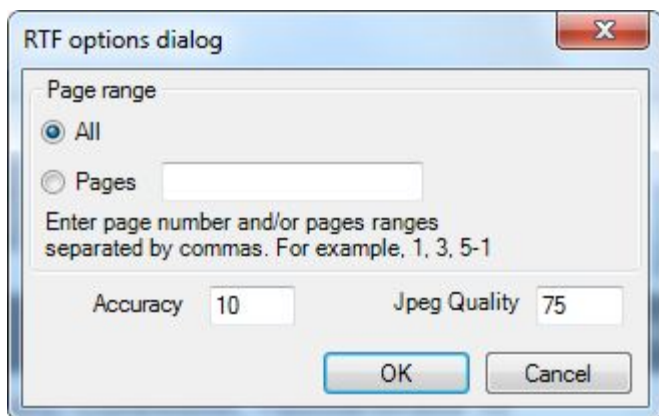
Visual Studio のツールボックスを使用して、プロジェクトに Excel XML エクスポートフィルタを追加できます。



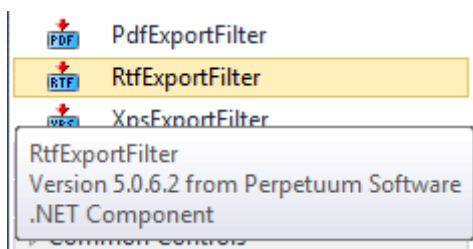
### RTF エクスポート

TextBox や AdvancedText オブジェクトのコンテンツは RTF にエクスポートされ、スタイルが保存されます。Picture、Shape、ZipCide、BarCode、RichText、WinFormsControl、Chart、Widget オブジェクトはイメージとしてエクスポートされます。Fill や TextFill のすべての塗りつぶし型が SolidFill としてエクスポートされます。

RTF オプションダイアログ



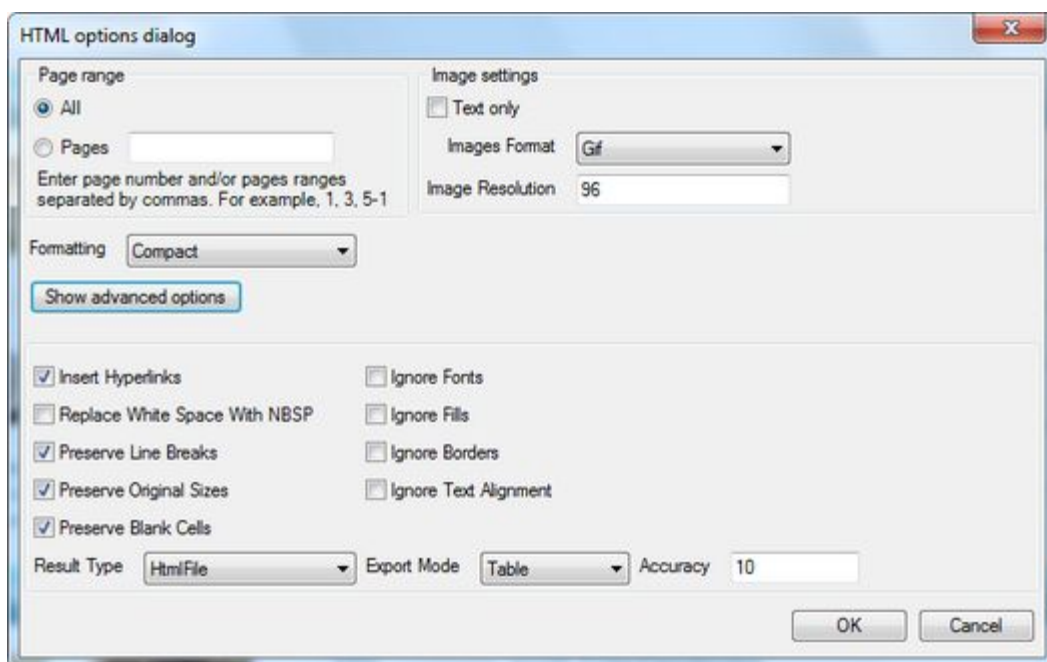
Visual Studio のツールボックスを使用して、プロジェクトに RTF エクスポートフィルタを追加できます。



## HTML エクスポート

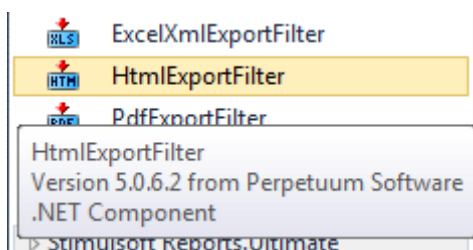
HTML エクスポートフィルタは `PerpetuumSoft.Reporting.Export.Html` アセンブリに位置付けられています。TextBox や AdvancedText オブジェクトのコンテンツは HTML にエクスポートされ、スタイルが保存されます。Picture、ZipCide、BarCode、RichText、WinFormsControl、Chart、Widget オブジェクトはイメージとしてエクスポートされます。Shape オブジェクトはエクスポートされません。テキスト情報だけを保存するには、エクスポート設定で「テキストのみ」にチェックを付けてください。Fill や TextFill のすべての塗りつぶし型は SolidFill としてエクスポートされます。「上級オプション」の設定で、エクスポートの付加的なパラメータを設定できます。

### HTML オプションダイアログ



Visual Studio のツールボックスを使用して、プロジェクトに HTML エクスポートフィルタを追加できます。

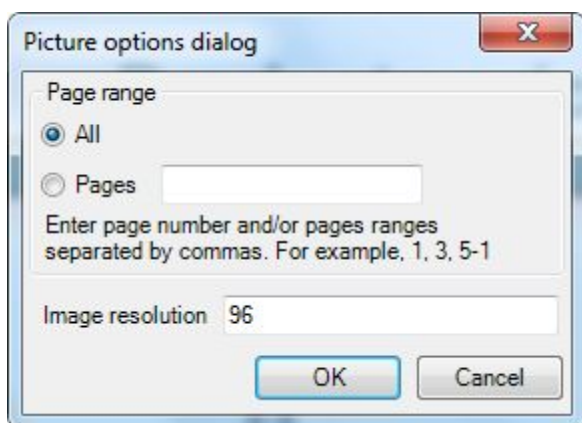




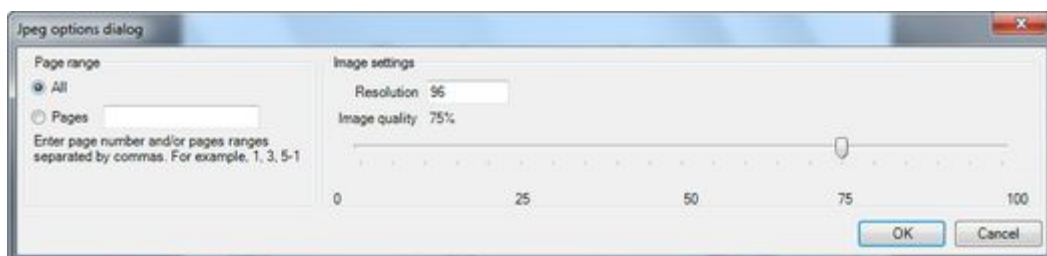
### GIF、JPG、BMP、EMF、TIFF、PNG エクスポート

ビットマップやベクタ (EMF) イメージのエクスポートフィルタは PerpetuumSoft.Reporting アセンブリに位置付けられています。ドキュメントのすべてのレポート要素が含まれます。ページはそれぞれ個別のファイルとしてエクスポートされます (TIFF へのエクスポートは除く)。ビットマップやベクタ (EMF) イメージの解像度や JPG エクスポートのイメージ品質も設定できます。

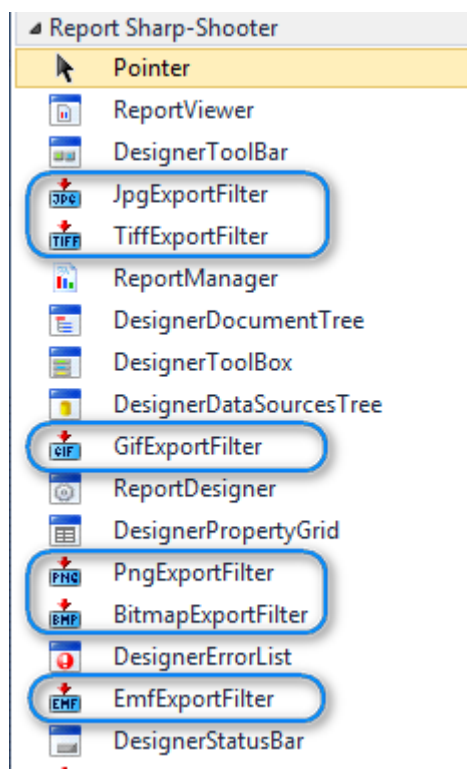
#### Picture オプションダイアログ



#### Jpeg オプションダイアログ



Visual Studio のツールボックスを使用して、プロジェクトエクスポートフィルタを追加できます。



## プロジェクト

### Win フォーム

#### レポートビューアの使用

レポートビューアは、レポートの見た目を設定することができるプロパティのセットを持った、独自のコンポーネントです。以下のプロパティで、ステータスバー、ツールバー、ツールバーの個別/グループのボタンを表示/非表示することができます。

プロパティ	説明
ShowContent	「コンテンツ表示」ボタンやドキュメントの目次を表示/非表示にします
ShowDesigner	「編集」ボタンを表示/非表示にします
ShowExport	「エクスポート」ボタンを表示/非表示にします
ShowFind	「検索」ボタンを表示/非表示にします
ShowNavigator	「戻る」「進む」ボタンを表示/非表示にします
ShowOpen	「(ドキュメントを)開く」ボタンを表示/非表示にします
ShowPageNavigator	「先頭ページに移動」、「前のページに移動」、「次のページに移動」、「最後のページに移動」ボタンを表示/非表示にします。
ShowPrint	「印刷」ボタンを表示/非表示にします
ShowRefresh	「更新」ボタンを表示/非表示にします
ShowSave	「保存」ボタンを表示/非表示にします
ShowScale	「パン」、「ダイナミックズーム」、「ズームイン」、「ズームアウト」、「矩形に合わせてズーム」、「ページ全体に合わせる」、「ページの幅に合わせる」、「実サイズ」ボタンを表示/非表示にします。
ShowStatusBar	ステータスバーを表示/非表示にします
ShowStatusBarGrip	グリッドをステータスバーに表示/非表示にします
ShowToolBar	ツールバーを表示/非表示にします

レポートビューアには **Actions** プロパティがあります。このプロパティはすべての動作へのアクセスを提供し、定義することができます。すべての動作の一覧は次の通りです。



動作	説明
Print	「印刷」ボタンがクリックされると呼び出されます
Load	「(ドキュメントを)開く」ボタンがクリックされると呼び出されます
Save	「保存」ボタンがクリックされると呼び出されます
Export	「エクスポート」ボタンがクリックされると呼び出されます
Pan	「パン」ボタンがクリックされると呼び出されます
DynamicZoom	「ダイナミックズーム」ボタンがクリックされると呼び出されます
ZoomIn	「ズームイン」ボタンがクリックされると呼び出されます
ZoomOut	「ズームアウト」ボタンがクリックされると呼び出されます
ZoomRectangle	「矩形に合わせてズーム」ボタンがクリックされると呼び出されます
Content	「コンテンツ表示」ボタンがクリックされると呼び出されます
WholePage	「ページ全体に合わせる」ボタンがクリックされると呼び出されます
PageWidth	「ページの幅に合わせて」ボタンがクリックされると呼び出されます
ActualSize	「実サイズ」ボタンがクリックされると呼び出されます
Zoom25	ズーム倍率を <b>25%</b> に設定します
Zoom50	ズーム倍率を <b>50%</b> に設定します
Zoom75	ズーム倍率を <b>75%</b> に設定します
Zoom100	ズーム倍率を <b>100%</b> に設定します
Zoom125	ズーム倍率を <b>125%</b> に設定します
Zoom150	ズーム倍率を <b>150%</b> に設定します
Zoom200	ズーム倍率を <b>200%</b> に設定します
Zoom300	ズーム倍率を <b>300%</b> に設定します
Zoom500	ズーム倍率を <b>500%</b> に設定します
Forward	「進む」ボタンがクリックされると呼び出されます
Backwards	「戻る」ボタンがクリックされると呼び出されます
CustomZoom	Ctrl+Z ホットキーまたはステータスバーの「ズーム」の（ズーム倍率を設定する）「カスタムズーム」のコンテキストメニューを選択すると呼び出されます
FirstPage	「先頭ページに移動」ボタンがクリックされると呼び出されます
PrevPage	「前のページに移動」ボタンがクリックされると呼び出されます
NextPage	「次のページに移動」ボタンがクリックされると呼び出されます
LastPage	「最後のページに移動」ボタンがクリックされると呼び出されます
GotoPage	現在のページやページ数の情報を表示するステータスバーをマウスクリックすると呼び出されます
EditReport	「編集」ボタンがクリックされると呼び出されます
RefreshReport	「更新」ボタンがクリックされると呼び出されます
SinglePage	「単一ページ」ボタンがクリックされると呼び出されます
ContinuedPage	「連続ページ」ボタンがクリックされると呼び出されます

規定値に設定された動作を再定義するには、**Executing** イベントハンドラを作成する必要があります。このイベントハンドラにデリゲートされた **ExecutingEventArgs** オブジェクトの **Handled** プロパティが **true** に設定された場合、**Execute** イベントで定義された標準の動作は呼び出されません。

**Export** の動作を再定義する例：

```
reportViewer1.Actions["Export"].Executing += new
Action.ExecutingEventHandler(Export);
and create event handler:
private void Export(object sender, ExecutingEventArgs e)
{
    // Do something

    e.Handled = true;
}
```





## レポートビューアなし

未定

## Web フォーム

### Web の特性

SharpShooter Reports を Web フォームアプリケーションで使用する場合、レポートは常にサーバー側で生成されてからクライアントに送られます。この場合、レポートを **html**、**pdf** などの様々な形式でクライアントに送ることができます。レポートは全部または指定したページだけ送ることができます。では、Web アプリケーションで SharpShooter Reports を使用したさまざまなサンプルについて説明します。

### SharpShooterWebViewer の使用

SharpShooter Reports には ASP.NET アプリケーションのレポートを表示する PerpetuumSoft.Reporting.Web.SharpShooterWebViewer コントロールがあります。このコンポーネントのレポートの表示モードを設定する ViewMode プロパティは PerpetuumSoft.Reporting.Web.ViewMode リストから次の値を取得できます。

- HtmlSinglePage – 単一ページ表示
- HtmlWholeReport – レポート全体を表示
- WindowsForms – PerpetuumSoft.Reporting.View.ReportViewer (Windows フォームアプリケーションでレポートを表示するコンポーネント) にレポートを表示

SharpShooterWebViewer のドキュメントのソースは、ReportSlot コンポーネントの継承元の ReportBase 型の Source プロパティに設定されます。SharpShooterWebViewer はレポート全体を pdf 形式で表示することもできます。SharpShooterWebViewer は表示したドキュメントをすべてキャッシュし、レポートをキャッシュに保存する時間を CacheTimeOut プロパティに設定します。ImageFormat プロパティはレポートに送るイメージ形式を設定します。

WebDemo サンプルは SharpShooterWebViewer の使い方を示しています。

### SharpShooterWebViewer なしで本製品を使用する

「WebPublish」サンプルは SharpShooterWebViewer を使わずにウェブページにレポートを表示する方法を示しています。レポートのタイトルとレポート形式は MainPage で設定します。「表示」ボタンをクリックすると、Click イベントハンドラのクエリが ReportPage のページに転送されます。

```
Response.Redirect("ReportPage.aspx?format="+DropDownList1.SelectedItem.Value.ToString()+ "&title="+TextBox1.Text)
```

ReportPage のページの Load イベントハンドラがレポートを指定した形式にエクスポートし、クライアントに送ります。

### HttpHandler の使用

SharpShooterWebViewer を使わずにレポートを表示する別の方法は、カスタム Http ハンドラを作成しそれに特定の拡張子 (例: \*.rst) を設定する System.Web.IHttpHandler インターフェイスを実装することです。このイベントハンドラは、リクエストで指定されたテンプレートによって生成されたドキュメントを返します。リクエスト処理は ReportHttpHandler クラスの ProcessRequest メソッドで実装されます。リクエストハンドラはレポートを指定した形式にエクスポートし、それを応答するスレッドに書き込みます。このようなハンドラを使用するには、Web.config に次の情報を入れる必要があります。

```
<configuration>
  <system.web>
    <httpHandlers>
      <add verb="*" path="*.rst" type="HttpHandler.ReportHttpHandler,HttpHandler"/>
    </httpHandlers>
  </system.web>
```



```
</configuration>
```

**verb** 属性は、このハンドラを呼び出すための HTTP 命令のリストを定義します。**path** 属性は、このハンドラを呼び出す URL または URL マスクを定義します。**type** 属性は、このハンドラのクラスとそれを位置付けるアセンブリを定義します。

特定の拡張子を使う場合、その拡張子をマッピングする必要があります。それには、Web サイトのプロパティの「ホームディレクトリ」タブの「構成」ボタンをクリックして「アプリケーションのマッピング」タブにある「追加」ボタンを押して実行ファイルを `{disk}:\{Microsoft.NET Framework path}\aspnet_isapi.dll` に、拡張子を `.rst` に設定してください。



## 本製品の高度な特徴を使用する

### カスタム レポート デザイナ

レポートデザイナの標準インターフェイスを変更する場合があります。レポートデザイナの要素（例：ツールバー、ステータス行、エラー一覧など）はすべて個別のコンポーネントとして使用できるので簡単にカスタマイズできます。以下に対象のコンポーネントの一覧を示します。

**DesignerDataSourceTree** データソースを表示するコンポーネント。標準のレポートデザイナでは、ツールウィンドウの「データソース」タブに表示されます。

**DesignerErrorList** スクリプトのエラーを一覧で表示するコンポーネント。標準のレポートデザイナでは、スクリプトのチェックを呼び出してスクリプトにエラーがある場合はフォーム下部に表示されます。

**DesignerPropertyGrid** 選択したオブジェクトのプロパティ一覧を表示するコンポーネント。標準のレポートデザイナでは、ツールウィンドウの「プロパティ」タブに表示されます。

**DesignerStatusBar** レポートデザイナのステータスバー

**DesignerToolBar** レポートデザイナのツールバー

**DesignerToolBox** ツールボックス。標準のレポートデザイナでは、フォームの左側に表示されます。

**ReportDesigner** 編集するレポートテンプレートを表示するメインコンポーネント。レポートデザイナはその他のすべてのコンポーネントにつながります。各コンポーネントの **Designer** プロパティに対象のレポートデザイナを指定するだけです。

カスタムレポートデザイナのサンプルは、「**CustomDesignerViewer**」フォルダにあります。カスタムレポートデザイナを呼び出すために、フォームには「カスタムデザイナ」ボタンがあります。このボタンをクリックすると、**button1\_Click** イベントハンドラが実行されます。1行目はレポートデザイナのフォームを設定します。

```
CustomDesignerForm f = new CustomDesignerForm();
```

このフォームには、レポートのデータソース、サブレポート、編集するレポートテンプレートの設定に使用する、**ReportDesigner** 型の指定された **Designer** プロパティが含まれています。

```
f.Designer.DataSourceManager = reportManager1.DataSources;  
f.Designer.SubReportResolver = reportManager1;  
f.Designer.Document = reportSlot1.LoadTemplate();  
次に、フォームの表示：  
f.ShowDialog();
```

最終レポートをカスタムレポートビューアに表示するために、**ReportDesigner** コンポーネントの **Actions** プロパティを使用して「レポートプレビュー」ボタンのクリックハンドラを再定義します。必要な動作は「**Preview**」です。「**Preview**」を呼び出すと、**Executing** イベントが呼び出されてから **Execute** イベントが呼び出されます。**Executing** イベントハンドラがパラメータによって処理されたことがわかると、**Execute** イベントは呼び出されません。**Execute** イベントを呼び出す場合、このイベントの標準ハンドラが呼び出されます。従って、レポートビューアを呼び出してイベントが処理されたことを通知するフラグを設定する、カスタムな **Executing** イベントハンドラを作成しなければなりません。そのイベントハンドラは次のようになります。

```
private void Preview(object sender, ExecutingEventArgs e)  
{  
    if (reportDesigner1.Document != null)  
    {  
        ReportManager manager = new ReportManager();
```

```
    InlineReportSlot slot = new InlineReportSlot();
    manager.Reports.Add(slot);
    manager.DataSources = this.DataSources;
    slot.SaveReport(this.Document);
    manager.OwnerForm = ParentForm;
    manager.ResolveSubReport += new
ResolveSubReportEventHandler(manager.ResolveSubReport);
    slot.RenderCompleted += new EventHandler(slot_RenderCompleted);
    slot.HyperlinkClick += new HyperlinkEventHandler(slot_HyperlinkClick);
    slot.GetReportParameter += new
GetReportParameterEventHandler(slot_GetReportParameter);
    slot.Prepare();
    e.Handled = true;
}
}
```

```
private void slot_RenderCompleted(object sender, EventArgs e)
{
    ReportSlot slot = sender as ReportSlot;
    using (ViewerForm f = new ViewerForm(slot))
    {
        f.WindowState = FormWindowState.Maximized;
        f.ShowDialog(this);
    }
}
```

このプロシージャを Preview の Executing イベントハンドラとして設定  
reportDesigner1.Actions["Preview"].Executing +=  
new Action.ExecutingEventHandler(Preview);

## カスタム レポート ビューア

例えば、アプリケーションのスタイルに合わない場合やツールバーを変更しなければならない場合など、標準のレポートビューアコンポーネントの標準インターフェイスを使用しなければならない場合があります。

このサンプルは、「CustomDesignerViewer」フォルダにあります。

(ReportViewer.ShowToolBar プロパティを False 設定して) ツールバーのないレポートビューアを作成し、「前のページ」、「次のページ」、「印刷」という 3 つのボタンをフォームに追加します。これを行うには、ReportViewer コンポーネントの Actions プロパティのコレクションを使用します。

コントロールと Action とのリレーションは ActionBind 型のメソッドで設定されます。ナビゲーションボタン (3 つのボタン) と Action をバインドするには、ActionBind の派生クラスを実装する必要があります。

下記に例があります。

```
public class ButtonActionBind : ActionBind
{
    public ButtonActionBind(Button button)
    {
        SetComponent(button);
    }

    public Button Button
    {
        get
        {
            return this.Component as Button;
        }
    }

    protected override void Bind()
    {

```



```
        Button.Click += new EventHandler(Button_Click);
    }

    protected override void Unbind()
    {
        Button.Click -= new EventHandler(Button_Click);
    }

    public override void Update()
    {
        if (Action != null)
        {
            Button b = Button;
            b.Enabled = Action.Enabled;
            b.Text = Action.Text;
            b.Visible = Action.Visible;
        }
    }

    private void Button_Click(object sender, EventArgs e)
    {
        if (Action != null)
            Action.ExecuteAction();
    }
}
```

レポートビューアのボタンと **Actions** をバインドする：

```
PerpetuumSoft.Reporting.Windows.Forms.Action action =
reportViewer1.Actions["PrevPage"];
ButtonActionBind bind = new ButtonActionBind(prevPageButton);
action.Bind(bind);
action = reportViewer1.Actions["NextPage"];
bind = new ButtonActionBind(nextPageButton);
action.Bind(bind);
action = reportViewer1.Actions["Print"];
bind = new ButtonActionBind(printButton);
action.Bind(bind);
```

このようにボタンをレポートビューアのアクションにバインドしました。

レポートビューアに生成されるレポートのソースは、**Source** プロパティに（**IReportSource** インターフェイスを実装する型を）設定します。**ReportViewer** が貼り付けられているフォームのコンストラクタは、**IReportSource** インターフェイスを実装する型のパラメータを受け取って、それを **Source** プロパティに設定します。

## カスタム ウィザード

未定

## カスタム レポート コントロール

カスタム レポート コントロールのサンプルは「**CustomReportControlDemo**」フォルダにあります。

このサンプルで作成されるコンポーネント - **Color** プロパティ（色を設定します）と **Checked** プロパティが追加された **Box** オブジェクトから派生されます。**Checked** プロパティが **true** の場合、オブジェクトの領域に 2 本の対角線（×印）が表示されます。**Stored** 属性は、ドキュメントの書き込み時にこのプロパティが保存されることを設定します。**ReportBindable** 属性は、このプロパティにその値を計算するためのスクリプトを含めることができることを設定します。

一部のクラスのプロパティの規定値を変更したい場合は、**InitNew()** メソッドにコードを記述しなければなりません。**PaintContent** メソッドはコンポーネントを描画します。そして主要メソッドは **Render** メソッドで



す。このメソッドは、最終レポートにコンポーネントを表示するために呼び出されます。このメソッドに、独自のオブジェクトの新しいインスタンスを作成しなければなりません。

```
CheckBoxReportControl result = new CheckBoxReportControl();
```

次にメソッドを呼び出してプロパティを設定

```
PopulateProperties(result);
```

最終レポートでのカスタムコントロールの位置設定

```
RenderLocation(result);
```

最終レポートのページに貼り付け

```
Engine.ProductionPage.Controls.Add(result);
```

コンポーネントを登録し、それをレポートテンプレートに追加するために、**Load** イベントハンドラに次のコードを追加：

```
PerpetuumSoft.Reporting.DOM.ReportControl.RegisterControlType(typeof(CheckBoxReportControl));
```

```
CheckBoxReportControl customControl = new CheckBoxReportControl();
```

```
customControl.Location = new Point(300, 300);
```

```
customControl.Size = new Size(300, 300);
```

```
Document template = reportSlot.LoadReport();
```

```
template.Pages[0].Controls.Add(customControl);
```

```
reportSlot.SaveReport(template);
```

## カスタムエクスポートフィルタ

未定

## カスタムテンプレートの保存

未定

## ライブレポート

**SharpShooter Reports** は、ハイパーリンクのクリックやレポートにある他の任意のオブジェクトを処理する標準メカニズムを再定義することができます。このサンプルは「**LiveReportExample**」フォルダにあります。レポートテンプレートには、**TextBox** オブジェクトと **2** つの **Shape** オブジェクトがあります。

**TextBox.Hyperlink** プロパティが新しいレポートに設定されています。ハイパーリンクのクリックを傍受するには、**ReportManager** コンポーネントの **HyperlinkClick** イベントを使用します。このリンクをクリックすると新しいレポートが表示されるようにします。**HyperlinkClick** イベントハンドラのコードは以下の通りです。

```
private void reportManager1_HyperlinkClick(object sender,
PerpetuumSoft.Reporting.Components.HyperlinkEventArgs e)
{
    if (e.Hyperlink == "New report")
    {
        e.Handled = true;
        try
        {
            reportSlot2.RenderDocument();
            using (PerpetuumSoft.Reporting.View.PreviewForm previewForm = new
PerpetuumSoft.Reporting.View.PreviewForm(reportSlot2))
            {
                previewForm.WindowState = FormWindowState.Maximized;
                previewForm.ShowDialog(this);
            }
        }
        Catch (Exception exception)
        {
            MessageBox.Show(exception.Message, "Report Sharp-Shooter",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

```
}  
}
```

HyperlinkEventArgs クラスには重要なプロパティが 2 つあります。

Hyperlink - ハイパーリンクの文字列

Handled - イベント処理する値を取得します

Shape オブジェクトをクリックするとオブジェクト名が表示されます。それには、ReportViewer コンポーネントの ViewObjectClick イベントを使用してください。このイベントハンドラには次のコードが含まれています。

```
private void reportManager1_ViewObjectClick(object Sender,  
PerpetuumSoft.Reporting.View.ReportViewEventArgs e)  
{  
    if(e.Control is PerpetuumSoft.Reporting.DOM.Shape)  
    {  
        MessageBox.Show("You click on shape: " + (e.Control as  
PerpetuumSoft.Reporting.DOM.Shape).Name);  
        e.Handled = true;  
    }  
}
```

ReportViewEventArgs クラスには 2 つのプロパティが含まれています。

PerpetuumSoft.Reporting.DOM.ReportControl 型の Control プロパティ - クリックされたオブジェクト

bool 型の Handled プロパティ (このプロパティはイベントが処理されたかどうかを取得します)

## レポート生成の操作

SharpShooter Reports は必要に応じて標準のレポート生成メカニズムを変更することができます。それには、対象の Page オブジェクトの ManualBuildScript プロパティにコードを記述する必要があります。手動でレポート生成を行うサンプルが「ManualBuildExample」フォルダにあります。このサンプルには 2 つのレポートが含まれています。1 つは乗算テーブル、もう一つは SharpShooter Reports がインストールされているフォルダがツリー表示されます。

まず、簡単な乗算テーブルのサンプルから説明します。レポートテンプレートには Detail とその中に 3 つの TextBox があります。各 TextBox オブジェクトの Value プロパティにはそれぞれ a、b、c が設定されています。a、b、c は Document.CommonScript プロパティに指定された int 型の変数です。このページの ManualBuildScript プロパティには次のコードが設定されています。

```
for (a = 1; a < 10; a++)  
    for (b = 1; b < 10; b++)  
    {  
        c = a*b;  
        detail1.Render();  
    }
```

つまり、a と b の変数に 1 から 9 までのすべての値をループで渡して値 c を計算し、detail1.Render() メソッドによってレポートに Detail が表示されます。この Detail に含まれている TextBox オブジェクトに表示される変数 a、b、c に新しい値を設定しました。

では、フォルダをツリー表示するサンプルについて説明します。テンプレートには detail1 とその中に textBox1 オブジェクトがあります。このテキストボックスは現在のフォルダまたはファイルを表示するために使用されます。Document.CommonScript プロパティには次のコードが設定されています。

```
private string fileName;  
private float x = 0;  
private void walkTree(DirectoryInfo dirInfo, int c)  
{
```



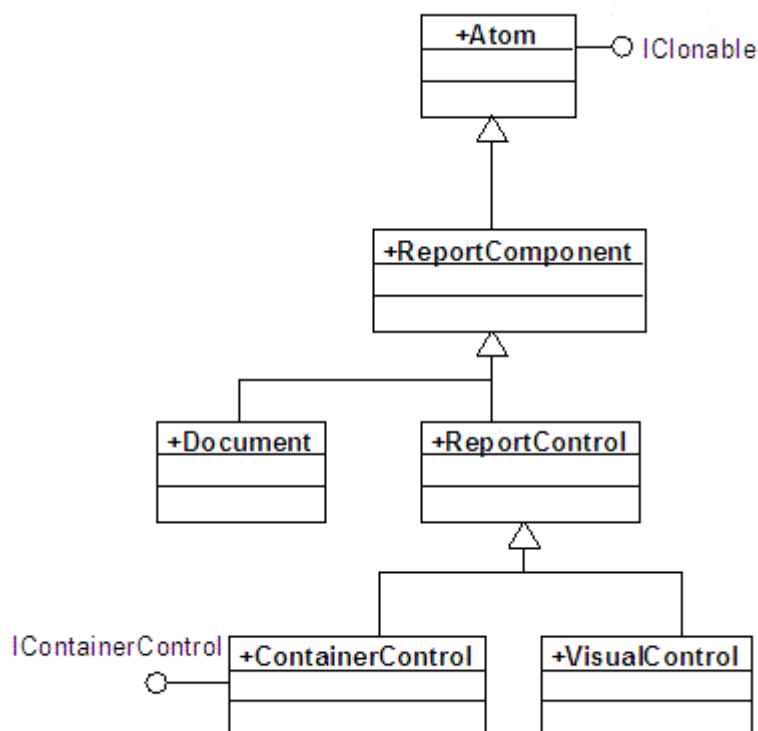
```
x = (float)c * 0.5f;
fileName = "[" + dirInfo.Name + "];
detail1.Render();
DirectoryInfo[] di = dirInfo.GetDirectories();
foreach(DirectoryInfo d in di)walkTree(d,c+1);
FileInfo[] fi = dirInfo.GetFiles();
x = (float)(c+1) * 0.5f;
foreach(FileInfo f in fi)
{
    fileName = f.Name;
    detail1.Render();
}
}
private void buildTree()
{
    string path =
(string)Microsoft.Win32.Registry.LocalMachine.OpenSubKey("SOFTWARE\\PerpetuumSoft\\
SharpShooter Reports").GetValue("Reports");
    path = path.Substring(0,path.LastIndexOf("Examples\\Reports"));
    DirectoryInfo dirInfo = new DirectoryInfo(path);
    walkTree(dirInfo,0);
}
```

スクリプトに 2 つの変数：「fileName」と「x」が設定されています。変数「fileName」は `textBox1` オブジェクトの `Value` プロパティに設定され、変数「x」は `textBox1` オブジェクトの `Location` と `Size` プロパティの新しい値を計算するために使用されます。`buildTree()` メソッドは `SharpShooter Reports` がインストールされているフォルダを算出し、`walkTree()` メソッドはフォルダツリーの再帰回路を作成してフォルダ名やファイル名を表示するために呼び出されます。レポートの生成時にフォルダのツリー表示をするには、`Page.ManualBuildScript` プロパティに `buildTree()` メソッドの呼び出しを追加する必要があります。



## 付録 1 ドキュメント オブジェクト モデル

最終レポートやテンプレートの作成時に使用されるクラスの階層構造について説明します。



**PerpetuumSoft.Framework.Atom** : SharpShooter Reports の全クラスの基本クラスです。このクラスは **System.Object** の派生クラスで、**System.IClonable** インターフェイスを実装します。

この **Atom** クラスは **ReportComponent** の親クラスで、すべての **SharpShooter Reports** コンポーネントの基本クラスです。**ReportComponent** と後述のすべてのクラスやインターフェイスは **PerpetuumSoft.Reporting.DOM** 名前空間にあります。このクラスには、オブジェクト名を設定する **Name** プロパティがあります。

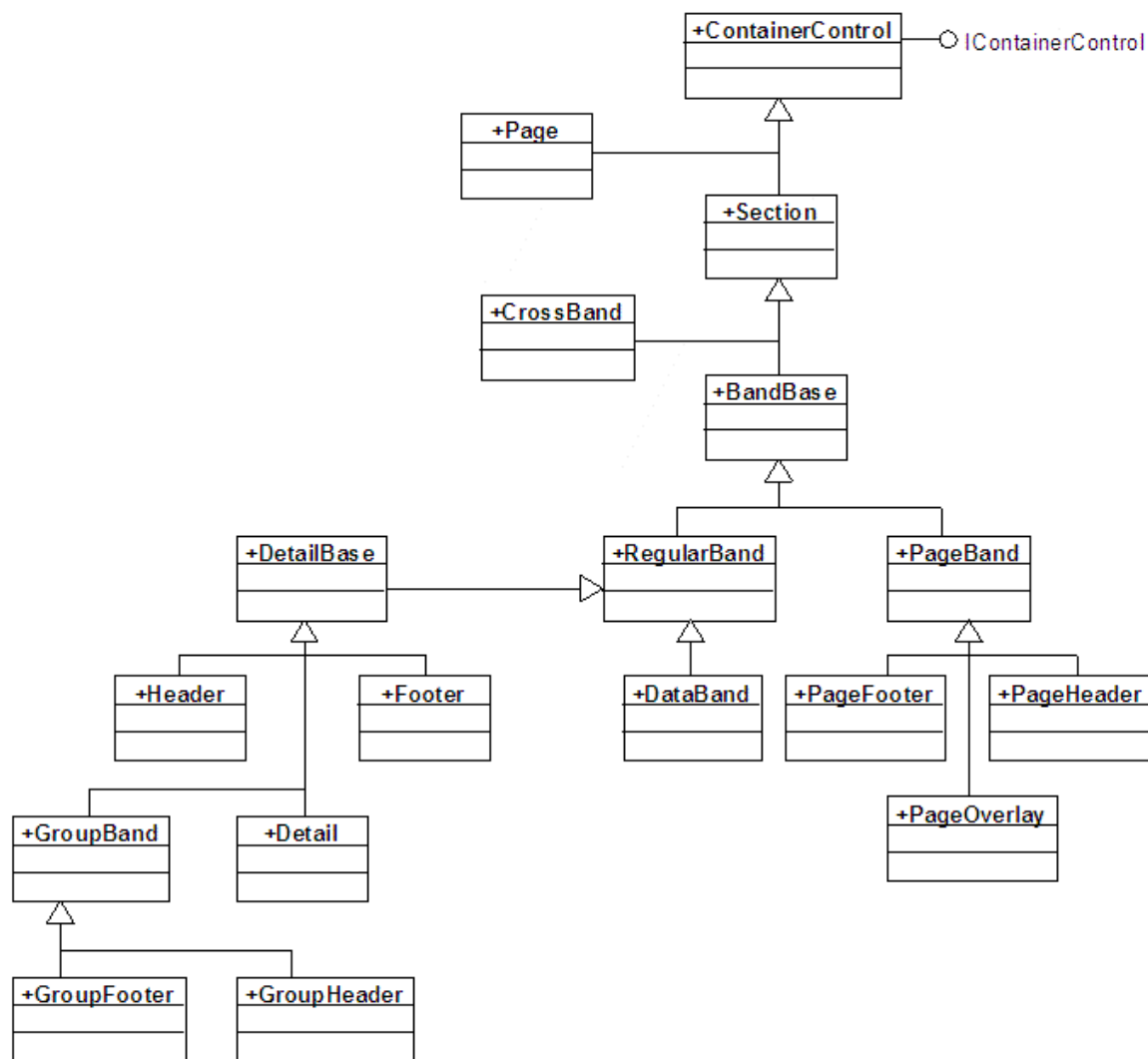
**Document** クラスと **ReportControl** クラスは **ReportComponent** から継承されています。**Document** クラスはレポートテンプレートや最終ドキュメントを表すために使用されます。**Pages** というドキュメントページのコレクションが格納されています。

**ReportControl** : テンプレートの全オブジェクトまたは最終ドキュメントの基本クラスです。

**ContainerControl** クラスと **VisualControl** クラスは **ReportControl** から継承されています。

**ContainerControl** クラスはレポートのすべての非ビジュアルオブジェクトの基本クラスで、**ReportControl** クラスから継承されたオブジェクトを格納することができます。**VisualControl** クラスはレポートのすべてのビジュアルオブジェクトの基本クラスです。

**ContainerControl** クラスについて説明します。このクラスは **Controls** プロパティが格納されている **IContainerControl** インターフェイスを実装します。**Controls** プロパティはこのオブジェクト内で入れ子になったオブジェクトのコレクションです。**IContainerControl** インターフェイスには **IsValidChild** メソッドがあります。このメソッドは、**ReportControl** 型のパラメータを取得し、渡されたオブジェクトをこの中に埋め込むことができる場合は **true** を設定します。以下に **ContainerControl** クラスから継承されたクラスの階層を示します。

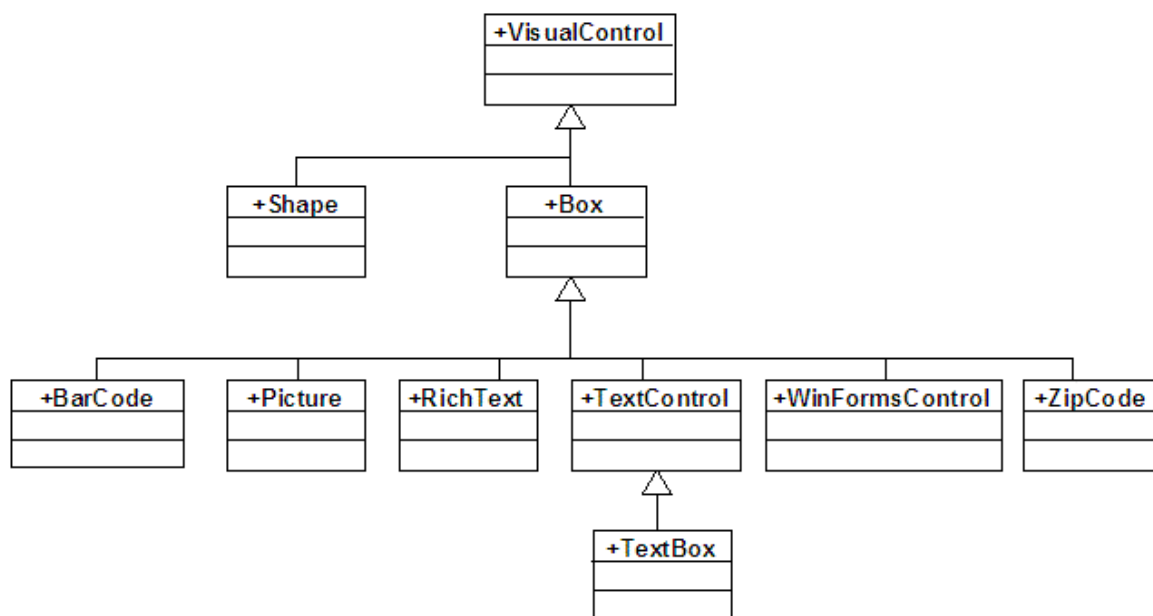


**Page** および **Section.Page** オブジェクトは **ContainerControl** から直接継承されています。 **Page** および **Section.Page** オブジェクトはテンプレートや最終ドキュメントのページを表します。

**Section** オブジェクトはすべてのバンドの基本オブジェクトで、**CrossBand** と **BandBase** は **Section** オブジェクトから継承されています。 **CrossBand** はレポートを左から右に表示することができます。 **PageBand** と **RegularBand** は **BandBase** から継承されています。 **PageBand** は各ページに表示する **PageHeader**、**PageFooter**、**PageOverlay** の基本クラスです。 **RegularBand** クラスはその他の全セクションの基本クラスです。この中に集計コレクション（**Aggregates** プロパティ）が格納されており、**DataBand** 型の **MasterBand** プロパティがあります。従って、**RegularBand** クラスから継承されているこれらのオブジェクトを **DataBand** セクションに配置することができます（**DataBand** 自体も含む）。これにより任意の入れ子レベルのマスター - サブレポートを作成することができます。

**DetailBase** クラスは **RegularBand** クラスから継承されています。このクラスはすべてのセクションの基本クラスで、ビジュアルコントロール要素を入れることができ、レポートのバンド内容を表示する **Render** メソッドがあります。 **Header** と **Footer** は **DetailBase** から直接継承されています。 **Detail** は **DetailBase** から継承されており、データソースや **GroupBand** の各レコードを出力します。 **GroupBandBase** は **GroupHeader** と **GroupFooter** の基本クラスで、式を設定する **Group** プロパティがあります。式を変更した場合、**GroupHeader** と **GroupFooter** に格納されているビジュアルコントロールが最終ドキュメントに出力されます。

では、VisualControl クラスについて説明します。



Shape クラスは VisualControl クラスから継承されています。Shape クラスはレポートにさまざまな形状を表示することができます。Box クラスも VisualControl クラスから継承されています。Box クラスは矩形領域に配置されるコントロール要素の基本クラスで、領域の枠を指定する Border プロパティとその領域の塗りつぶしを指定する Fill プロパティ、マージンを定義する Margins プロパティがあります。

Barcode、Picture、RichText、WinFormsControl、ZipCode は Box クラスから継承されています。Barcode はバーコードを出力するためのコントロール要素です。Picture はイメージを出力するためのコントロールです。RichText はテキストを RTF 形式で出力するコントロール要素です。WinFormsControl は Windows Forms コントロール要素を出力するためのコントロール要素です。ZipCode は Zip コードを出力するためのコントロール要素です。TextControl クラスは Box クラスから継承されています。TextControl はレポートにテキストを出力するための基本クラスで、テキストの回転角度を設定できる Angle プロパティ、出力するテキストに応じて自動的にオブジェクトの高さを変更できる CanGrow および CanShrink プロパティ、フォントを設定する Font プロパティ、テキストの位置合わせをする TextAlign プロパティ、テキストの塗りつぶしを設定する TextFill プロパティ、テキスト形式を設定する TextFormat プロパティがあります。TextBox は TextControl から継承されています。TextBox クラスにはテキストのスタティック値を設定する Text プロパティ、生成時にテキストを計算するための式を設定する Value プロパティがあります。TextBox クラスにはテキスト情報を検索するための FindText メソッドがあります。