|  |
| --- |
|  |

|  |
| --- |
| **Microsoft レポートによる帳票の作成** |
|  |
|  |
| Microsoftレポートを用いた.NET Frameworkによる帳票の作成について記述します。 |
|  |
|  |
| **2013/12/14** |
|  |

# Microsoftレポートとは？

MicrosoftレポートとはVisual Studioが帳票の作成をサポートするために提供しているコントロールです。レポートのデザイン機能と、アプリケーションでレポート処理および表示をするためのReportViewerコントロールが提供されています。

次の図はMicrosoftレポートが、どのように帳票を作成するかを簡単に表したものです。

**帳票**

**レポート定義**

**データソース**

**Report Viewer**

**コントロール**

**データソース：**

アプリケーションが使用するデータのことで特にアプリケーションで操作する必要のあることが明確なデータです。

**レポート定義：**

帳票のレイアウトや、どのデータを作成するかをXMLで記述したものですVisual Studioのレポートデザイナを使用して変更が可能です。

**ReportViewerコントロール：**

データセットとレポート定義より帳票を作成します。

## 作成できる帳票

Microsoftレポートは形式の帳票を作成できます。以下に、いくつかの例を紹介します。

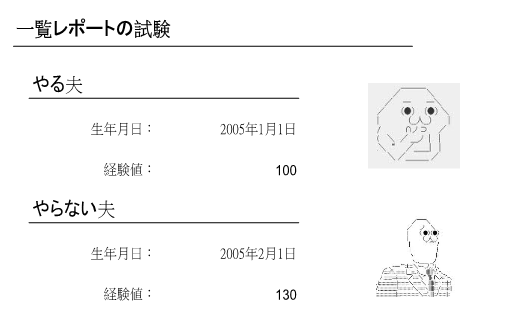
### 一覧形式



Excelのように一覧としてデータを表示します。

各フィールドでは書式の指定や数式の指定が行えます。必要であれば、開発者は自分の目的にあった数式を自作して利用することもできます。

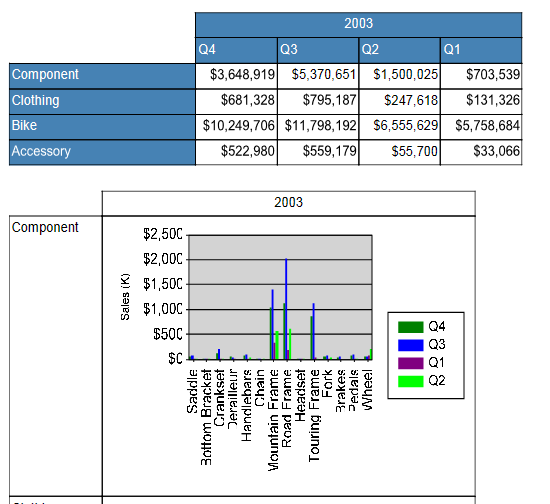
### 単票形式



単票としてデータの詳細を自由なフォーマットで表現します。

数字や、文字だけでなく、画像も使用できます。

### グラフ



データをグラフとして表示することができます。

棒グラフだけでなく、折れ線グラフや円グラフなど、様々な表現方法が実現できます。

## 動作環境

Visual Studio 2005以降でProfessional以上であれば使用できます。

ExpressにはReportViewerコントロールは提供されていません。

なお、マイクロソフトが提供するチュートリアルを行う場合、SQL Serverが別途必要になります。

このドキュメントで紹介するチュートリアルは以下の開発ツールがインストール済みであることを前提とします。

Visual Studio2013 Professional

SQL Server2012 Express

SQL Server Management Studio(SSMSと略される場合がある)

# チュートリアル

Microsoftは下記のページにサンプルとチュートリアルを用意しています。

サンプルとチュートリアル

<http://msdn.microsoft.com/ja-jp/library/ms251686%28v=vs.90%29.aspx>

VisualStudio2013とSQLSERVER2012では、このチュートリアル通りおこなっても、サンプルデータすら作成できません。

この章ではチュートリアルの補足を行います。

## サンプルで使用するデータベースの構築

チュートリアルで使用するサンプルを動かすにはSQLSERVERでテスト用のデータベースを作成する必要があります。

チュートリアル : AdventureWorks データベースのインストール

<http://msdn.microsoft.com/ja-jp/library/aa992075%28v=vs.90%29.aspx>

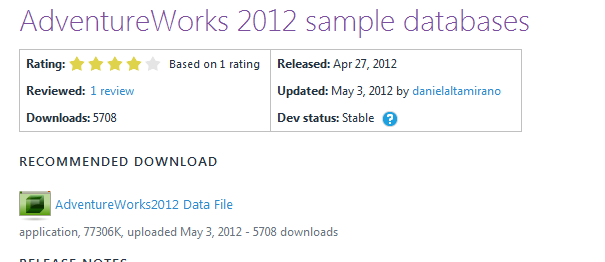
ここではテストデータのはいったSQLSERVERのデータベースを開発者のPCに構築するための手順が記述してあります。

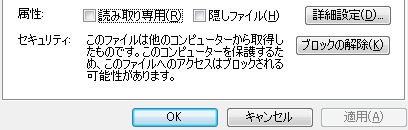
ただし、ここで記述している通りにやっても、ファイルのダウンロードすらできません。以下の手順を参考にしてSQLSERVERを構築してください。

1. まずテスト用のデータベースを下記のサイトから取得します。

AdventureWorks Sample Databases for the SQL Server 2012 Database Training Kit

<http://sql2012kitdb.codeplex.com/releases/view/86805>



1. ダウンロードしたMDFファイルをSQLSERVERにアタッチできる場所にコピーします。  
   SQLSERVERをデフォルトの設定でインストールした場合は以下のパスになるでしょう。  
   C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA  
     
   もし使用しているSQLSERVERがEXPRESSの場合は次のパスになります。  
   C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS\MSSQL\DATA  
     
   コピーを行うさい、ダウンロードしたMDFファイルの「ブロックの解除」が行われていることを確認してください。これはファイルを右クリックしてプロパティで確認できます。  
   
2. SQLServer　Management Studioを起動してローカルのSQLSERVERに接続します。  
   「localhost」のWindows認証で接続できるはずです。  
   もし、Expressの場合は「Localhost\SQLEXPRESS」を指定してください。
3. オブジェクトエクスプローラーで「データベース」ノードを右クリックして「アタッチ」を選択してください。
4. 「追加」をクリックしてコピーしたMDFファイルを選択してOKボタンを押します。
5. 「データベースの詳細」リストからファイルの種類で「ログ」を選択して削除します。
6. OKボタンを押せばデータベースはアタッチできます。
7. これをすべてのMDFファイルに対して行ってください。

## チュートリアル：ReportViewerレポートの作成

ここでは下記のチュートリアルを行うための補足説明を行います。

チュートリアル : ReportViewer レポートの作成

<http://msdn.microsoft.com/ja-jp/library/ms252073%28v=vs.90%29.aspx>

このチュートリアルではテーブルを用いて一覧形式の帳票を作成します。

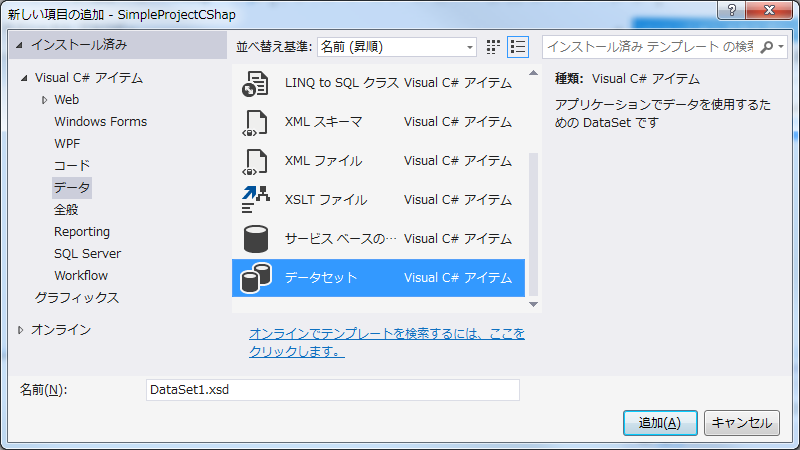
SQLSERVER上のテーブルが最新のものと違うため、ドキュメント通りやっても動作しないので注意してください。

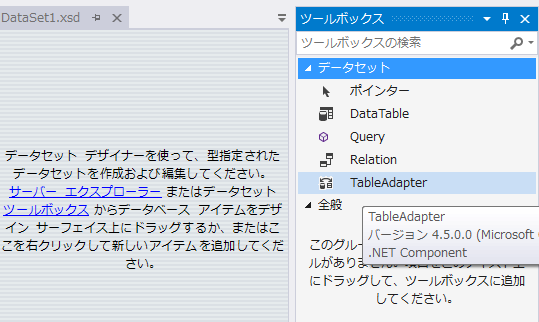
また、チュートリアルでの言語はVB.NETですが、C#でも同じ手順で動作します。

### 「Windowsフォームアプリケーション」を作成します

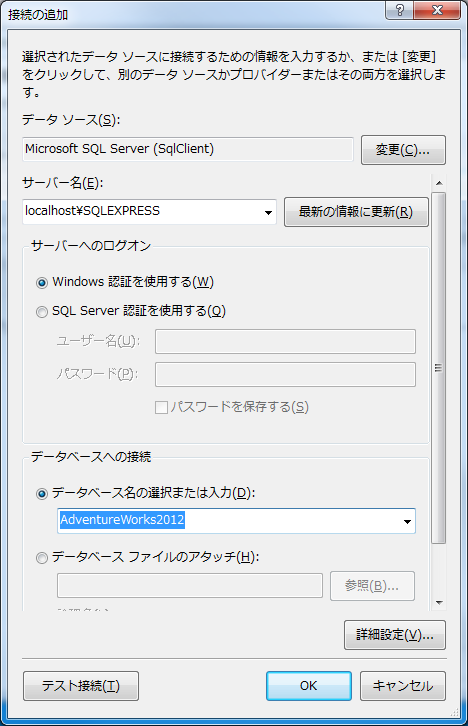
Windowsフォームアプリケーション用のプロジェクトを作成してください。

### 「データソース接続」と「データテーブル」を定義します

1.ソリューションエクスプローラーから「新しい項目」の追加で「データセット」を追加します。  


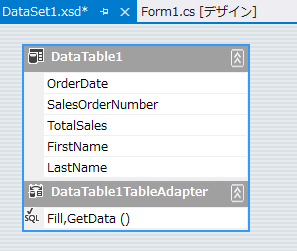
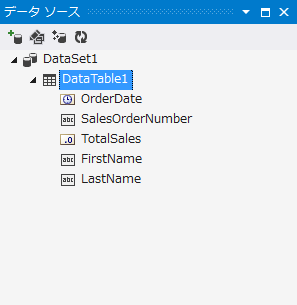
2.データセットデザイナが表示されるので、ツールボックスの「TableAdapter」コントロールをドロップします。  
  
これにより、TableAdapter構成ウィザードが開きます。

3.[データ接続の選択] ページで [新しい接続] をクリックします。

4.以下のような設定を行ってください  
  
※VisualStudio2008とSQLSERVER2012の場合はエラーが発生します。この時は付録の「[VS2008からSQLSERVER2012に接続する](#_VS2008からSQLSERVER2012に接続する)」を参照してください。

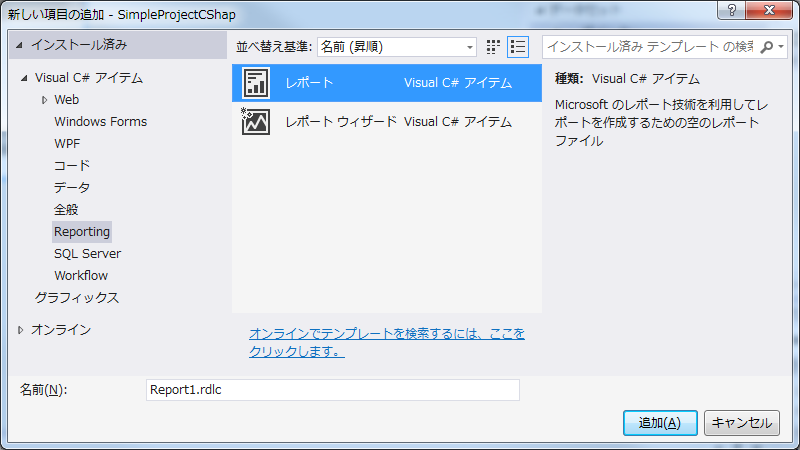
5.接続を追加後、TableAdapter構成ウィザードの作成を進めていくとSQLステートメントの入力を求められます。  
この際、チュートリアル通りにやるとエラーになります。次のように修正してください。

|  |
| --- |
| SELECT S.OrderDate,   S.SalesOrderNumber,  S.TotalDue AS TotalSales,   C.FirstName,   C.LastName FROM HumanResources.Employee AS E INNER JOIN  **Person.Person** AS C  ON E.**BusinessEntityID** = C.**BusinessEntityID** INNER JOIN Sales.SalesOrderHeader AS S ON E.**BusinessEntityID** = S.SalesPersonID |

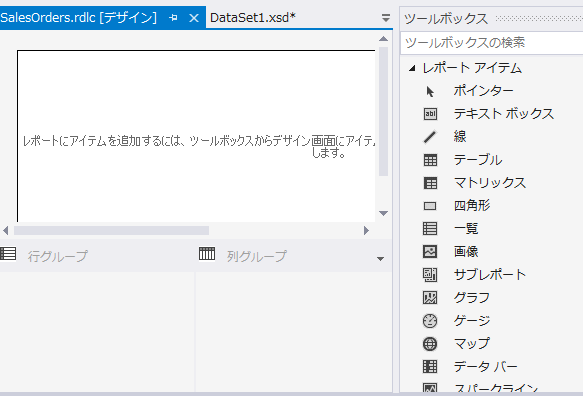
6.これにより、データセットに次のようなデータが作成されました。  
  
これはデータソースウィンドウから使用できます。  


### 新しいレポート定義ファイルを追加

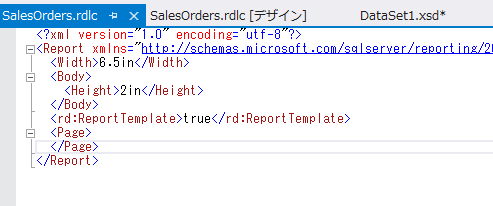
1.「プロジェクト」メニューの「新しい項目」を選択します。

2.Repotingに存在する「レポート」を選択してください。  


3.[ファイル名] に「SalesOrders.rdlc」と入力し、[追加] をクリックすると、デザイン画面が開かれます。

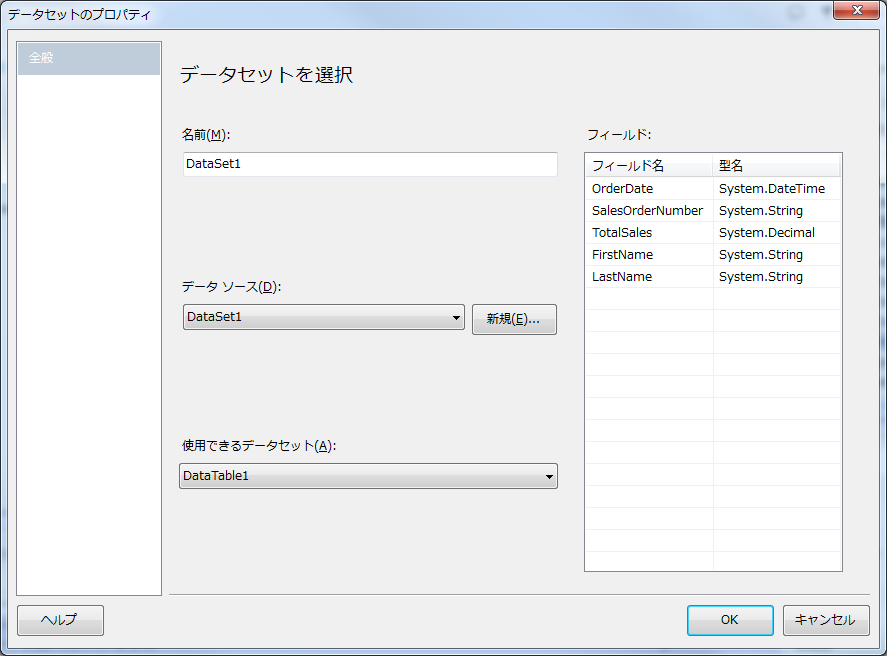
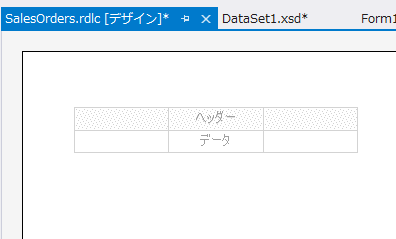
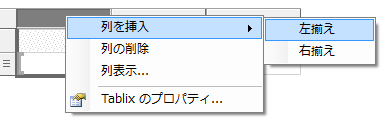
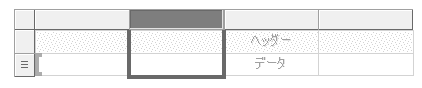
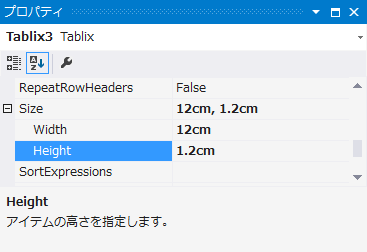


このようにGUIでレポートを編集するアプリケーションを「レポートデザイナ」といいます。レポート定義ファイルは「レポートデザイナ」だけでなく、下記のように「XML（テキストエディタ）」でも開くことができます。

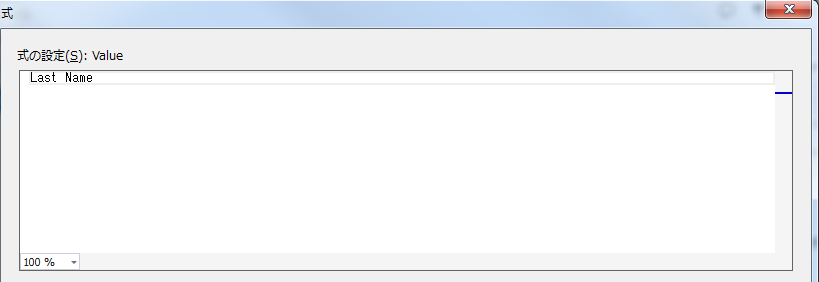
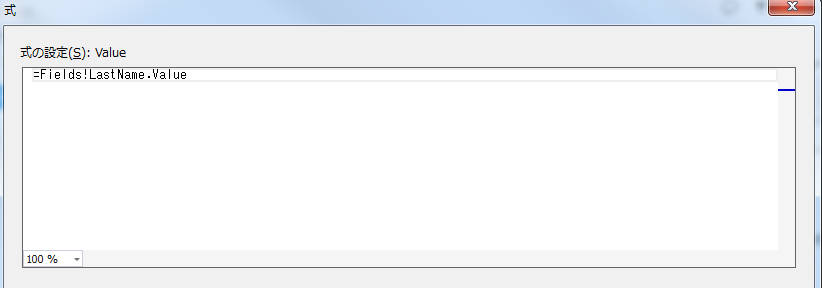
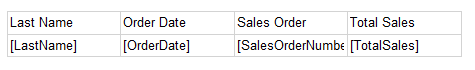


### レポートレイアウトでテーブルを追加する

テーブルを利用して一覧形式の帳票を作成します。

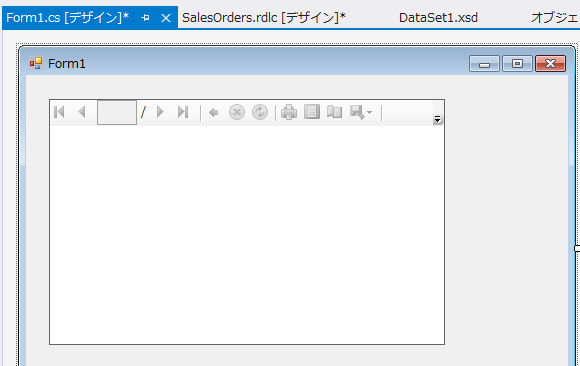
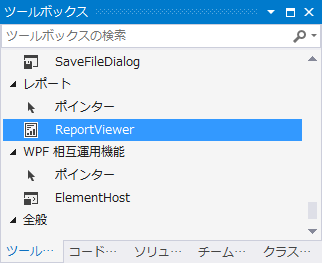
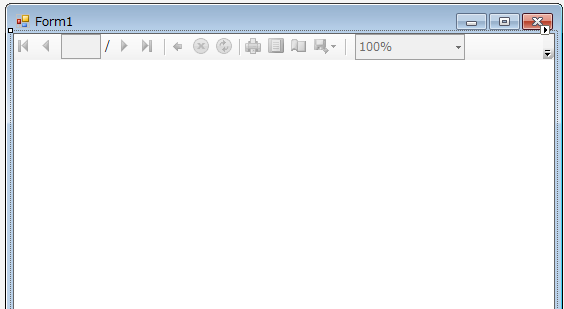
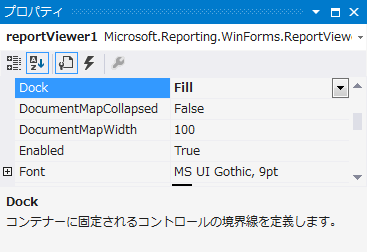
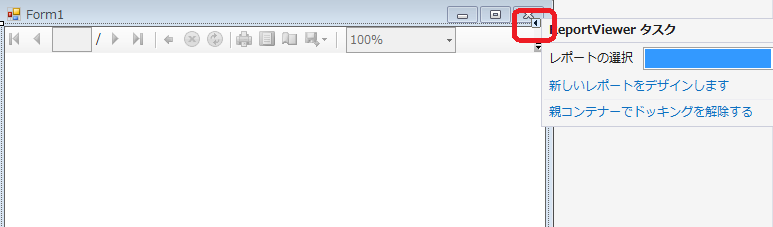
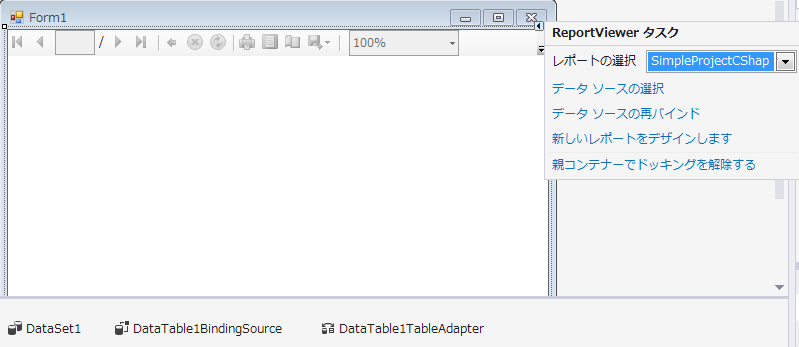
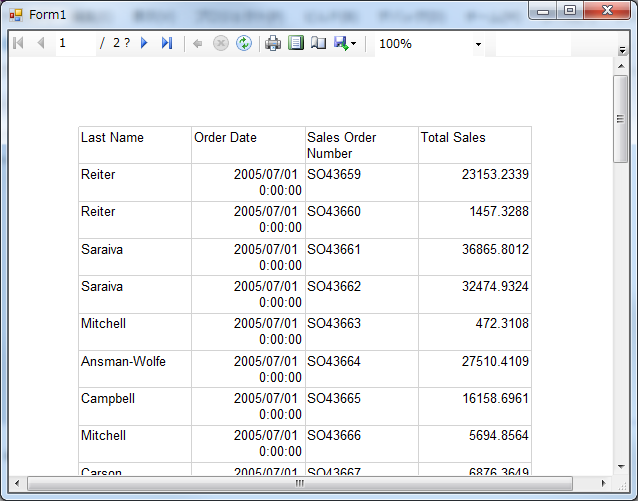
1. レポートにテーブルを追加するために、ツールボックスの「テーブル」を選択後、「レポートデザイナ」をクリックします。  
     
   これによって、「データセット」プロパティが起動するので、データソースにDataSet1を選択してOKボタンを押下してください。  
     
   以後、作成したテーブルとデータセットが関連づけられ、指定したデータセットのフィールドをテーブルのセルで使用することが可能になります。  
     
   また、データセットプロパティを閉じると「レポートデザイナ」には３列のテーブルが作成されます。  
   
2. 作成されたテーブルをクリックすると列ハンドルと行ハンドルが表示されます。  
   
3. 左端の列ハンドルを右クリックして左側に列を挿入します。  
     
   
4. 列の幅を調整します。「プロパティ」ウィンドウで「tablib*n*」というコントロールを選択して、「Size」ノードの「Width」に任意の数値を入力します。  
   
5. テーブルに表示するデータを指定します。
   1. 最初の列の中央行にマウスオーバをすると「データソース」のアイコンが出現するのでクリックをします。



* 1. 関連付けるフィールドの一覧がメニューとして表示されるので「LastName」を選択してください。  
       
       
     その結果、テーブルには以下のような値が書き込まれます。  
       
     １行目と２行目にどのような違いがあるのでしょうか？  
     実は１行目はただの「文字」ですが、２行目は「式」になります。  
     各セルを右クリックして「式…」を押下するとそのことが確認できます。  
       
     １列目－１行目  
       
       
     １列目―２行目  
       
       
     1行目は「Last Name」というただの文字で「＝」がなく式ではありません。  
     ２行目は「=Fields!LastName.Value」という式になっています。  
     この式は指定したデータセットのフィールドLastNameの値をこのセルに表示することを意味します。
  2. 同様にOrderData,SalesOrderNumber、TotalSalesを２～４列目に指定します。  
       
     これによりテーブルのすべてのセルに何らかの値が入力されました。

### フォームにReportViewerコントロールを追加する

フォームにReportViewerコントロールを追加して、作成した帳票を表示するようにします。

1. フォームをデザイナで開いて、ツールボックスから「ReportViewerコントロール」を張り付けてください  
   
2. ReportViewer　コントロールの「Dock」プロパティを「Fill」にすることでコントロールがFormの大きさに合わせて自動リサイズされるようになります。  
   
3. ReportViewerコントロールの右上隅の三角形をクリックします。  
   これにより「ReportViewerタスク」メニューが表示されます。
4. レポートの選択でさっき作成したレポート定義ファイルを選択します。  
   これにより、Formには次のコントロールが追加されます。これらのコントロールはレポート定義にデータベース上のデータを表示するために使用されます。
5. この時点でビルドして実行すると次のようなウィンドウが起動されます。  
   

主なアイコンの説明は次の通りです。

  
ページを遷移するのに使用します。  
  


印刷を実行するためのダイアログを表示します。



印刷プレビューにします。解除するにはもう一度このボタンを押してください。  

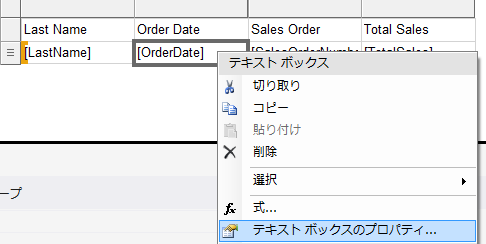
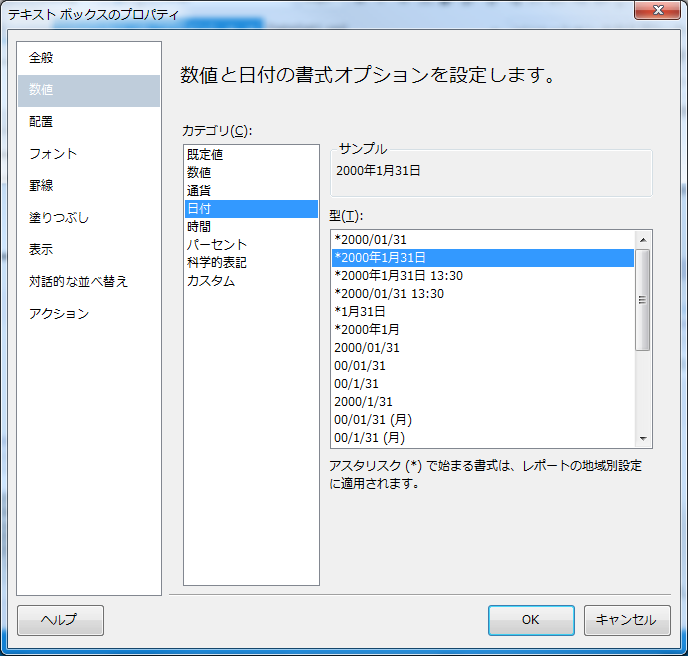
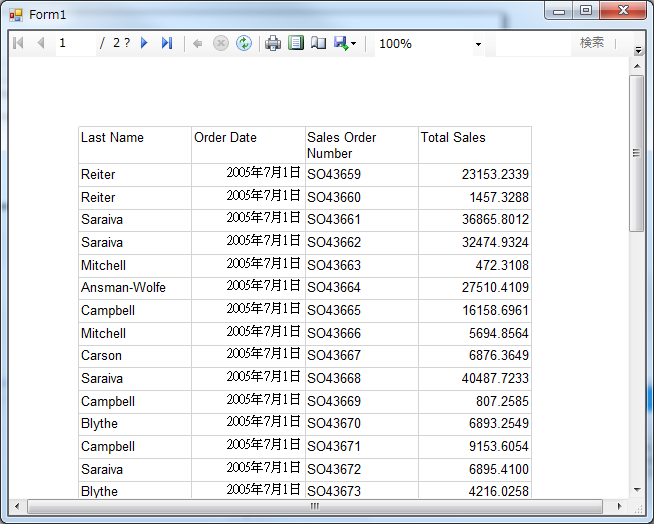
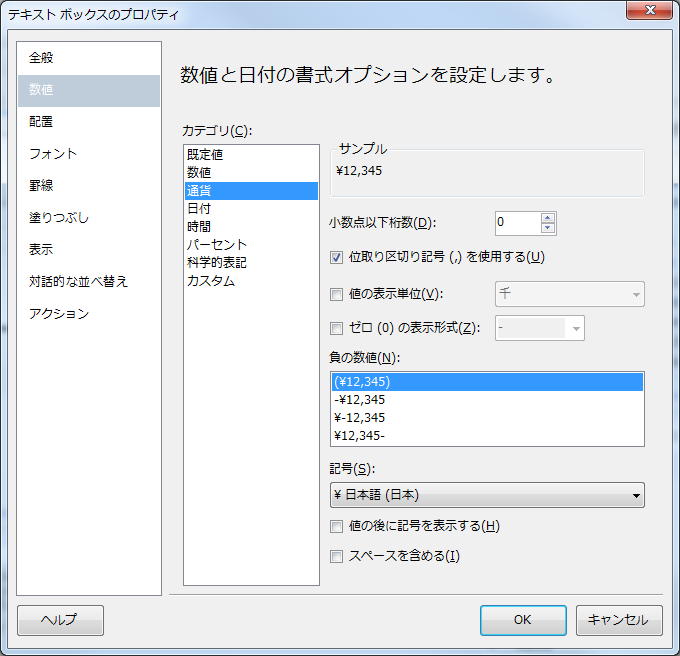
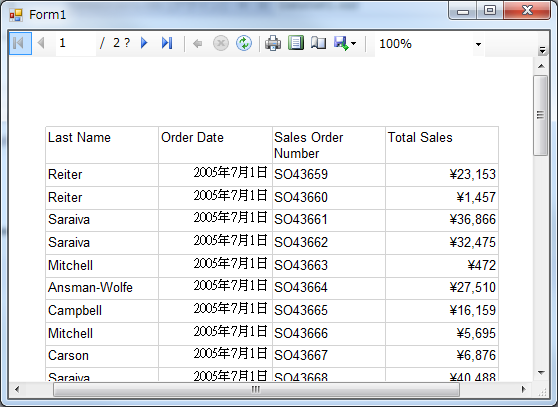

ページの設定を行うためのダイアログを表示します。  
  
　別の形式でファイルを保存します。Excel,PDF,Wordのいずれかを選択できます。



　任意の単語で検索を行えます。「次へ」を使用することで別ページの単語も検索します。

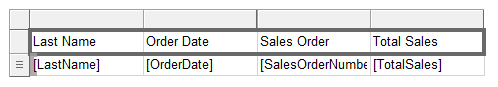
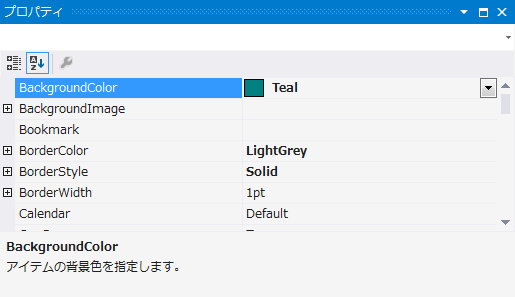
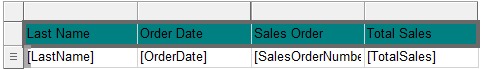
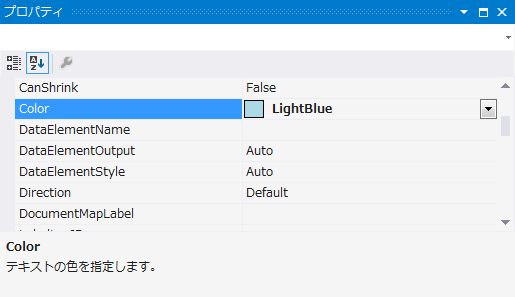
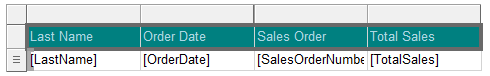
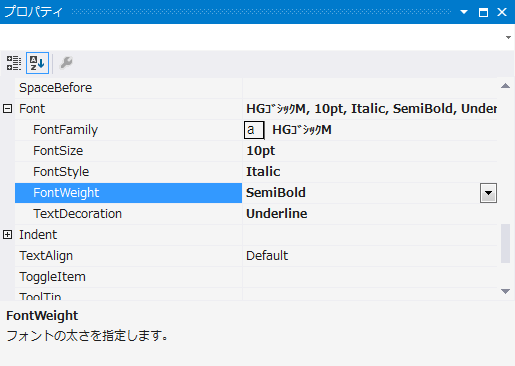
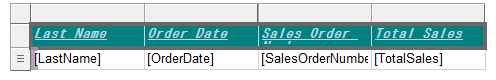
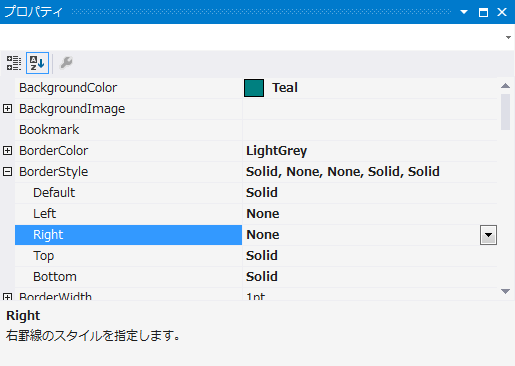
### フィールドの書式を指定する

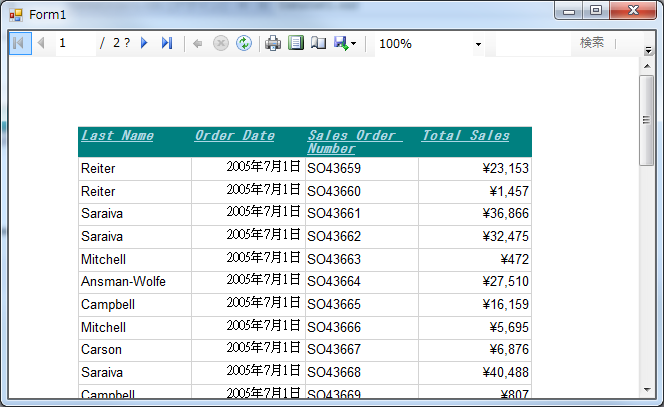
レポートデザイナでは、フィールドの書式を設定することができます。  
まず、OrderDateの日付を日本語の書式に変更してみましょう。

1. 書式を指定したいセルを右クリックして「テキストボックスのプロパティ」を選択します。  
   
2. テキストボックスのプロパティダイアログでは「数値」項目を選択すると様々な書式を設定できます。  
   カテゴリで「日付」を選択して書式を選択して、OKでダイアログを終了してください。  
   
3. ビルドして実行すると下記のようにOrderDateが選択した書式のものになっていることが確認できます。  
   
4. 同様にTotalSalesを選択して書式を「通貨」にします。  
     
   小数点以下桁数や区切り記号の有無などを指定できます。
5. これを再度ビルドして実行すると指定した書式でTotalSalesが表示されます。  
   

### セルの修飾

各セルは、それぞれフォント、背景色、罫線などで修飾してデータを見やすくすることが可能です。

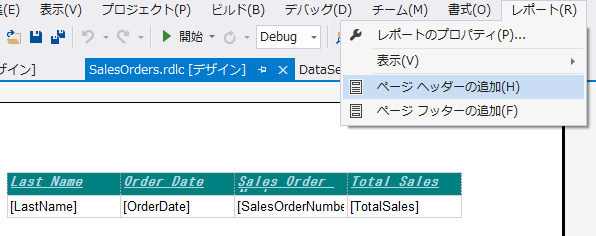
1. 修飾を行いたいセルを選択します。  
   
2. この状態でプロパティのBackgroundColerで任意の背景色を選択します。  
     
     
   これにより選択した範囲の背景色が変更されました。
3. プロパティのColorに任意の値を指定すると文字の色が変更されます。  
     
   
4. フォントについてはプロパティのFontを変更することで、フォントの種類やサイズなどが変更できます。  
     
   
5. 罫線はBorderStyleプロパティで変更できます。この例だと上下は実線、左右には線を引かない設定です。  
   
6. ここまで変更したうえでビルドして実行すると、次のような表示になります。

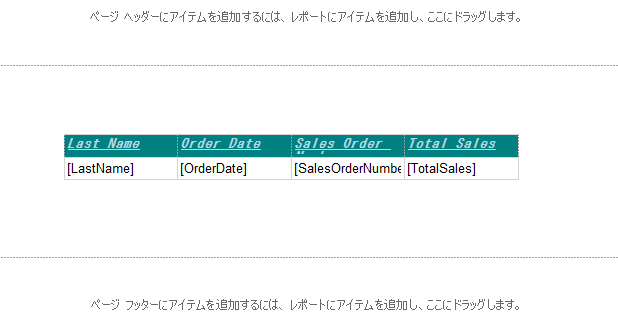


### ヘッダーとフッターの設定

印字した時のページ毎にヘッダーとフッターを指定できます。

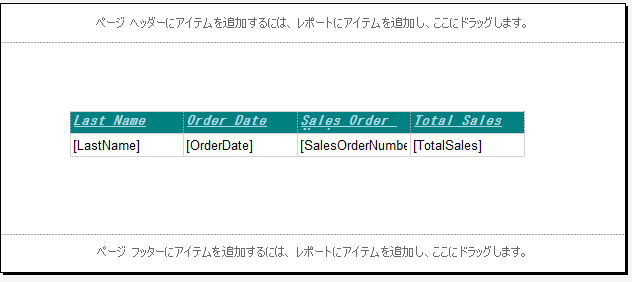
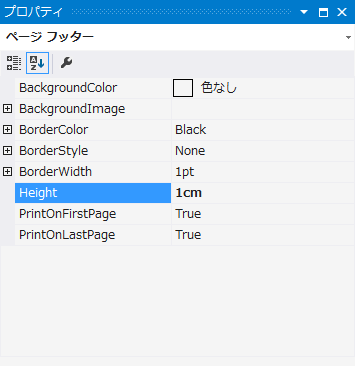
レポートデザイナを編集中に表示される「レポート」メニューより「ページヘッダーの追加」「ページフッターの追加」を選択してください。





これにより、ページヘッダーとページフッターを表示できるようになりました。

ヘッダー、フッターの高さを変更するにはヘッダー、フッターのプロパティのHeightで変更ができます。



ヘッダー、フッターには下記のコントロールのみが配置可能です。テーブルやマトリックスといったコントロールは配置できません。

**ヘッダー、フッターに配置可能なコントロール**

テキストボックス

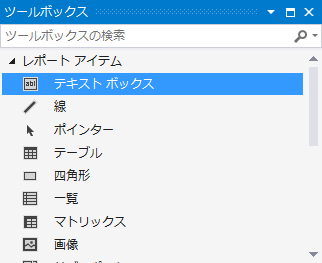
線

四角形

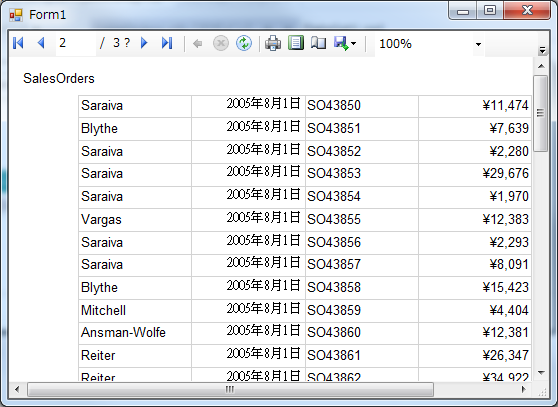
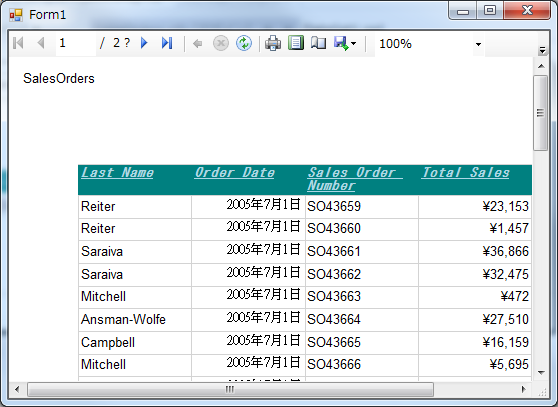
画像

#### ヘッダー、フッターに固定の文字を表示

この例ではヘッダーに帳票名を表示します。ヘッダーにツールボックスよりテキストボックスを選択してヘッダーに張り付けます。貼り付けたテキストボックスに任意の文字を入れてください。

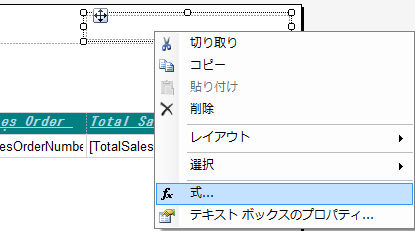


テキストボックスの枠線が表示されていますが、実際、ReportViewerコントロールを実行すると非表示になっています。実際に実行すると複数ページにわたり、入力した文字が出力されていることを確認できます。



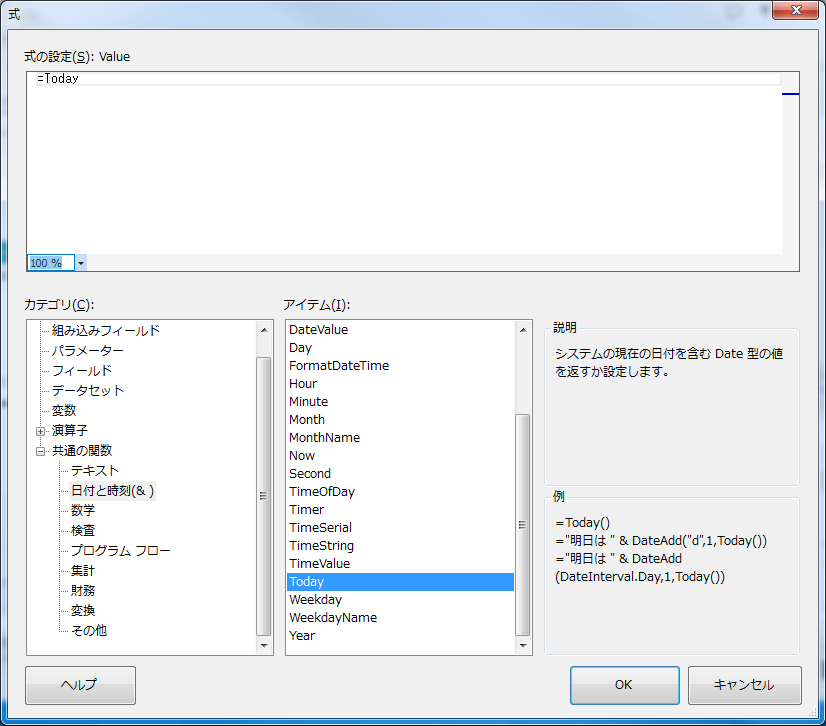
#### ヘッダー、フッターに現在の日付を表示する

テキストボックスには固定の文字だけでなく式を入力することで、日付などを表示することが可能です。固定文字の例のようにヘッダーまたはフッターにテキストボックスを張り付けます。その後、右クリックでメニューを表示して「式」を選択します。

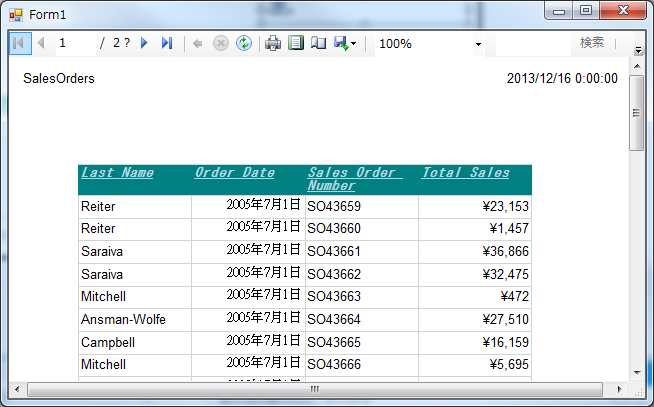


「式」ダイアログを表示さるので、ここに式を入力することが可能です。式を直接入力することも可能ですし、「カテゴリ」一覧から使用できる数式を選択ですることも可能です。

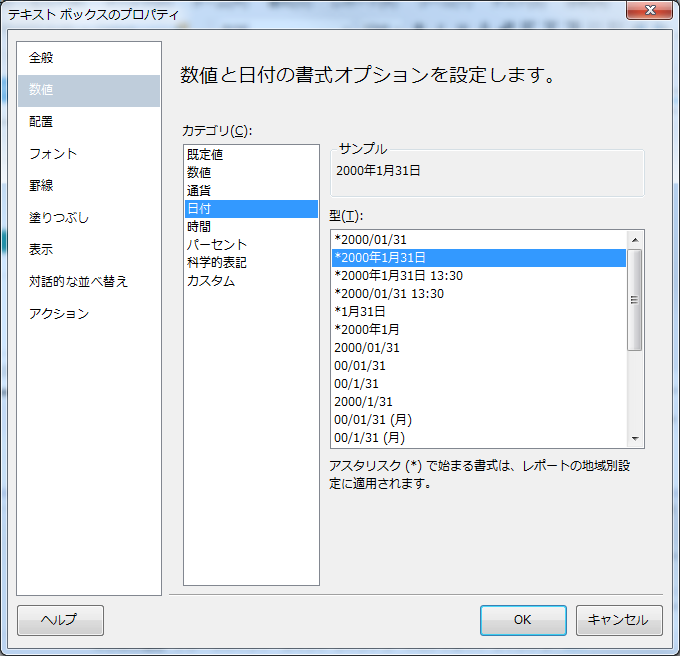
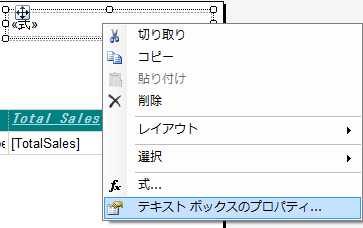
今日の日付を表示するには「カテゴリ」の「共通の関数→日付と時刻」を選択し、アイテムで「Today」を選択します。



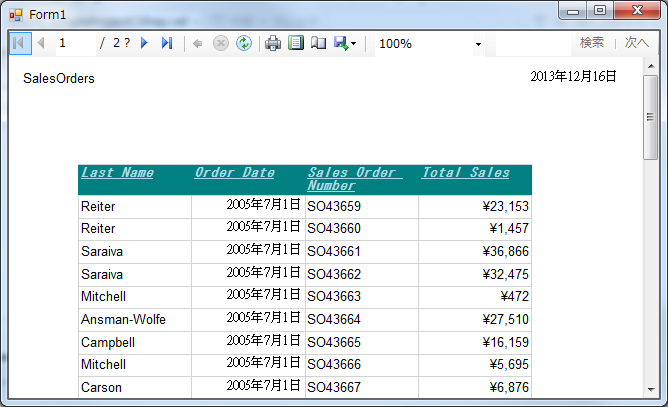
この時点で表示をすると「2013/12/16 0:00:00」というような書式になります。



テキストボックスの書式は、テキストボックスのプロパティから変更が可能です。



これにより以下のように適切な書式の日付が表示されます。



#### ヘッダー、フッターにページ数

日付と同様に、テキストボックスの式を用いてページ数を表示することが可能です。

ヘッダーにテキストボックスを張り付けて以下の式を入力します。

|  |
| --- |
| =Globals!PageNumber & "/" & Globals!TotalPages |

式で文字として連結を行う場合は「＆」演算子を使用します。

これにより、ヘッダーは次のように、「現在のページ数 / 総ページ」を表示します。

式には様々な機能が存在しており、次のページから確認することが可能です。

レポートの一般的な式 (Visual Studio レポート デザイナ)

<http://msdn.microsoft.com/ja-jp/library/ms251668%28v=vs.90%29.aspx>

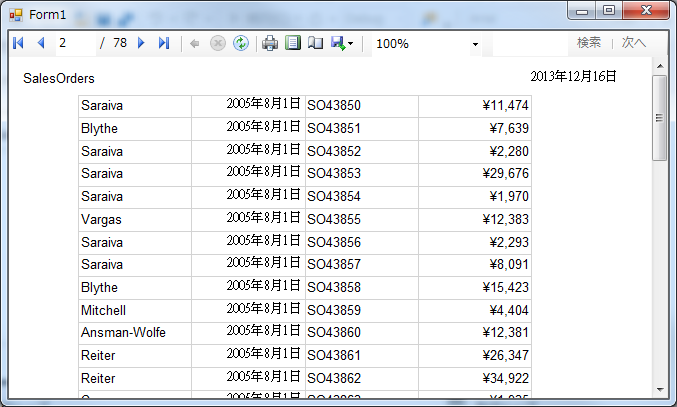
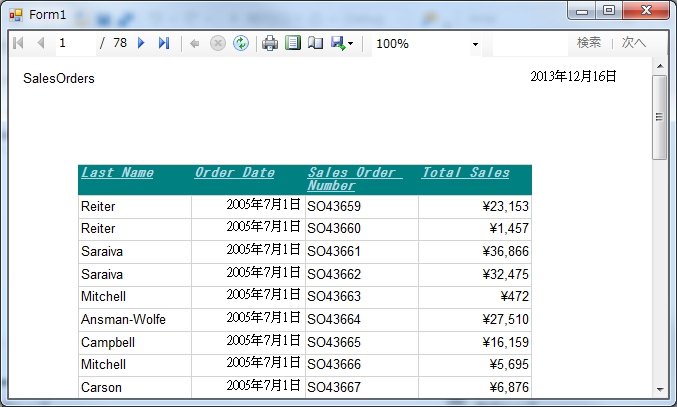
式のリファレンス (レポート ビルダーおよび SSRS)

<http://msdn.microsoft.com/ja-jp/library/ee240847.aspx>

開発者は独自の数式を作成することも可能で、その方法は「[カスタムコードの作成](#_カスタムコードの作成)」にて説明します。

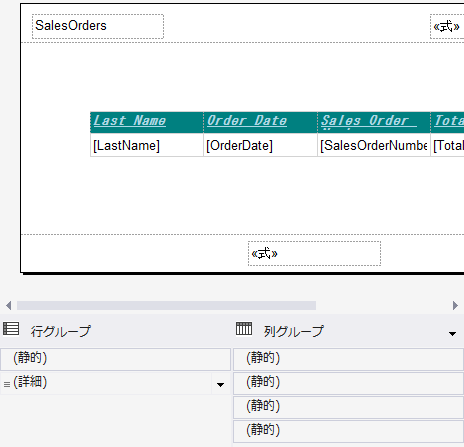
### 各ページにテーブルのヘッダーを表示する

ヘッダーとフッターの説明を行いました。しかし、説明した方法では大量のデータを保持するテーブルのヘッダーを表示ができません。

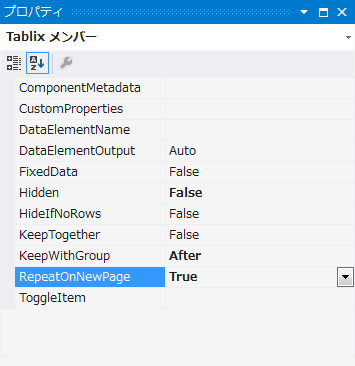


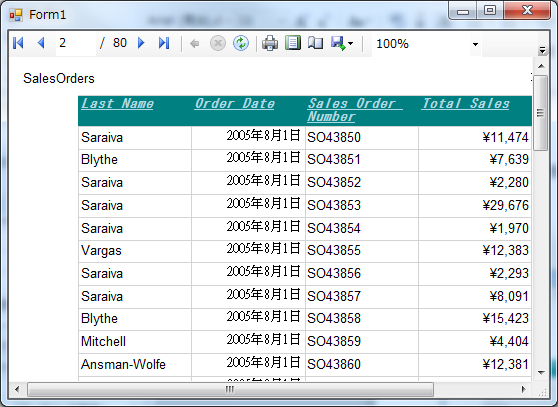
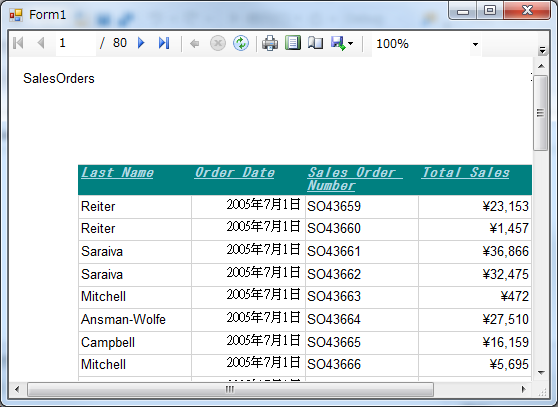
次のページに表示したい

テーブルのヘッダーをページ毎に表示するには、レポートデザイナの行グループのプロパティを修正する必要があります。そのために、まず、行、列グループの「詳細設定モード」にチェックをつけます。



詳細設定モードをチェックすることで、行グループ、列グループに「(静的)」というアイテムが表示されます。各ページでテーブルの行のヘッダーを繰り返し表示するには、行グループの「(静的)」を選択して、プロパティの「RepeatOnNewPage」をTrueにしてください。

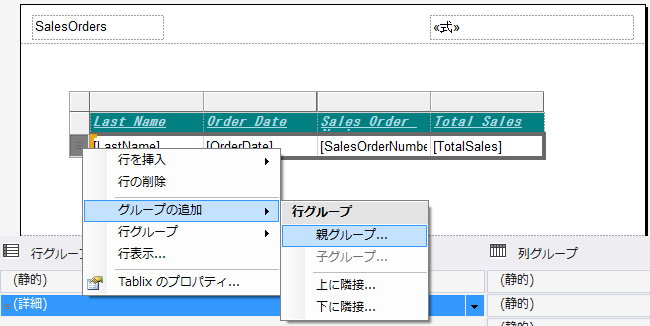
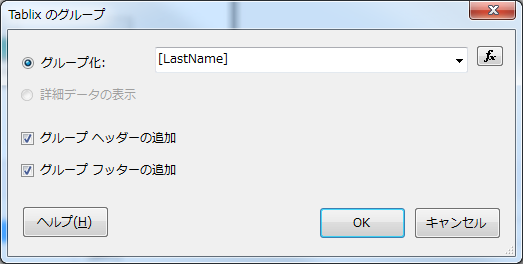
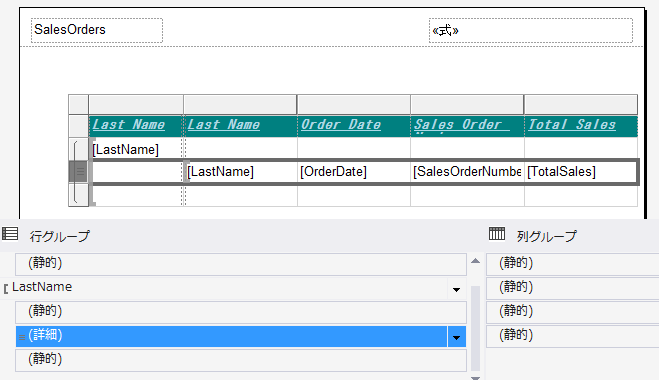
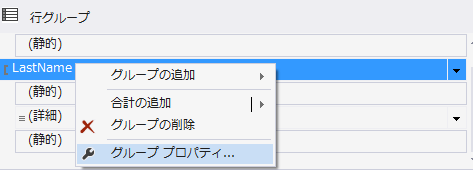


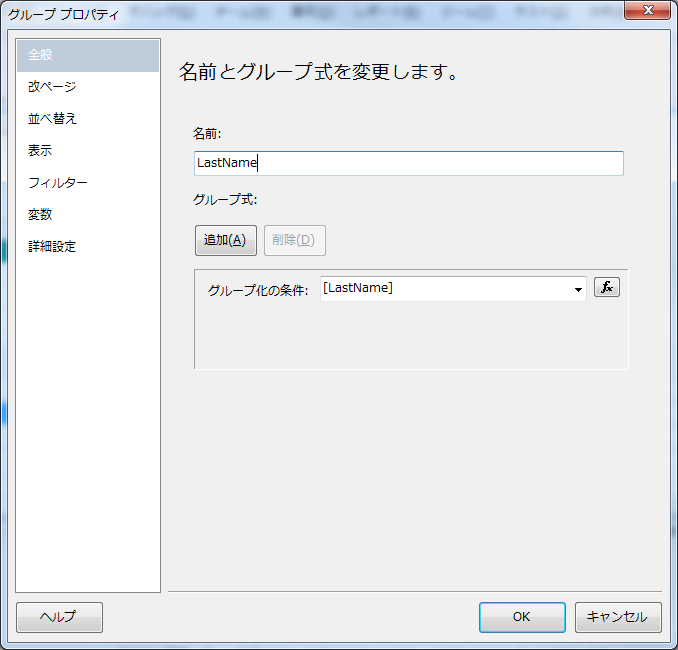
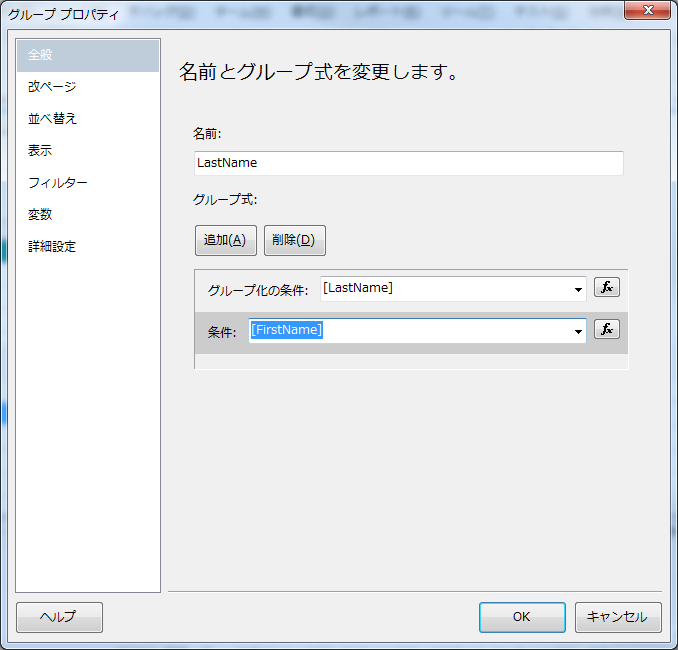


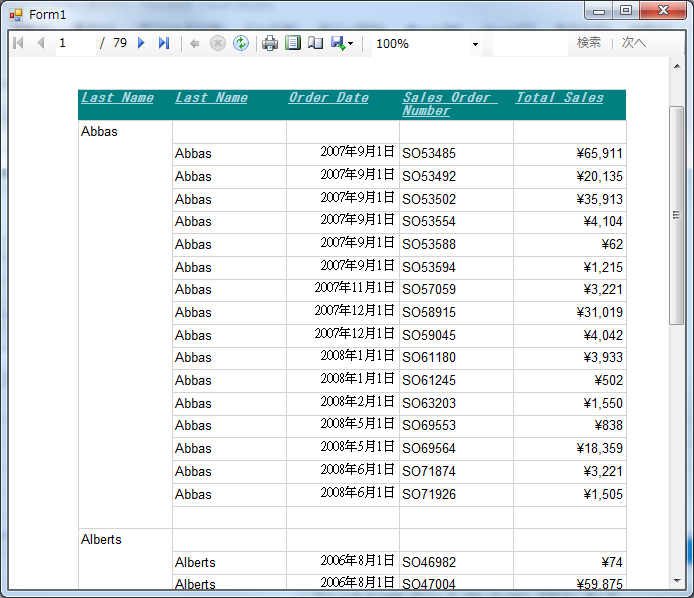
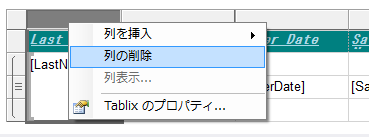
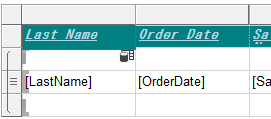
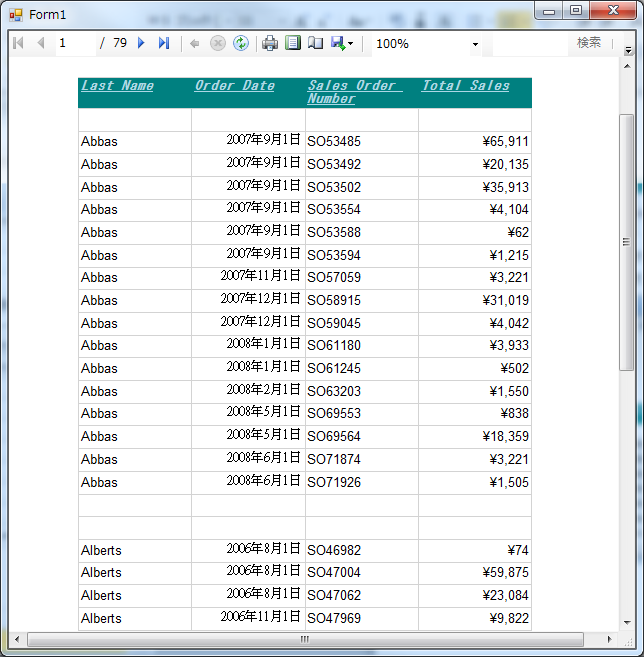
これにより、複数ページにわたりテーブルのヘッダーを表示することができます。

### テーブルのグループを定義するには

ReportViewerコントロールのテーブルでは、与えられたデータをグループ化して表示することができます。今回の例では、LastNameとFirstNameでデータをグループ化して各個人の売り上げの帳票を出力します。

1. 行ハンドルを右クリックして「グループの挿入」＞「親グループ」を選択します。  
   
2. グループダイアログが表示されるので、グループ化のコンボボックスで「LastName」を選択し、「グループヘッダーの追加」と「グループフッターの追加」にチェックをします。  
     
     
   これより、テーブルにはグループの名称を表示する列と、行グループにグループが追加されました。
3. この例ではグループ化の条件が[LastName]だけです。これでは同じLastNameを持つ別の人物も同じグループにしてしまいます。そのため、グループ化の条件に「FirstName」も追加する必要があります。  
   まず、行グループで条件を追加したいグループを選択して右クリックして「グループプロパティ」を選択します。  
   

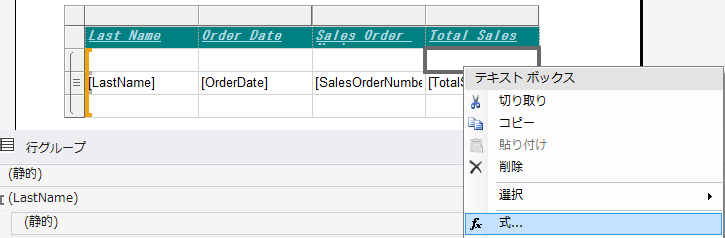
  
「グループプロパティ」ダイアログではグループ化についてのプロパティを表示確認できます。条件を追加するには、グループ化の追加ボタンを押して条件に「FirstName」を追加します。  


1. この状態でレポートを出力すると次のようになります。  
     
   各グループの先頭と、末尾にグループヘッダーとグループフッターが付与されていることが確認できます。
2. この帳票ではグループ化を行う際に、テーブルに列が自動的に挿入されましたが、これは表示にしようするだけなので、削除してもグループ化は解除されません。  
   テーブルで挿入された列を右クリックして、「列の削除」を選択します。  
   　   
   この状態でレポートを作成すると次のように、グループ用の列が存在しなくても、グループ化されていることがわかります。  
   

|  |
| --- |
| グループを削除するには？  グループを削除するには、削除したい行グループを選択して、右クリックで「グループの削除」を行います。 |

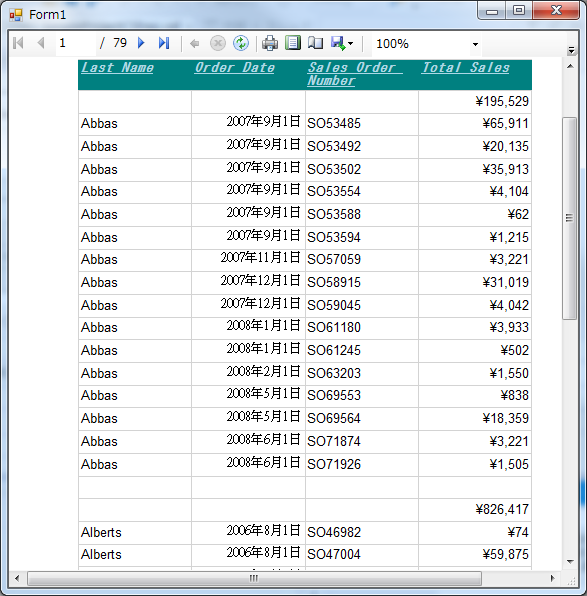
### データをグループ別に集計する

グループ化を行うことで、グループ別にデータを集計することができます。今回はセールスマン毎での売り上げの合計を求めます。

1. グループヘッダーで合計を出力したい箇所を右クリックして「式」を選択します。  
   
2. 式ダイアログが表示されるので、「式の設定」に次の式を入力します。

|  |
| --- |
| =Sum(Fields!TotalSales.Value) |

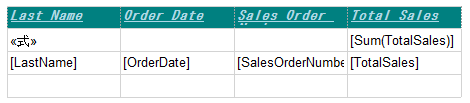
1. 追加を行ったら、このセルに対して通貨型の書式を設定します。「[フィールドの書式を指定する](#_フィールドの書式を指定する)」を参考にしてください。
2. これを実行することで、各グループのヘッダーに売り上げの総計が表示されるようになります。

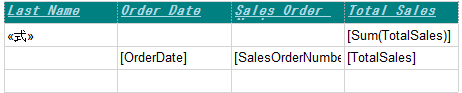
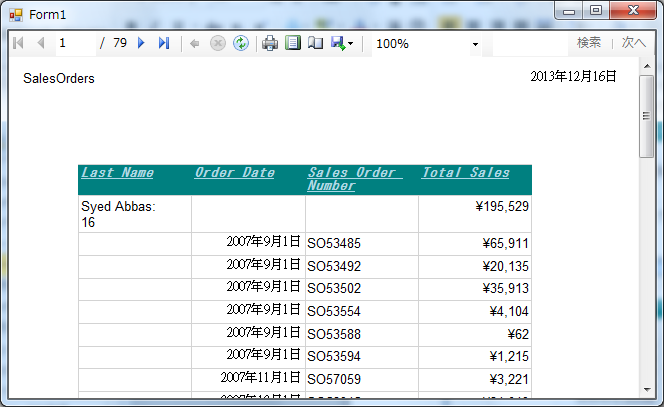
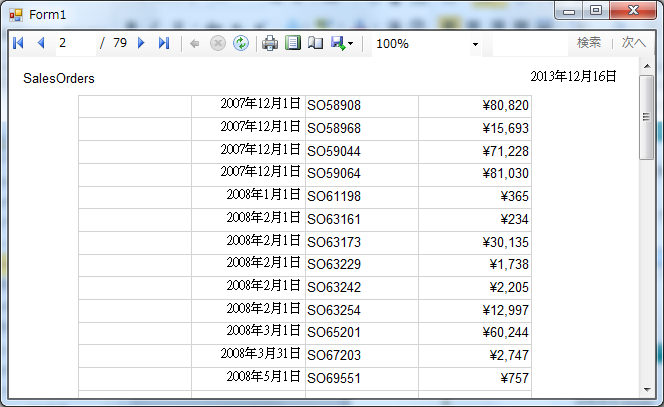
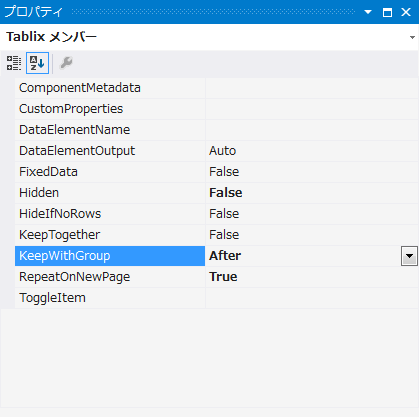
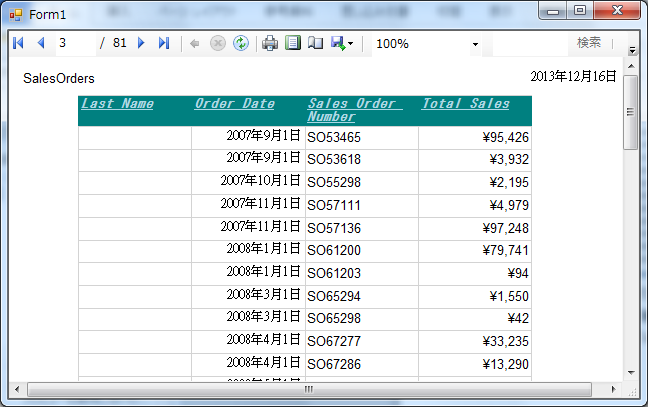
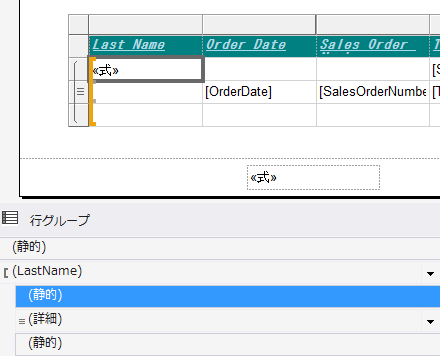


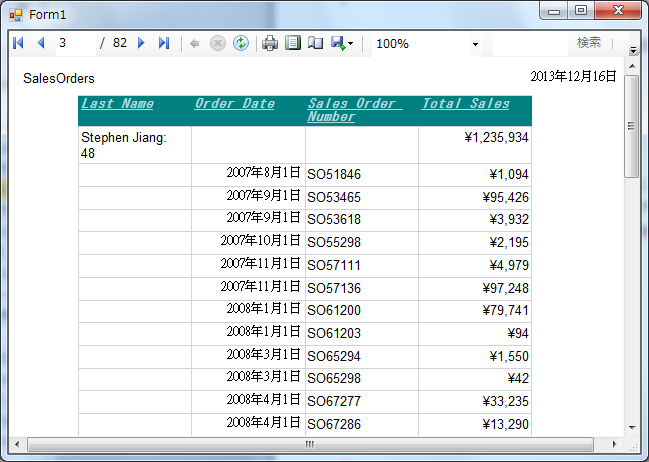
次にグループヘッダーにフルネームとデータ件数を表示します。

1. テーブルの１行目、１列目に式を入力します。  
   次の式を入力します。

|  |
| --- |
| =Fields!FirstName.Value + " " + Fields!LastName.Value + ": " + vbCrLf + Count(Fields!SalesOrderNumber.Value).ToString() |

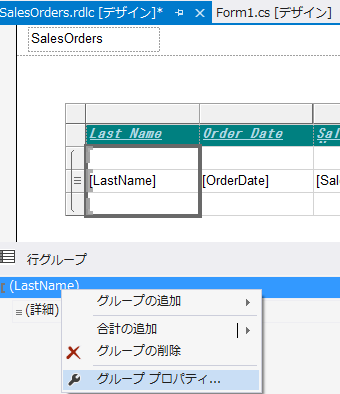


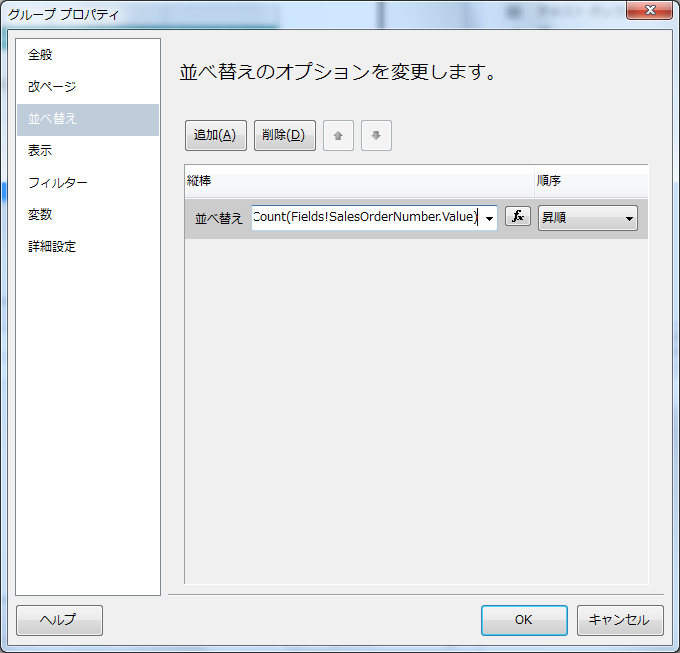
1. 「LastName」が行毎に表示されているので、それを削除します。これにより、名前が行ごとにでるのではなく、グループヘッダーのみに表示されます。  
   
2. この状態でレポートを出力することでグループヘッダーにフルネームとデータ件数が表示されます。  
   
3. レポートで、次のページをみてみます。以下のように先ほど設定したはずのテーブルのヘッダーが表示されなくなっています。  
     
   これはグループ化を行ったため、行グループの状態が、さっきと変更されたために発生します。
4. 行グループの一番上の「静的」のプロパティで「RepeatOnNewPage」を「True」、「KeepWIthGroup」を「After」にしてください。  
   　
5. その後、実行すると次ページ以降にもテーブルのヘッダーが表示されます。  
   
6. 次にグループが複数ページにまたがった場合に、グループヘッダーを同様にページの先頭に表示するようにします。行グループで、グループ名の直下の「静的」を選択して同様にプロパティの「RepeatOnNewPage」を「True」、「KeepWIthGroup」を「After」にしてください。  
   
7. これにより、各ページの先頭にグループのヘッダーが表示されるようになります。



### 表形式のレポート内のグループを並べ替えるには

ReportViewerでは並び替えの機能を提供しています。これにより、作成したグループを並べ替えることができます。この例ではグループが保持するデータの数順に並び替えます。

1. 行グループの最上位のアイテムを選択して右クリックして、「グループプロパティ」を選択します。  
   
2. 「グループプロパティ」ダイアログが開くので「並べ替え」を選択して「追加」ボタンをおします。



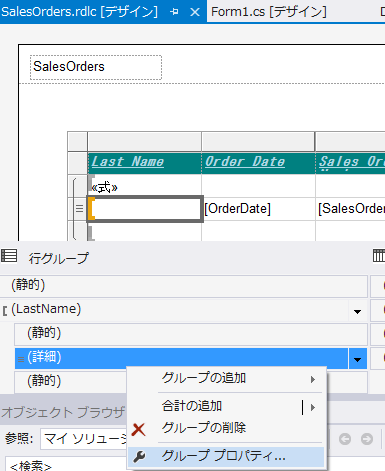
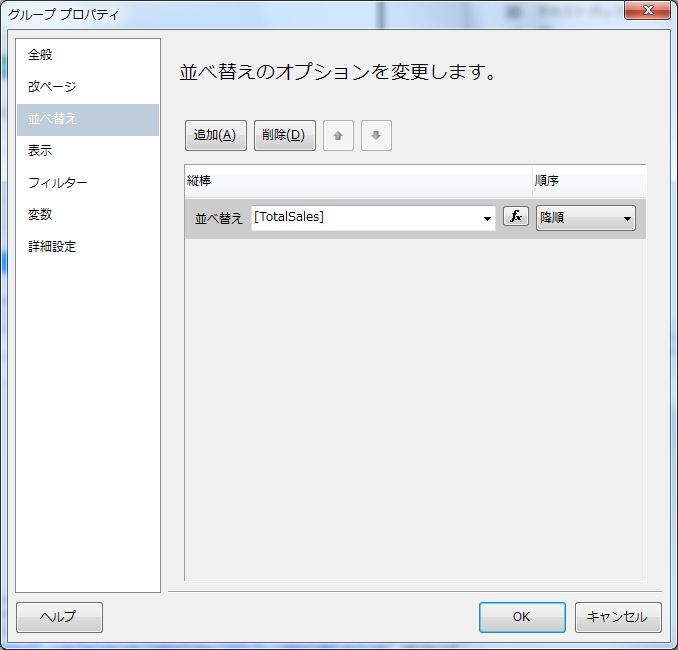
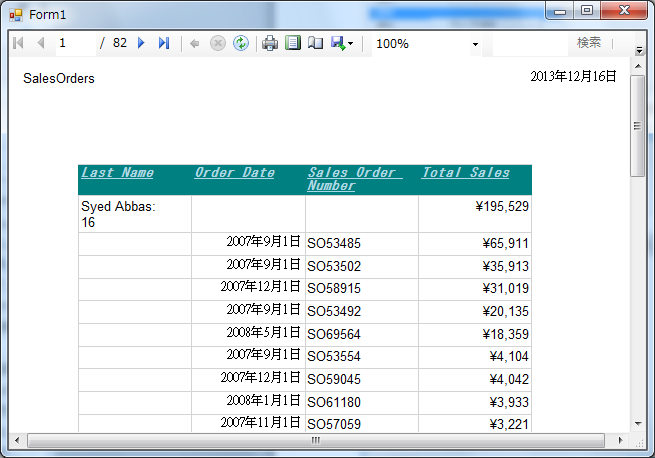
並び替えの条件には以下の式、順序は「昇順」としてOKを押してください。

|  |
| --- |
| =Count(Fields!SalesOrderNumber.Value) |

1. ここで実行を行うと販売数の少ない人ほど最初のページに表示されるようになります。

### 表形式のレポートのグループ内の詳細行を並べ替えるには

次にグループ内の詳細行を指定の順序に並びかえます。この例では売上額が高い順に並び替えます。

1. 行グループで、グループ内の詳細を選択して「グループプロパティ」を選択します。  
   
2. グループプロパティダイアログで、「並び替え」に縦棒を追加します。  
   [TotalSales]をコンボボックスで選択して、順序を「降順」にしてください。  
   
3. ここで実行すると、売り上げの降順でグループ内が並び替えされているようになります。  
   

## マトリックスを用いたクロス集計

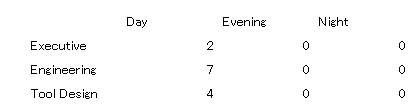
この項目ではマトリックスを用いたクロス集計を行います。クロス集計とは項目を掛け合わせて、それぞれの項目が交わるセルに合計値や数などを記載した表です。

このチュートリアルは下記のMicrosoftのチュートリアルを説明しています。

チュートリアル : ローカル処理モードでのデータベース データ ソースと ReportViewer Windows フォーム コントロールの使用

<http://msdn.microsoft.com/ja-jp/library/ms251724%28v=vs.90%29.aspx>

このサンプルでは縦軸に部署、横軸にシフト（朝、夕、夜）をとり、それぞれ従業員が何人出勤しているかを表示します。



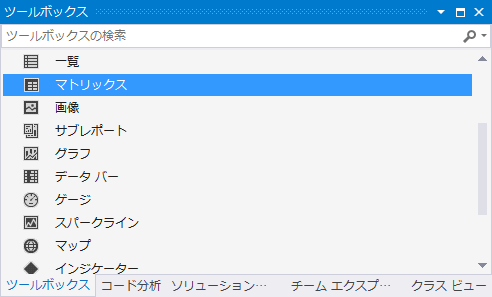
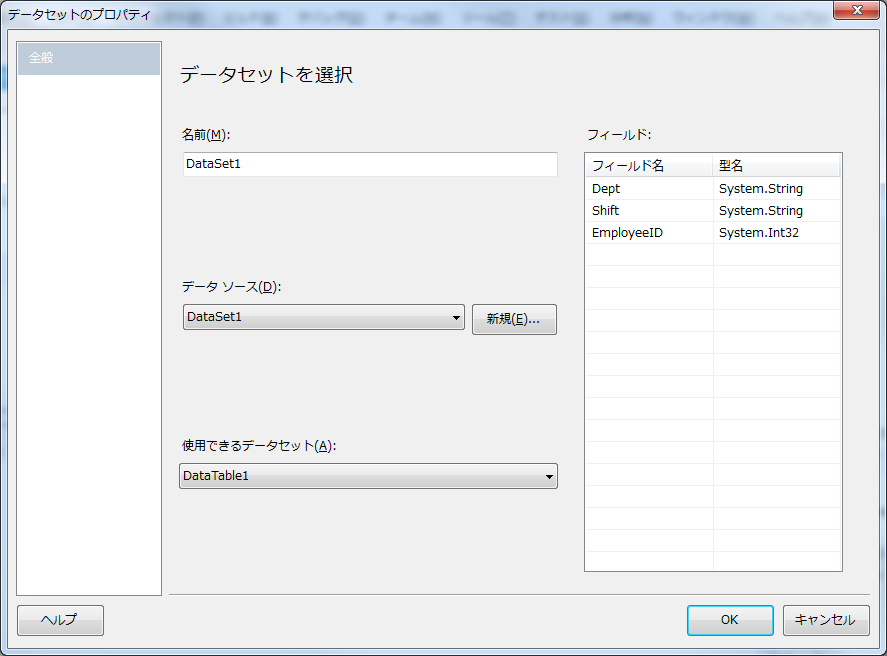
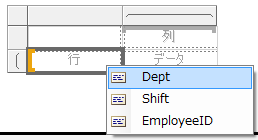
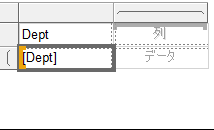
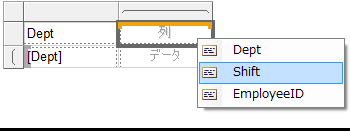
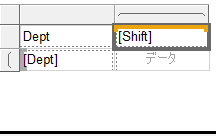
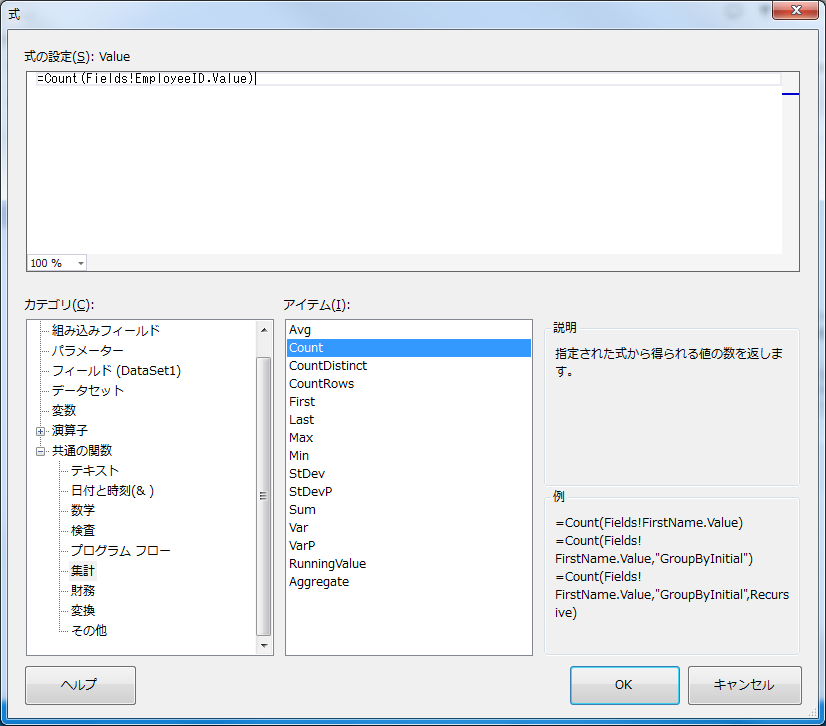
### データソース接続とDataTable定義

レポートに使用するDataSetの作成を行います。

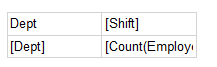
1. [プロジェクト]メニューの[新しい項目の追加]をクリックします
2. [新しい項目の追加]ダイアログで[データセット]を選択して追加します。この際、任意の名称を入力できます。ここでは規定の名前であるDataSet1.xsdを使用します
3. ツールボックスから[TableAdapter]をデータセットデザイン画面にドラッグします。これによりウィザード画面が表示されます。先のチュートリアルで実行した『[「データソース接続」と「データテーブル」を定義します](#_「データソース接続」と「データテーブル」を定義します)』を参照してAdventureWorks2012に接続をします。
4. SQLステートメントの入力まで進めて次のSQLを入力します。

|  |
| --- |
| SELECT d.name as Dept, s.Name as Shift, e.BusinessEntityID as EmployeeID  FROM (HumanResources.Department d  INNER JOIN HumanResources.EmployeeDepartmentHistory e  ON d.DepartmentID = e.DepartmentID)  INNER JOIN HumanResources.Shift s  ON e.ShiftID = s.ShiftID |

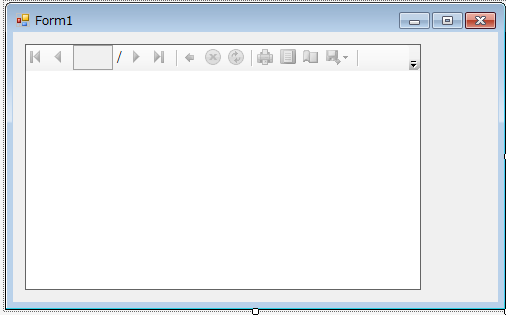
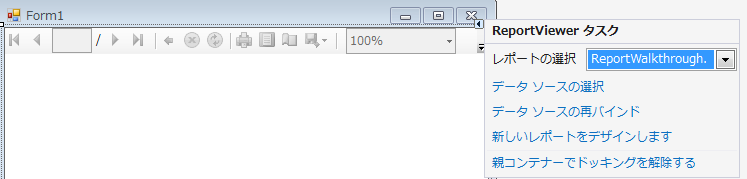
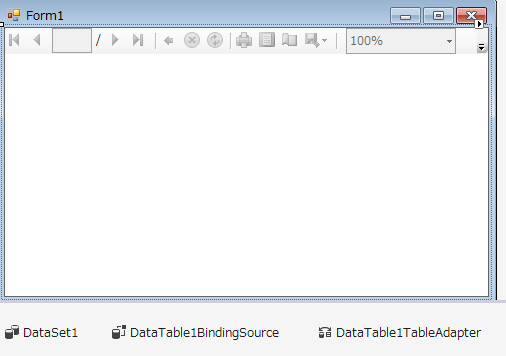
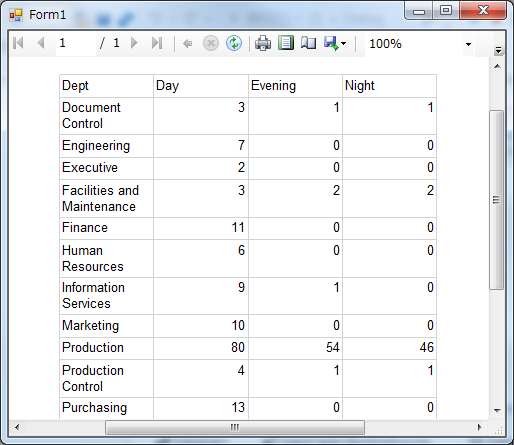
### レポートのデザイン

1. [プロジェクト]メニューの[新しい項目の追加]をクリックします
2. Reportingの[レポート]を選択します。これでレポートデザイナが起動します。
3. ツールボックスのマトリックスを選択してレポートデザイナにドロップしてください。  
   
4. データセットのプロパティが起動するので、先ほど追加したDataSetを選択してください。  
     
   これでOKを押すことでマトリックスがレポートに追加されます。  
   
5. 「行」と記述しているセルに部署を表すDeptを選択します。  
   　
6. 「列」と記述しているセルにシフトを表すShiftを追加します  
   　　
7. 「データ」とあるセルを右クリックして式プロパティダイアログを表示して式を入力します。  
   

|  |
| --- |
| =Count(Fields!EmployeeID.Value) |



### アプリケーションへのReportViewerコントロールの追加

1. FormデザイナにおいてツールボックスのReportViewerコントロールをFormに張り付けます。  
   
2. ReportViewerコントロールのプロパティの[Dock]を[Fill]に指定します。これでReportViewerコントロールはForm大きさに一致するようになります
3. 右上隅の三角形をクリックしてReportViewerタスクメニューを表示して、レポート選択コンボボックスで作成したレポートを選択します。  
   
4. 指定したレポートにデータを表示するためにDataSetコントロール、DataTableBindingSourceコントロール、DataTableAdapterコントロールが追加されます。  
   
5. この時点でビルドして実行すると次のようなクロス集計を行うレポートが作成されます。  
   

## 単票形式の帳票作成

このチュートリアルでは一覧形式では表現できない帳票を作成します。

この例では以下のようなユーザー情報を名刺として印字するような帳票を作成します。

この例では画像データを使用するので、下記からダウンロードするか、ご自分でPNGファイルをご用意してください。

<http://needtec.sakura.ne.jp/doc/ReportCard.zip>

### データソースとして使用するビジネスオブジェクトの作成

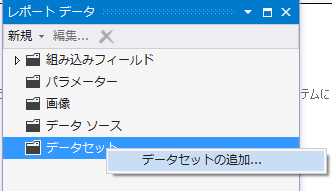
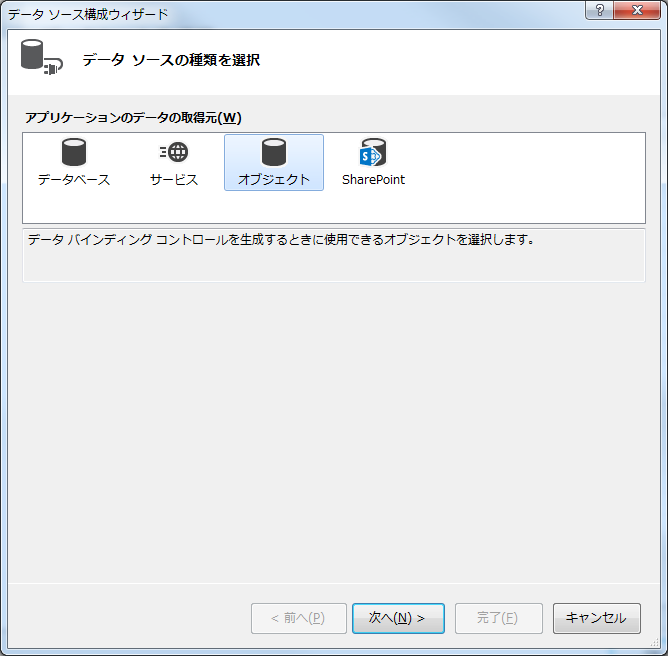
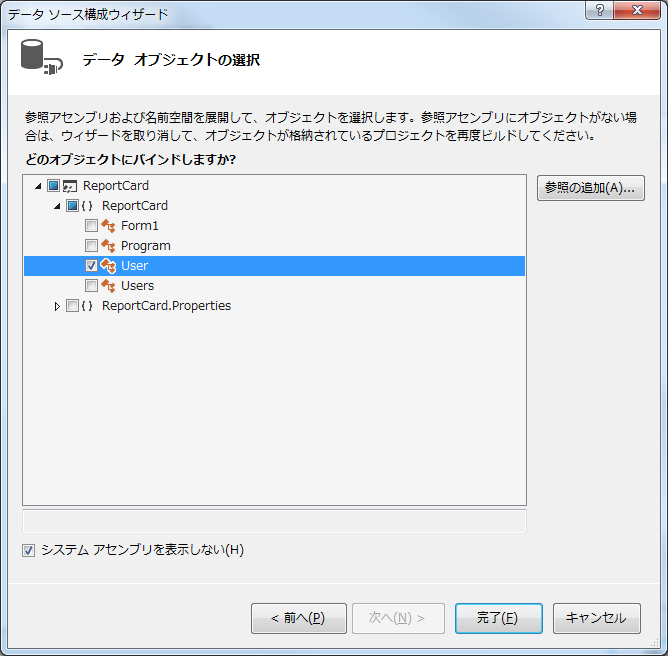
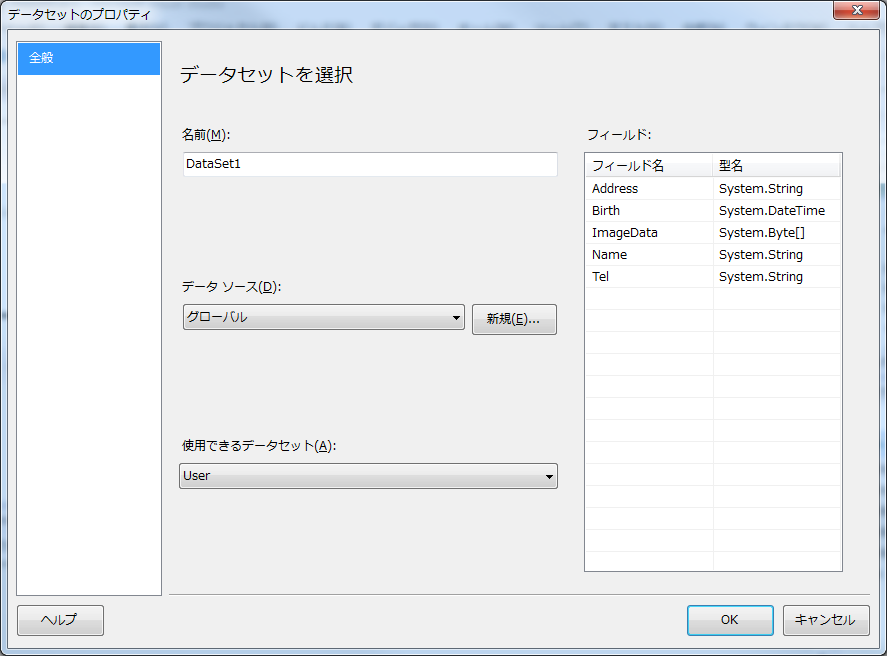
今回の使用するデータはデータベースではなく、クラスで定義した情報を使用します。

Users.csというクラスを追加して以下のような実装をします。

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Drawing;  class User  {  public string Name { get; set; }  public DateTime Birth { get; set; }  public string Tel { get; set; }  public string Address { get; set; }  public Byte[] ImageData { get; set; }  public User(string name, string birth, string tel, string address, string imagepath)  {  Name = name;  try  {  Birth = DateTime.Parse(birth);  }  catch (Exception ex)  {  throw new ArgumentException("birthの書式が不正です\n" + ex.Message);  }  Tel = tel;  Address = address;  try  {  Image img = Image.FromFile(imagepath);  ImageConverter imgcv = new ImageConverter();  ImageData = (byte[])imgcv.ConvertTo(img, typeof(byte[]));  }  catch (Exception ex)  {  throw new ArgumentException("imagepathの指定が不正です\n" + ex.Message);  }    }  }  class Users  {  private List<User> users;  public Users()  {  users = new List<User>();  users.Add(new User("やる夫", "2005/1/1", "000-000-1111", "2ch ニュー即 VIPスレ 000-333-555 紙スレ", @"..\..\Data\img001.png"));  users.Add(new User("やらない夫", "2005/2/1", "000-000-1112", "2ch VIPスレ", @"..\..\Data\img002.png"));  users.Add(new User("ゆっくり霊夢", "2008/1/1", "000-001-1111", "ニコニコ　実況", @"..\..\Data\img003.png"));  users.Add(new User("ゆっくり魔理沙", "2008/2/1", "000-001-1112", "ニコニコ　実況", @"..\..\Data\img004.png"));  users.Add(new User("はちゅねミク", "2009/3/1", "000-001-1113", "ニコニコ　実況", @"..\..\Data\img005.png"));  }  public List<User> GetUsers()  {  return users;  }  } |

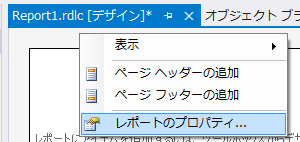
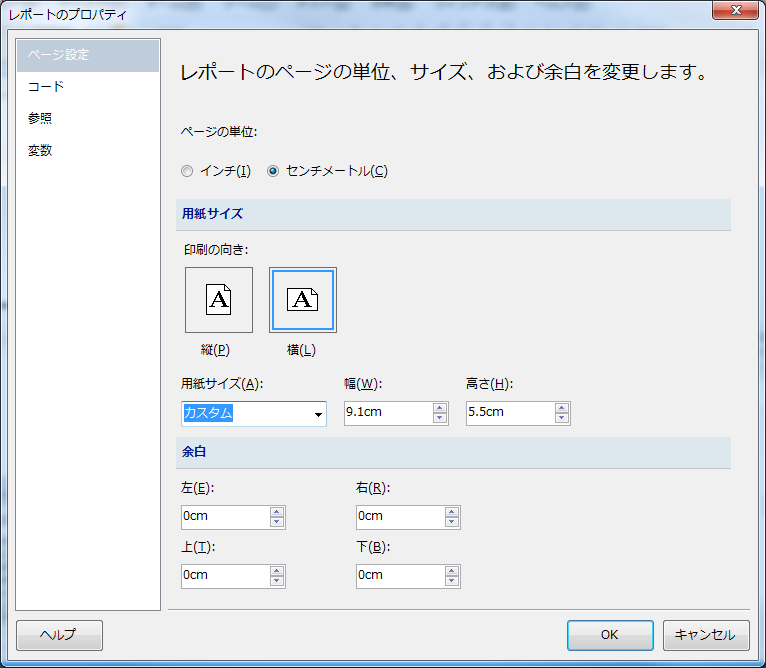
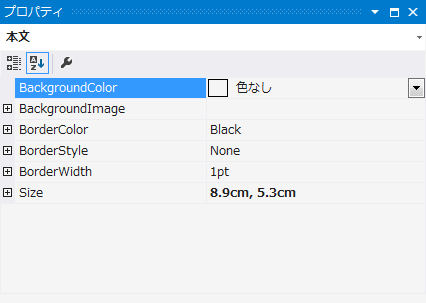
### レポート定義にオブジェクトを関連づける

ここではレポート定義ファイルを作成してデータソースとしてオブジェクトを関連付ける方法を記述します。

1. プロジェクトメニューの「新しい項目の追加」でレポートを追加してください。
2. レポートのデザイナを選択中に「表示」メニューから「レポートデータ」を選択することで「レポートデータ」ウィンドウが表示されます。これはCTRL＋ALT＋Dで代替できます。
3. レポートデータウィンドウにて、「データセット」を右クリックして「データセットの追加」を選択してください。  
   
4. データソース構成ウィザードが起動するので、ウィザードを利用してデータソースを作成します。  
   まずオブジェクトを選択して「次へ」を押します  
     
   オブジェクトの一覧が表示されるので作成したUserオブジェクトが表示されるまで展開をしてUserオブジェクトにチェックをつけ「完了」します。  
   
5. データセットプロパティ　ダイアログでデータセットが作成されるのでOKを押します。  
     
   以降、レポート定義ではデータベースをデータソースとした時と同様に、データを用いてレポートを作成できます。

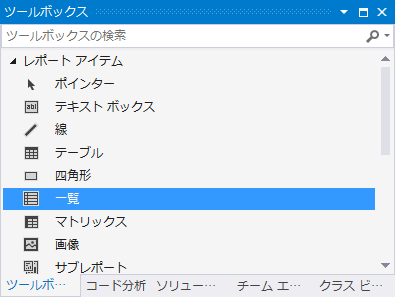
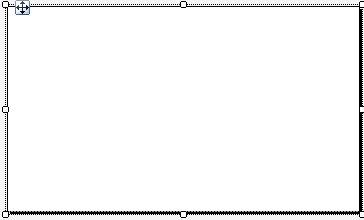
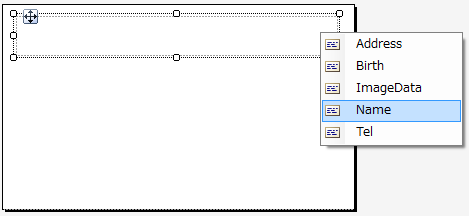
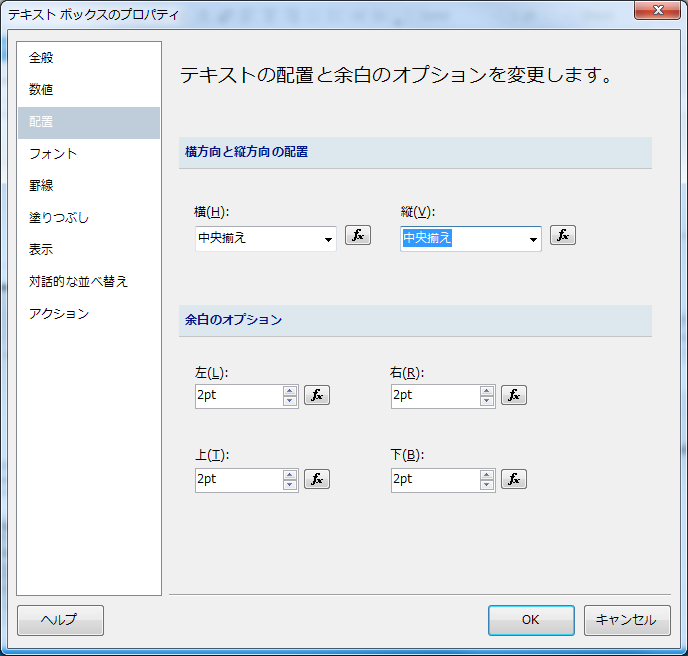
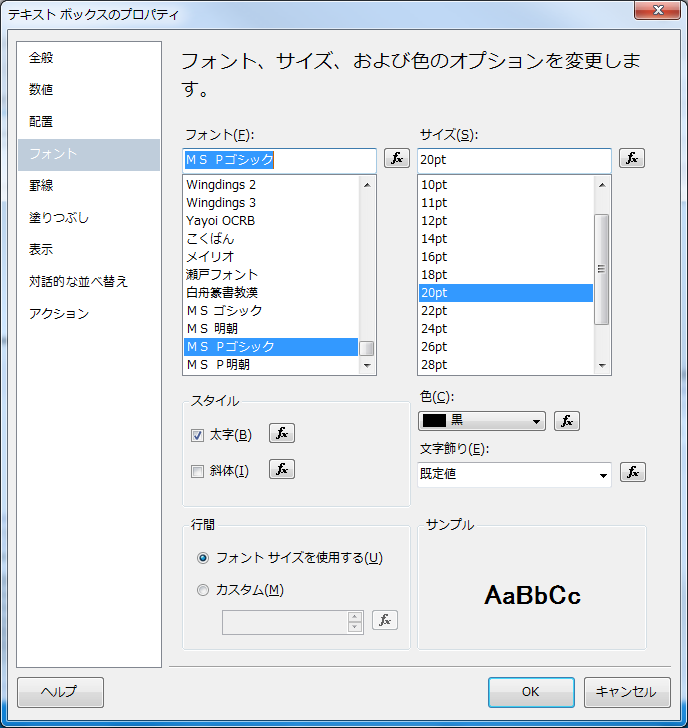
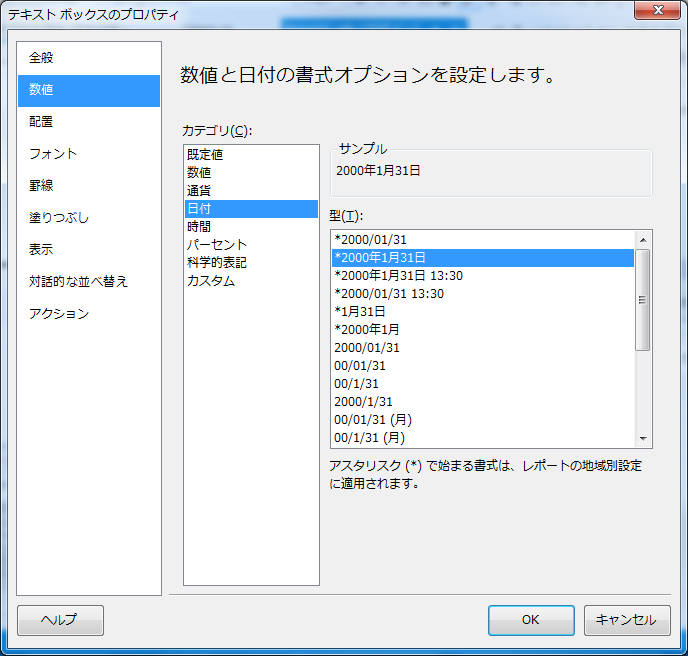
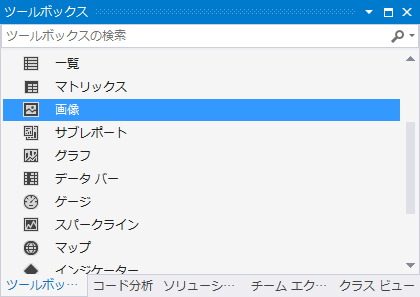
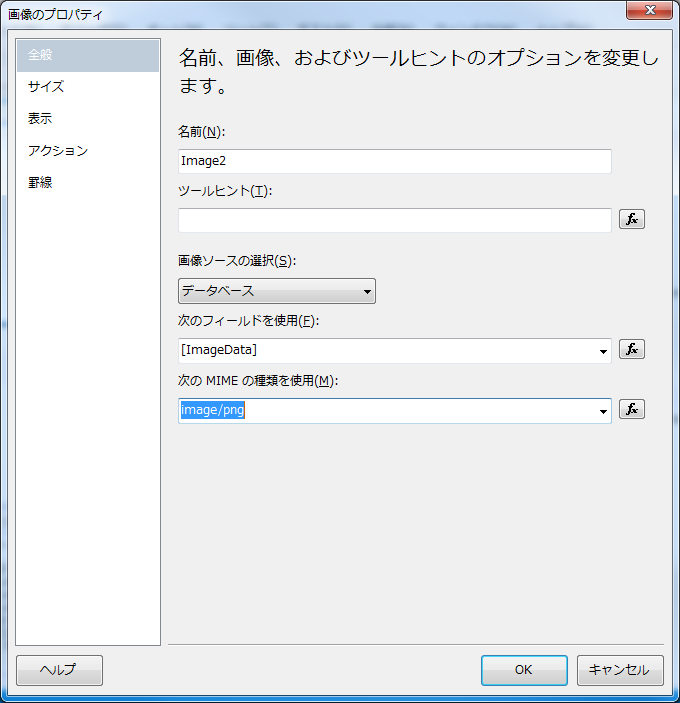
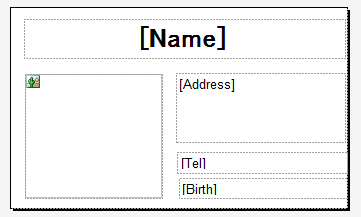
### レポートの大きさを名刺サイズにする

レポートの大きさを9.1cm x 5.5cmの名刺サイズに変更します。

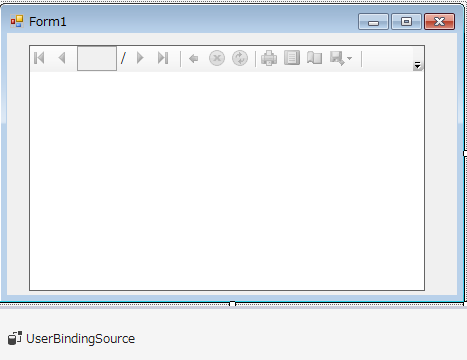
1. レポートデザイナで灰色の余白部分を右クリックして「レポートのプロパティ」を選択します。  
   
2. ページ設定で用紙のサイズを「カスタム」にして「幅」9.1cm、「高さ」5.5cmとします。また、余白はすべて0cmとします。  
   
3. レポートデザイナで本文を選択してプロパティのSizeで用紙の大きさを指定します。用紙と同じサイズだとはみ出て、改行してしまうので、２，３ｍｍ程度少なく設定します。  
   

### 一覧データによる繰り返しデータの自由形式でのレイアウト

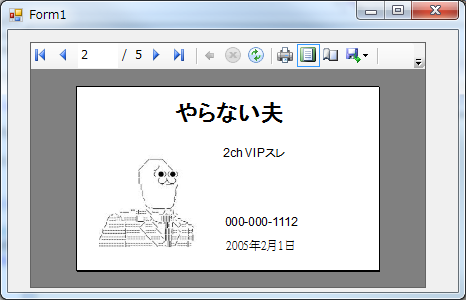
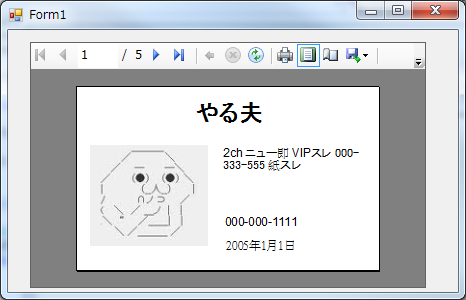
ここでは一覧データを用いて、繰り返しデータを一覧表という形式ではなく自由なレイアウトでデザインします。

1. レポートデザイナにツールボックスから「一覧」を選択してドロップします。この際、作成された一覧の大きさを本文いっぱいに広げてください。  
   　
2. 名前を表示するテキストボックスを追加します。  
   データのアイコンをクリックして「Name」を選択してください。  
     
   表示するデータを指定したら、レイアウトを指定します。テキストボックスのプロパティを開いて配置で縦、横を[中央揃え]にします。  
     
     
   次にフォントを指定します。MS Pゴシック、サイズを20pt、太字にチェックをつけます。  
   
3. Nameと同様に、Address、Tel、Birthに関してテキストボックスで表示します。Birthに関してはテキストボックスのプロパティの「数値」で書式を設定します。  
   
4. プロフィール画像を表示するために画像をドロップします。  
     
   ドロップをすると画像のプロパティが表示されるので、画像ソースの選択で「データベース」、次のフィールドを適用で「[ImageData]」を選択して、MIMEの種類は「image/png」にします。  
     
   ここでOKを押すことにより、画像が表示されるようになります。  
   

### アプリケーションへのReportViewerコントロールの追加

1. FormデザイナにおいてツールボックスのReportViewerコントロールをFormに張り付けます。
2. ReportViewerコントロールのプロパティの[Dock]を[Fill]に指定します。これでReportViewerコントロールはForm大きさに一致するようになります
3. 右上隅の三角形をクリックしてReportViewerタスクメニューを表示して、レポート選択コンボボックスで作成したレポートを選択します。  
   指定したレポートにデータを表示するためにUserBindSourceコントロールが追加されます。  
   
4. Form\_LoadイベントでRefreshReportを行う前に、UserBindSourceにデータを追加します。

|  |
| --- |
| private void Form1\_Load(object sender, EventArgs e)  {  Users users = new Users();  UserBindingSource.DataSource = users.GetUsers();  this.reportViewer1.RefreshReport();  } |

1. この状態でビルドを実行して印刷プレビューを見ると以下のようになります。  
   

このようにReportViewrを使用することで単票形式の帳票が容易に作成できることが確認できました。

## プレビューを使用しないローカルレポートの印字

ReportViewerコントロールでプレビューを表示しないで直接印字するには、ReportViewコントロールのLocalReportを画像にExportしてそのデータをプリンタに送ります。  
この方法は、以下に記述してあります。

チュートリアル : プレビューを使用しないローカル レポートの印刷

<http://msdn.microsoft.com/ja-jp/library/ms252091%28v=vs.90%29.aspx>

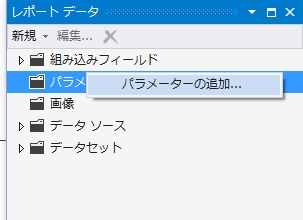
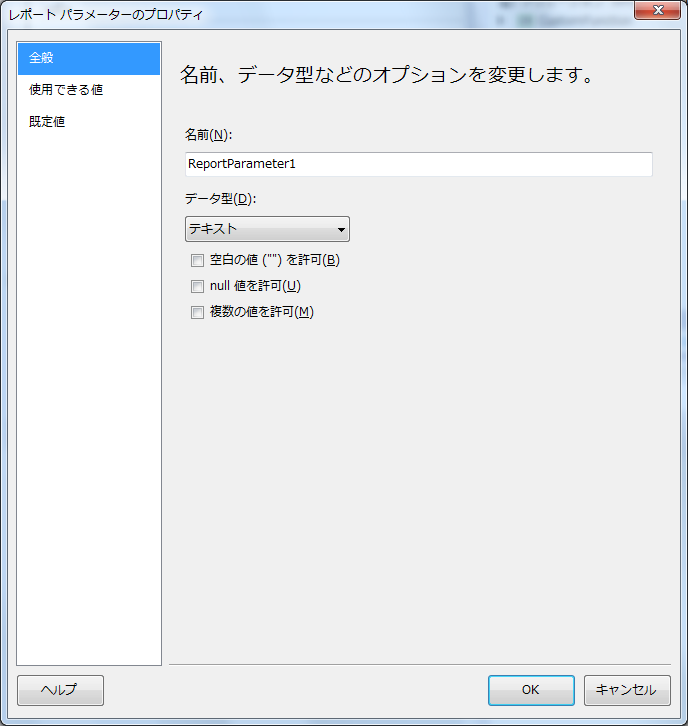
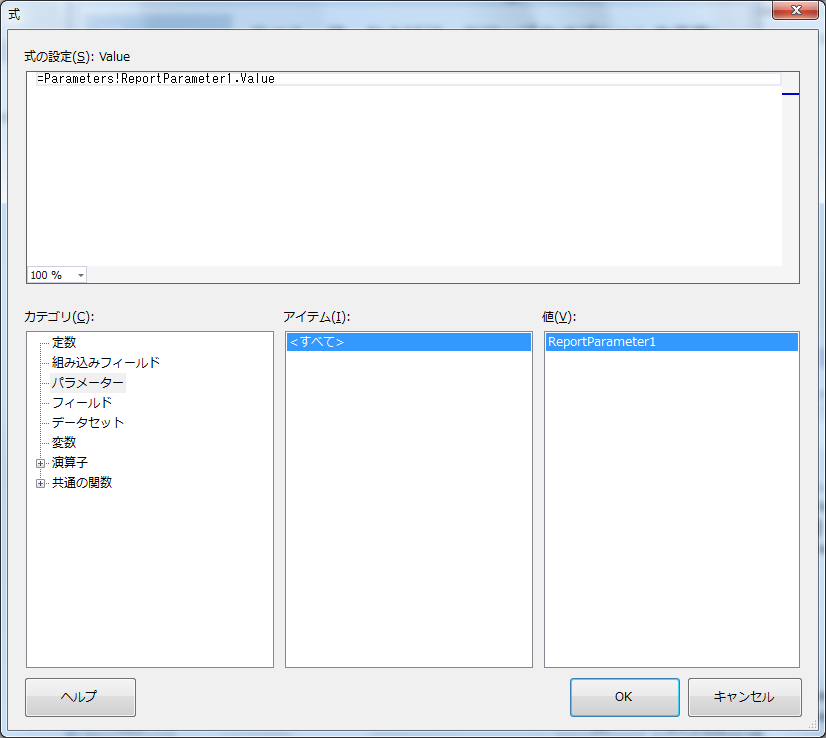
このチュートリアルを実行するまえに次のページのページを参照して、Report.rdlcファイルとData.xmlを作成してください。

印刷チュートリアルのサンプル データおよびレポート

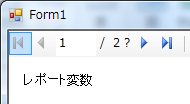
<http://msdn.microsoft.com/ja-jp/library/ms251734%28v=vs.90%29.aspx>

## レポートパラメータの使用方法

レポート定義ファイルでレポートパラメータを使用することで、実行時に任意の値を与えて、それをレポート中に表示することができます。

1. レポートのデザイナを選択中に「表示」メニューから「レポートデータ」を選択することで「レポートデータ」ウィンドウが表示されます。これはCTRL＋ALT＋Dで代替できます。
2. レポートデータウィンドウで「パラメーター」を選択して右クリックを押し、「パラメーターの追加」を押下します。  
   
3. 「レポート パラメーターのプロパティ」ダイアログで任意の名称を付与します。  
   
4. 以後、このレポート定義ファイルでは式に追加したパラメーターが使用できます。以下のように式ダイアログのカテゴリ「パラメーター」に今追加したパラメーターが増えているのを確認できます。  
     
   この例では帳票の左上にレポートパラメータを表示するテキストボックスを追加します。  
   
5. レポートコントロールが表示される前に、レポートパラメータを以下のようにして与えます。

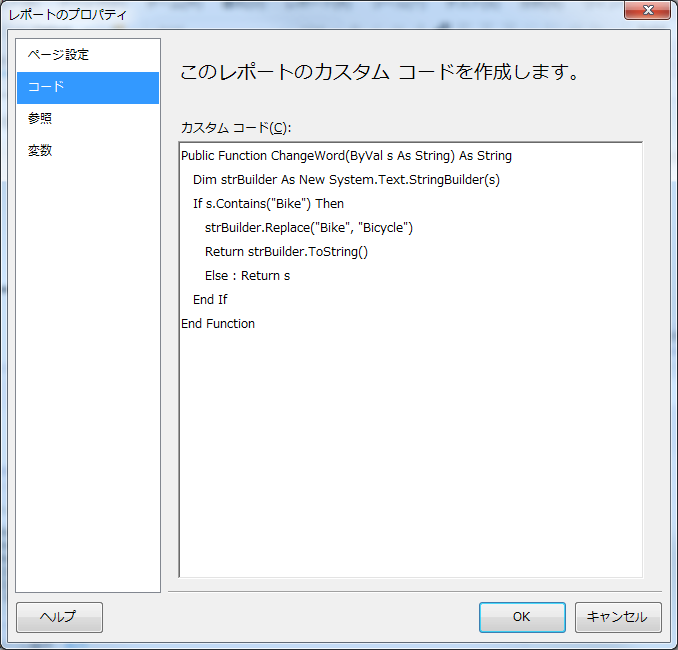
|  |
| --- |
| private void Form1\_Load(object sender, EventArgs e)  {  this.DataTable1TableAdapter.Fill(this.DataSet1.DataTable1);  // レポートパラメータを作成  ReportParameter param = new ReportParameter("ReportParameter1","レポート変数");  reportViewer1.LocalReport.SetParameters(param);  // レポート表示  this.reportViewer1.RefreshReport();  } |

1. 実行をするとプログラムで与えた文字がレポートに表示されます。  
   

## カスタムコードの作成

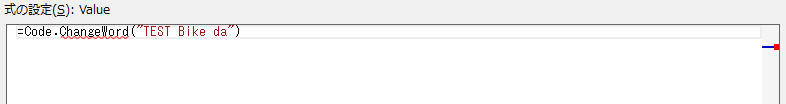
カスタムコードを用いることで、独自の数式をReportViewer内で使用できます。カスタムコードの書き方は２種類存在しており、レポート定義ファイルにコードを組み込む方法と、グローバルアセンブリを使用する方法があります。

### レポート定義ファイルにコードを組み込む

1. レポートデザイナ使用中に「レポート」メニューの「レポートのプロパティ」を選択して「レポートのプロパティ」を表示します。
2. コードを選択してカスタムコードに任意のコードを入力します。  
   

このコードは指定した文字列中にBikeという文字があったら、Bicycleに変更します。

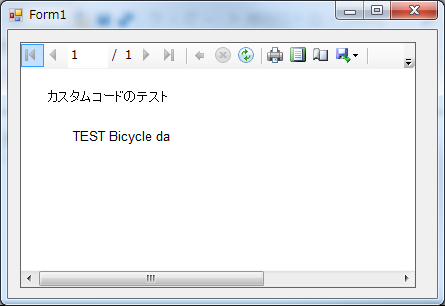
|  |
| --- |
| Public Function ChangeWord(ByVal s As String) As String  Dim strBuilder As New System.Text.StringBuilder(s)  If s.Contains("Bike") Then  strBuilder.Replace("Bike", "Bicycle")  Return strBuilder.ToString()  Else : Return s  End If  End Function |

1. レポート定義にテキストボックスを追加して式に数式を入力します。  
   

|  |
| --- |
| =Code.ChangeWord("TEST Bike da") |

なお、インテリセンスは効きません

1. この状態でレポートを表示すると次のようになります。



### グローバルキャッシュアセンブリを使用したカスタムコード

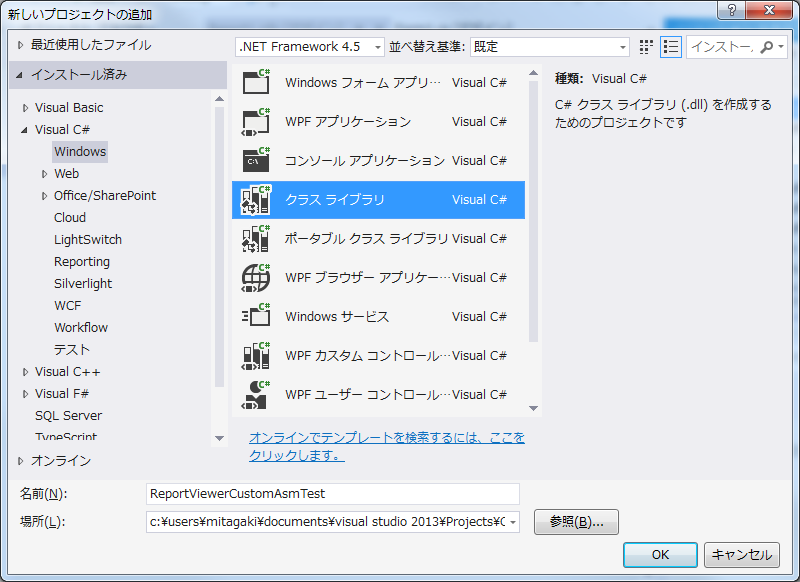
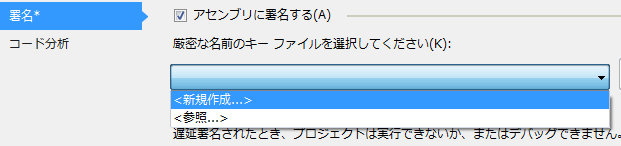
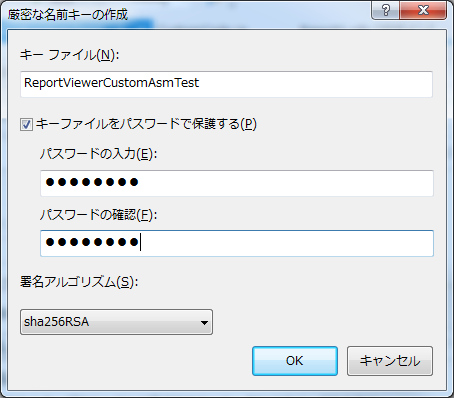
VB.NETなどでクラスモジュールを作成して、それをグローバルキャッシュアセンブリとして登録することで、ReportViewerはそのカスタムコードを利用できます。

.NET 4.0より前では以下のCustom Assembliesのやり方で作成できるはずです。<http://www.codeproject.com/Articles/38554/Microsoft-Reporting-Services-Part-II>

.NET 4.0以降において、このやり方は通用しません。

以降、.NET4.0以降でカスタムアセンブリを追加する方法を紹介します。

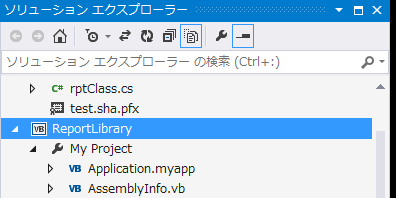
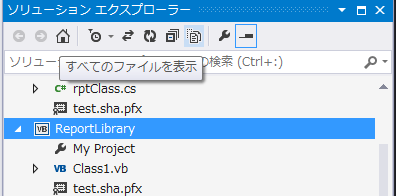
#### 厳密な名前を持つクラスライブラリを作成する

1. クラスライブラリを作成します。今回は「ReportViewerCustomAsmTest」という名称で作成します。  
   
2. 追加した「ReportViewerCustomAsmTest」のプロジェクトのメニューで「ReportViewerCustomAsmTestのプロパティ」を選択してください。
3. 署名タブにて「アセンブリに署名する」をチェックして「厳密な名前のキーファイルを選択してください」で＜新規作成＞を選択します。  
   
4. 「厳密な名前キーの作成」ダイアログが表示されるのでキーファイル名とパスワードを入力してOKを押してください。  
   
5. デフォルトで作成されたCustomCode.csを次のように置き換えます。

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ReportViewerCustomAsmTest  {  public class CustomCode  {  public string CutText(string txt)  {  if (txt.Length >= 30)  {  return txt.Substring(0, 26) + "...";  }  else  {  return txt;  }  }  public static String SharedCutText(string txt)  {  if(txt.Length >= 30 )  {  return txt.Substring(0,26) + "...";  }  else  {  return txt;  }  }  }  } |

1. AllowPartiallyTrustedCallers属性を指定します。  
   C#の場合は、AssemblyInfo.csに下記を追記してください。

|  |
| --- |
| [assembly: System.Security.AllowPartiallyTrustedCallers()] |

もしVB.NETで作成していた場合、AssemblyInfo.vbはデフォルトで表示されていません。表示するにはソリューションエクスプローラーで「すべてのファイルを表示」を押してください。  


これによりAssemblyInfo.vbが表示されるので、次を追記してください。

|  |
| --- |
| <Assembly: System.Security.AllowPartiallyTrustedCallers()> |

1. ビルドしてクラスライブラリを作成します。
2. 「開発者コマンドプロンプト for Visual Studio 2013」を管理者モードで起動します。  
   このツールはVisualStudioをインストールすると次のフォルダに作成されます。

|  |
| --- |
| C:\Program Files (x86)\Microsoft Visual Studio 12.0\Common7\Tools\Shortcuts |

1. コマンドプロンプト上で下記のコマンドを実行してグローバルキャッシュアセンブリとして登録します。

|  |
| --- |
| gacutil /i ReportViewerCustomAsmTest.dll |

ただしく実行されると次のようなメニューが表示されます。

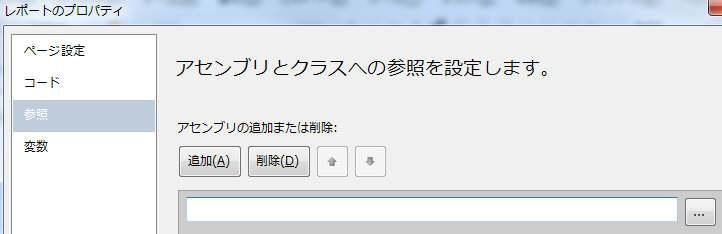
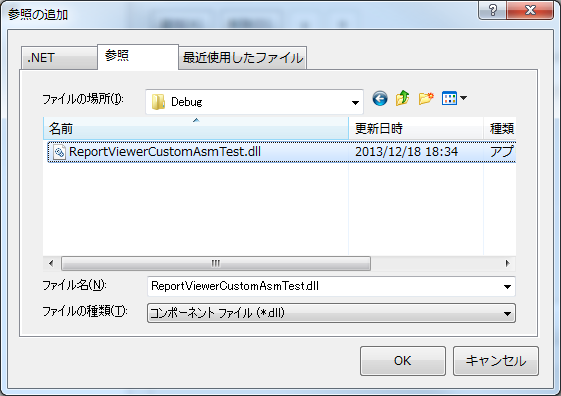
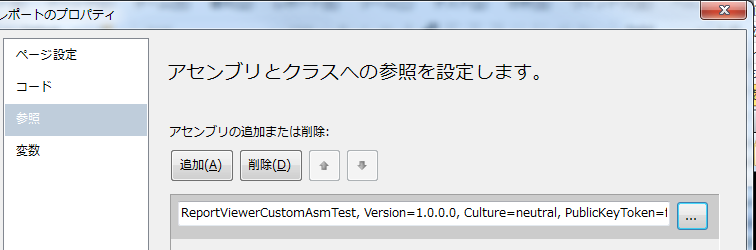
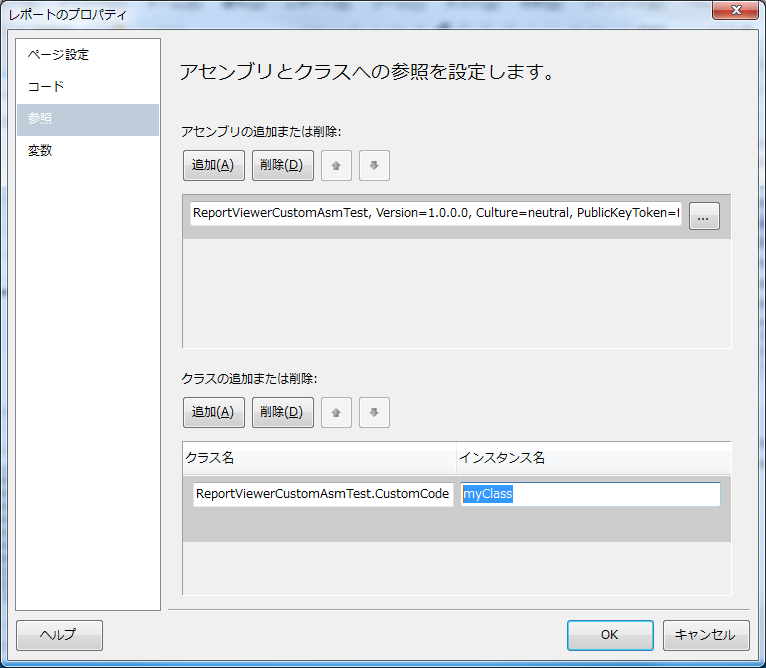
|  |
| --- |
| Microsoft (R) .NET Global Assembly Cache Utility. Version 4.0.30319.33440  Copyright (c) Microsoft Corporation. All rights reserved.  アセンブリが正しくキャッシュに追加されました |

**グローバルキャッシュアセンブリを削除するには？**

アセンブリ名（ファイル名ではない）を指定して/uオプションをつけてgacutilを実行してください。

|  |
| --- |
| gacutil /u ReportViewerCustomAsmTest |

#### グローバルキャッシュされたアセンブリをレポートで使用する

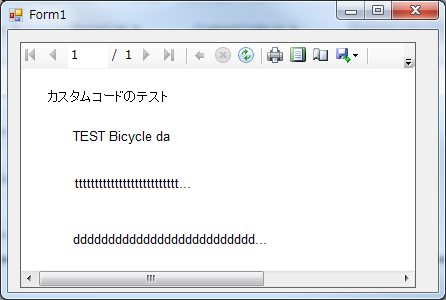
1. レポートデザイナを編集中に「レポート」メニューより「レポートのプロパティ」を選択します。
2. 「レポートのプロパティ」ダイアログの参照タブを選択して、アセンブリを追加します。  
     
   […]ボタンを押すと、「参照の追加」ダイアログが開くので作成したDLLを選択してOKボタンを押してください。  
     
   参照対象のアセンブリが追加されます。  
   
3. スタティックなメソッドを使用するだけならこれでいいのですが、もし、インスタンスを作成する必要がある場合は、クラス名の追加をする必要があります。  
   今回は以下のような内容を追加します。  
   クラス名：ReportViewerCustomAsmTest.CustomCode  
   インスタンス名：myClass  
   
4. レポート定義でテキストボックスに式を追加します。  
   スタティックなメソッドを利用する例：

|  |
| --- |
| =ReportViewerCustomAsmTest.CustomCode.SharedCutText("tttttttttttttttttttttttttttttttttttttttt") |

インスタンスメソッドを利用する例：

|  |
| --- |
| =Code.myClass.CutText("dddddddddddddddddddddddddddasdfasdfadfd") |

インスタンスを使う場合はプロパティで追加したインスタンス名を使用する必要があることがわかります。なお、やはり**インテリセンスはききません。**

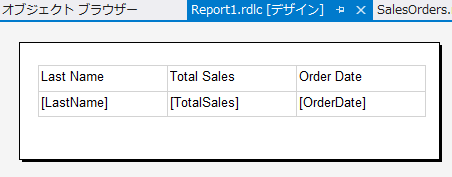
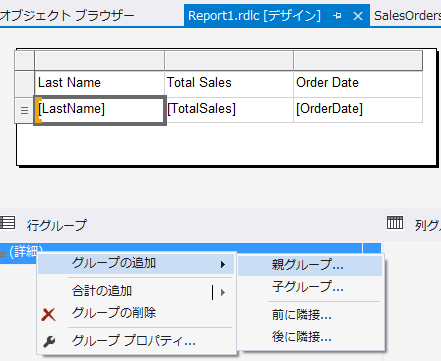
1. この状態で実行することで、カスタムコードが実行されていることが確認できます。  
   

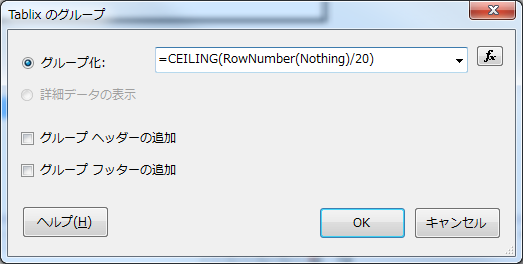
## 改ページの小技

ここでは改ページに関する小技について記述します。

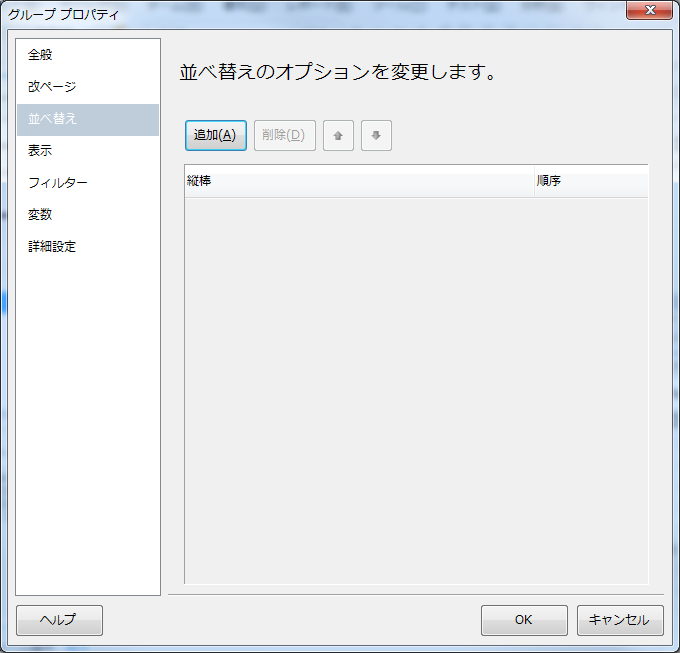
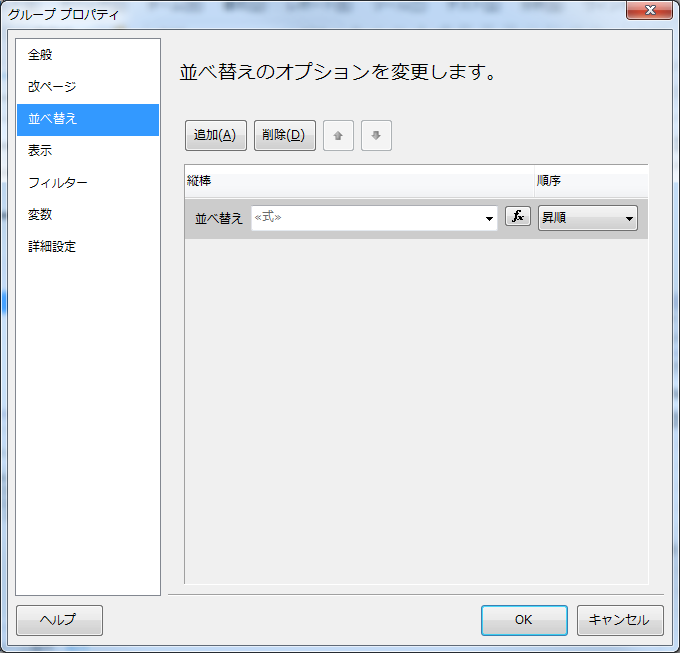
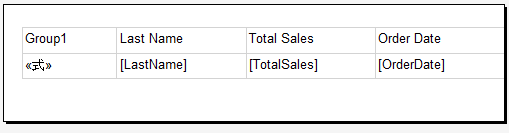
### 特定の行数毎ごとに改行をするには？

ここでは特定の行数毎に改行する方法を紹介します。

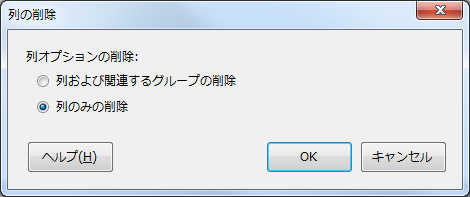
1. 「[チュートリアル：REPORTVIEWERレポートの作成](#_「データソース接続」と「データテーブル」を定義します)」と同じ手順でデータセットを作成します。
2. レポートデザイナでテーブルを追加して以下のような一覧を作成します。  
   
3. 行グループでグループを追加します。  
   

グループ化をする際の条件にRowNumberを用いて行数でグループ化するようにします。この例では20行おきでグループかします。  


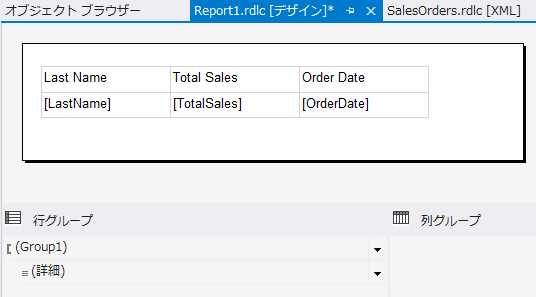
|  |
| --- |
| =CEILING(RowNumber(Nothing)/20) |

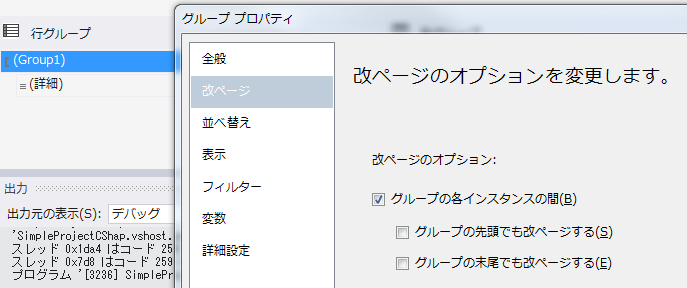
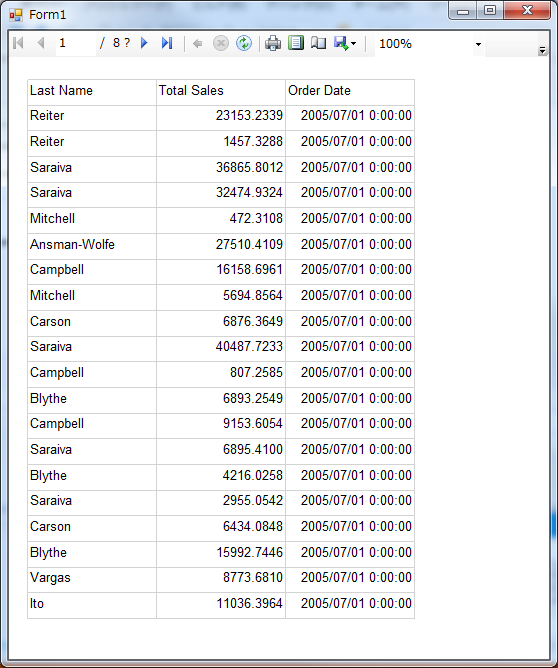
1. グループのプロパティで「並び替え」タブを選択して「式」を削除します。  
   グループ化の条件と並び替えの条件の式は先に入力したものと同じになります。しかし、並び替えにおいてRownumber数式は使用できないので、これを削除します。  
   
2. グループ用の列が表示されるのでグループを残して列のみを削除します。  
   

列を削除しようとすると「列の削除」ダイアログが表示されるので「列のみの削除」を選択します。



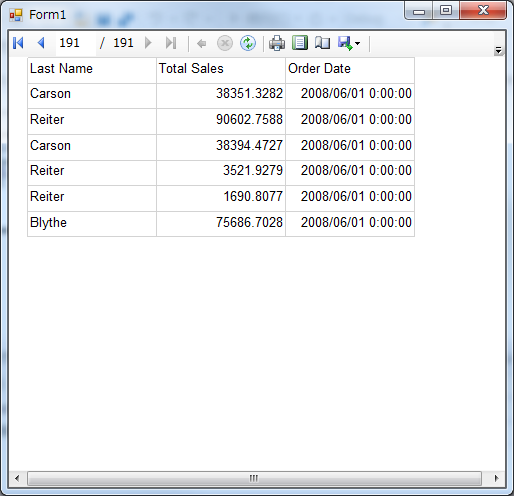
これにより、グループを残して列を削除できます。



1. グループごとに改ページを行うようにします。  
   グループプロパティーを開いたのち、改ページのタブで「グループの各インスタンスの間」にチェックを付与します。  
   
2. これにより、１ページあたり20行で改行されるようになります。  
   

### 一覧形式で最後のページにおいて最後まで罫線を引くにはどうしたらいいか？

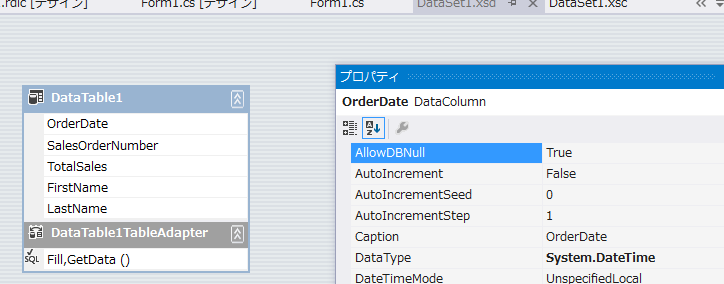
特定の行数毎に改ページができることは説明しました。この方法でも、最後のページに大きな余白が作成されます。ここでは最後のページにおいても最後まで罫線を引く方法を説明します。



Excelの場合は、Excelオブジェクト経由で罫線の描画ができるので、プログラムでいかようにも調整できます。

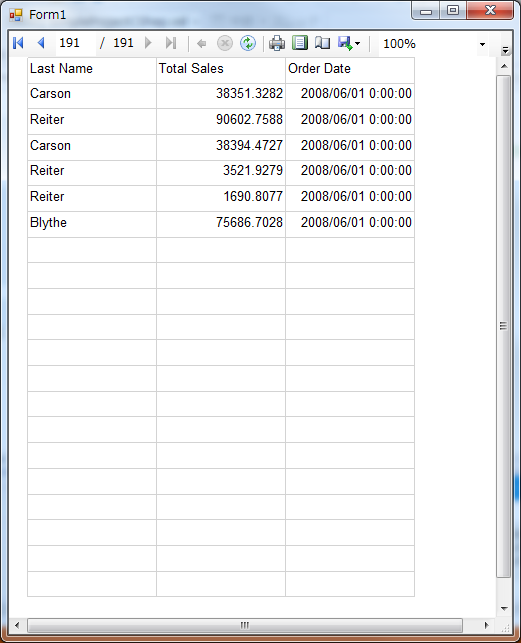
ReportViewerの場合はどうしたらいいのでしょうか？

レイアウトで何かをするのは無理です。これはページがちょうどに終わるように、DataSetに格納するレコードの数を調整するしかありません。

まず、DataSetのデザインで使用するすべてのフィールドのプロパティでAllowDbNullをTrueにします。  


あとは、ReportViewを表示するまえに、ダミーのデータを追加します。

|  |
| --- |
| private void Form1\_Load(object sender, EventArgs e)  {  // TODO: このコード行はデータを 'DataSet1.DataTable1' テーブルに読み込みます。必要に応じて移動、または削除をしてください。  this.DataTable1TableAdapter.Fill(this.DataSet1.DataTable1);  // 末尾にダミーデータ追加  int addcnt = 20 - (DataSet1.DataTable1.Count % 20);  for (int i = 0; i < addcnt; ++i)  {  DataRow r = DataSet1.DataTable1.NewRow();  DataSet1.DataTable1.Rows.Add(r);  }  this.reportViewer1.RefreshReport();  } |

これで実行することで最後まで罫線が引かれていることがわかります。  


# ExcelとReportViewerの比較

帳票の出力方法として、Excelオブジェクトを操作する方法があります。この章ではExcelを用いる方法とReportViewerを使用する方法で、どのような差があるか比較してみます。

## ReportViewerのメリット

ここではReportViewrを採用することで、Excelで帳票を作成する場合に比べてどのような利点が発生するかを説明します。

### ReportVIewerではユーザー環境にOffice製品が不要になる。

ReportViewerでは動作環境でOffice製品が不要になります。

Excelを使用する場合、動作環境にExcelが存在しなければなりません。

このことは動作する環境を選ぶことになります。特にパッケージソフトの場合、初めから、Office製品をもっていない層を販売対象から外すという選択が適切とは考えられません。

なるべく多くの環境で動作させる必要がある場合、ReportViewerの採用は大きなメリットがあります。

### ReportViewerでは単票形式の帳票を作成しやすい

Excelは表計算ソフトなので、当然、単票形式の帳票を作るのに向いていません。

シートをコピーするか、出力する際にレイアウトをプログラムで一々記述するか、１ページ出力するたびにExcelオブジェクトから作り直すか・・・いずれの方法も、開発の手間またはパフォーマンスに影響を与えます。

それにくらべて、チュートリアルを読んでいただけると、ReportViewerでは単票形式のデータを作成しやすいことがわかります。

## ReportViewerのデメリット

ここでは逆にReportViewrにどのようなデメリットがあるか説明します。

### Excelとまったく同じものは作れない

ReportViewrはExcelにエクスポートする機能も有していますが、Excelとまったく同じものは作成できません。エンドユーザーがすでにExcelを駆使して業務をおこなっているシステムの場合、そのことが不満につながる場合があります。

Excelで出力を提供できるなら、むりにそれを変更するのは実際使用するエンドユーザーのメリットにはならないでしょう。

### 認知度の低さ

ReportViewrはMicrosoftが提供しているコントロールですが、意外と使われていません。

おそらく、検索しても大した情報を得ることはできないでしょう。

いくら優れた機能を提供していても認知度が低いといくつかのリスクが発生します。

１つはトラブルシュートの事例の少なさです。Excelなどは大量にユーザーがいて様々な問題が報告され、それの対策がいくらでも取得できます。しかし、導入事例が少ないと、そのような情報を入手することが困難になります。

事実、この資料を作成するときに、発生したいくつかのトラブルの解決方法は日本語ではヒットすらしません。

次に開発要員の確保のしづらさです。Excelを触った人間はいくらでもいますが、ReportViewerを触ったことのある人間はそんなに多くありません。

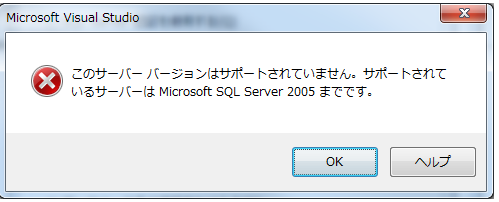
Excelで開発をおこなっている場合、いざとなったら、データを特定のシートにコピーするとこまでプログラマが開発して、実際のデザインなどはExcel使えるだけの人員にお願いするという開発体制がとれます。

ReportViewerの場合、その方法はとれません。どんな簡単なデザインの修正であれ、プログラミングの知識がないと無理でしょう。

# 付録

## VS2008からSQLSERVER2012に接続する

VisualStudio2008+SQLSERVER2012の場合、接続しようとしただけで下記のエラーが発生します。

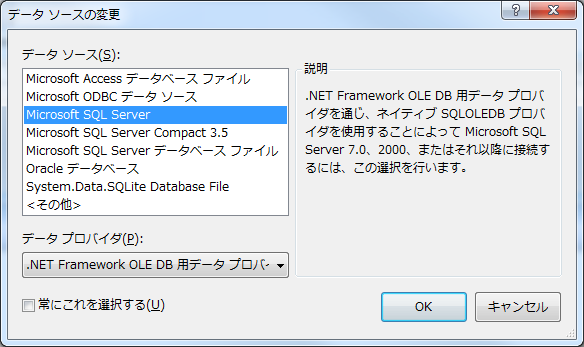


この場合は以下の点を確認してください。

・VisualStudio2008SP1が適用されており、かつ、すべての重要な更新プログラムが当たっていること。

・OLE DB用のデータプロバイダを使用する

OLEDB用のデータプロバイダはデータソースの変更で選択できます。

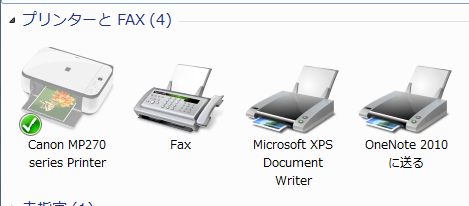


## 帳票のテストをどうやるか？

帳票のテストを行う場合、実際に紙に出力するのは、安定するまで待った方がいいでしょう。

不安定な状況でのテストは紙と時間の無駄になります。

幸いWindows Vista以降のOSにはXPS(XML Paper Specification)と呼ばれるドキュメントファイルがサポートされています。コントロールパネルのデバイスとプリンタを見ると、「Microsoft XPS Document Writer」と呼ばれるものがあります。



通常のプリンタに出力するかわりにXPS Document Writerに出力することにより、紙ではなくファイルとして保存できます。

動作が安定するまではこれを用いて動作確認をするといいでしょう。

また、ファイルに保存できるということは機械的に以前出力した結果と変わっていないか確認することができます。これ**は回帰テストを自動で行える**ということを意味します。

印字のたびにファイル名を効かれるのを抑制するには、下記のプロパティに任意のファイル名を指定するといいでしょう。

System.Drawing.Printing.PrinterSettings

ただし、単純にXPSを比較した場合、まったく同じ出力結果であっても、作成日や作成者によるメタデータによって違う結果とみなされます。これを防ぐにはXPSを一旦、画像に変換して比較を行うとよいでしょう。

[XpsToImg.zip](http://needtec.sakura.ne.jp/release/XpsToImg.zip)

http://needtec.sakura.ne.jp/release/XpsToImg.zip

任意のXPSファイルをPNGファイルに変換するツールです。

当然、数式などで出力日、作成日を表示すると異なる結果になるので、レポートパラメータなどを用いて、テスト用に固定の文字を指定できるようにします。

一つ注意しなければならないのは、XPSで出力できるからといって紙で出力する試験をやらなくていいわけではありません。  
出力するプリンタにより、思わぬ出力結果になることがあります。すべての帳票の改ページを含むデータで必ず１度は動作確認すべきです。