

# **FastReport.Net**

# **User's Manual**

(C) 2008- 2011 Fast Reports Inc.



# 目次

<b>章 1</b>	<b>基本事項</b>	<b>12</b>
レポート		12
レポートデザイナー		12
レポートオプション		13
レポートページ		16
ページの管理		17
ページ設定		18
バンド		21
デザイナー内のバンド		21
バンドの構成		23
バンドの出力		25
バンドのプロパティ		26
レポートオブジェクト		27
オブジェクトの共通プロパティ		28
テキストオブジェクト		29
テキストの編集		30
式の表示		30
データ列の表示		32
HTML タグ		33
オブジェクトのプロパティ		34
リッチ テキストオブジェクト		35
図オブジェクト		37
線オブジェクト		40
図形オブジェクト		41
バーコードオブジェクト		42
チェックボックス オブジェクト		45
表オブジェクト		46
行列オブジェクト		47
MS Chart オブジェクト		47
郵便番号オブジェクト		48
セル形式テキストオブジェクト		49
FastReport でレポートを作成する		51
例 1 :レポートの手動作成		51
例 2 :ウィザードを使ったレポートの作成		54
<b>章 2</b>	<b>レポートの作成</b>	<b>60</b>
レポート用データの選択		61
動的なレイアウト		62
CanGrow, CanShrink プロパティ		62
Shift Mode プロパティ		63
GrowToBottom プロパティ		63
Anchor プロパティ		64
Dock プロパティ		64
書式設定		66

罫線と塗りつぶし	66
テキストの書式設定	67
スタイル	67
データの書式設定	69
条件付き強調表示	70
ゼロの値を非表示にする	74
重複する値を非表示にする	74
データの偶数行を強調表示する	75
1つのデータバンドを持つレポート	77
バンドをデータソースへ接続する	77
テキストの出力	77
データの並べ替え	78
データのファイルタリング	79
データのヘッダーおよびフッター	80
データのブレークおよび同一ページに出力	82
空のデータ行の出力	84
"データなし" テキストの出力	86
階層の出力	87
マスター/ 詳細レポート	90
マスター/ マスター レポート	94
マスター/ 詳細/ 詳細レポート	96
複数列のレポート	97
ページの段組み	97
データバンドの列	98
小冊子タイプのレポート	100
レポートへのページの追加	100
ページ設定	100
奇数/ 偶数ページ上の印刷	101
グループ化と集計	103
グループの作成	105
データの並べ替え	107
入れ子のグループ	108
グループの管理	109
集計値の出力	110
ヘッダーとフッターの繰り返し	112
グループのプロパティ	114
サブレポート	115
出力モード	115
横並び表示のサブレポート	116
入れ子のサブレポート	116
表タイプのレポート	117
列の設定	117
列のサイズの管理	117
行の設定	118
行のサイズの管理	118
セルの設定	119
セルの結合および分割	119
セルにオブジェクトを挿入する	120
表の出力	120
複雑なヘッダーの出力	123

集計の使用	125
表のレイアウト	125
例	126
例 1 :表全体を上から下へ出力する	126
例 2 :繰り返し行を含む表を上から下へ出力する	127
例 3 :表全体を左から右へ出力する	128
例 4 :繰り返し列を含む表を左から右へ出力する	128
例 5 :繰り返し行/列を含む表を出力する	129
例 6 :データソースの使用	130
例 7 :改ページの挿入	131
例 8 :集計の出力	132
行列タイプのレポート	134
行列の概要	134
行列の設定	136
ヘッダーの設定	137
セルの設定	137
行列のスタイル	138
行と列のサイズ管理	139
例	140
例 1 :シンプルな行列	141
例 2 :複数レベルのヘッダー	143
例 3 :月名の出力	144
例 4 :条件付き強調表示	144
例 5 :偶数行の強調表示	146
例 6 :式の使用	147
例 7 :セル内の図	149
例 8 :セル内のオブジェクト	150
例 9 :行列データの手動埋め込み	152
対話型レポート	156
ハイパー・リンク	156
ハイパー・リンクの設定	156
URL へのリンク	157
ページ番号へのリンク	158
ブックマークへのリンク	158
詳細レポートへのリンク	158
詳細ページへのリンク	159
カスタム リンク	161
レポートのアウトライン	161
例	163
例 1 :Web ページへのリンク	163
例 2 :詳細レポートの作成	164
例 3 :対話型の行列オブジェクト	169
例 4 :目次、ナビゲーションおよびアウトライン付きのレポート	173
レポートの継承	176
レポートの作成	176
基本レポートの変更	177
制限事項	178
グラフ付きのレポート	179
グラフの要素	179
グラフ エディター	181

系列の処理	182
外観の設定	183
グラフをデータへ接続する	184
データの並べ替え	185
データのグループ化	186
データの収集	187
値の分割	189
系列の自動作成	189
対話型グラフ	191
<b>章 3 データ</b>	<b>196</b>
[データ] ウィンドウ	196
データソース	197
データソースの作成	198
SQL クエリの作成	201
クエリパラメーター	204
パラメーターへ値を渡す	205
接続の編集	206
データソースの編集	206
エイリアス	206
階層データソース	207
関係	207
関係の作成	211
関係の編集	212
システム変数	212
関数	213
数学	213
Abs	214
Acos	214
Asin	214
Atan	215
Ceiling	215
Cos	215
Exp	216
Floor	216
Log	216
Maximum	217
Minimum	217
Round	218
Sin	218
Sqrt	219
Tan	219
Truncate	219
文字列	220
Asc	220
Chr	220
Insert	220
Length	221
LowerCase	221
PadLeft	221

PadRight	222
Remove	222
Replace	223
Substring	223
TitleCase	224
Trim	224
UpperCase	225
日付と時刻	225
AddDays	225
AddHours	225
AddMinutes	226
AddMonths	226
AddSeconds	226
AddYears	226
DateDiff	227
DateSerial	227
Day	227
DayOfWeek	228
DayOfYear	228
DaysInMonth	228
Hour	229
Minute	229
Month	229
MonthName	229
Second	230
Year	230
書式	230
Format	230
FormatCurrency	235
FormatDateTime	236
FormatNumber	236
FormatPercent	237
変換	238
.ToBoolean	238
ToByte	238
ToChar	238
ToDateTime	239
ToDecimal	239
ToDouble	239
ToInt32	239
ToRoman	240
ToSingle	240
ToString	240
ToWords	241
ToWordsEnGb	242
ToWordsRu	243
プログラム フロー	244
Choose	244
IIf	244
Switch	245

集計	245
集計の作成	246
条件付き集計	248
累計	249
ページ集計	249
ヘッダーに集計を出力する	250
レポートパラメーター	251
パラメーターの作成	252
レポート内でのパラメーターの使用	252
<b>章 4 式</b>	<b>256</b>
式エディター	256
レポートオブジェクトの参照	257
.NET 関数の使用	258
データ要素の参照	259
データソースの参照	260
システム変数の参照	261
集計値の参照	262
レポートパラメーターの参照	262
<b>章 5 スクリプト</b>	<b>264</b>
全般情報	266
イベントハンドラー	267
レポートのイベント	268
.NET オブジェクトの使用	269
レポートオブジェクトの参照	270
Report および Engine オブジェクト	270
データソースの参照	274
システム変数の参照	275
集計値の参照	275
レポートパラメーターの参照	276
例	277
例 1 :オブジェクトの外観を変更する	277
例 2 :バンドの偶数行を強調表示する	278
例 3 :データをフィルター処理する	279
例 4 :合計を計算する	280
例 5 :出力位置を移動する	281
<b>章 6 ダイアログ フォーム</b>	<b>284</b>
コントロール	284
コードからのコントロールの参照	286
データのフィルター処理	286
自動フィルタリング - 動作方法	287
フィルター演算	288
レポートへのフィルターの追加	289
データ範囲によるフィルタリング	290
関連付けられたデータ列によるフィルタリング	290
カスクードリストを使用したフィルタリング	291
コードからのフィルタリング制御	291
例	292

例 1 :"Hello, Fast Report!"	292
例 2 :ユーザーにテキスト入力を求める	293
例 3 :ダイアログ コントロールを処理する	293
例 4 :レポートオブジェクトを処理する	294
例 5 :シンプルなフィルター	295
例 6 :自動フィルタリング	297
例 7 :範囲による自動フィルタリング	298
例 8 :関連付けられたデータ列によるフィルタリング	299
<b>章 7 プレビュー、印刷、エクスポート</b>	<b>304</b>
レポートの編集	306
レポートの印刷	307
レポートのエクスポート	308
FPX 形式で保存	309
Adobe Acrobat (PDF)へのエクスポート	309
Word (RTF)へのエクスポート	311
HTML へのエクスポート	312
MHT (Web アーカイブ)へのエクスポート	313
Excel (XML)へのエクスポート	313
Excel 2007 へのエクスポート	314
PowerPoint 2007 へのエクスポート	315
OpenOffice Calc へのエクスポート	316
CSV へのエクスポート	317
TXT へのエクスポート	318
イメージへのエクスポート	319
レポートデザインにおける推奨事項	321
レポートを電子メールで送信する	322



# 第1章

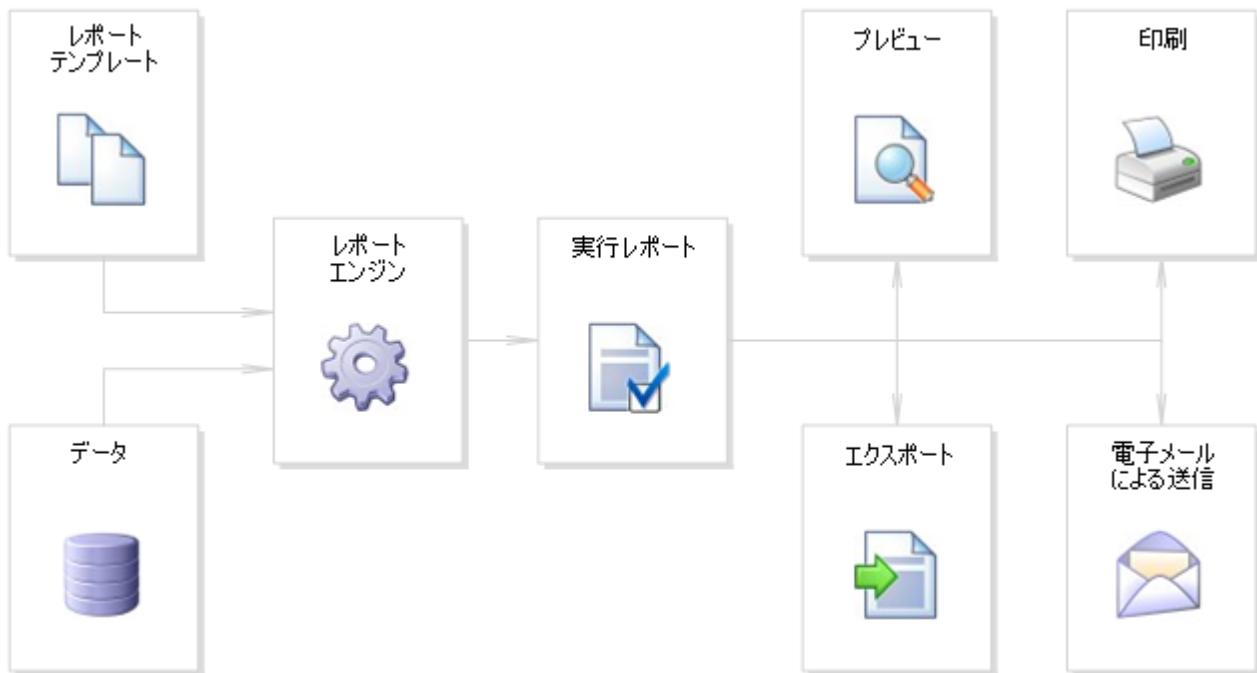
## 基本事項

# 基本事項

この章では, FastReport におけるレポートの基本的な作業について説明します。また, レポートページ, バンド, およびレポートオブジェクトなどのレポート要素についても詳しく説明しています。

## レポート

レポートの作成プロセスは次のように表すことができます。



レポートテンプレート(以下「レポート」と言います)- これはデザイナーに表示されるレポートです。.FRX という拡張子を付けたファイルに保存されます。このレポートはデザイナーまたはプログラムを利用して作成することができます。

データ- これはプログラムで定義されたデータ または MS SQL などのDBMS のデータを含め、あらゆるデータに対応しています。FastReport ではビジネスロジックオブジェクト(以下「ビジネスオブジェクト」と言います)を使って作業することもできます。

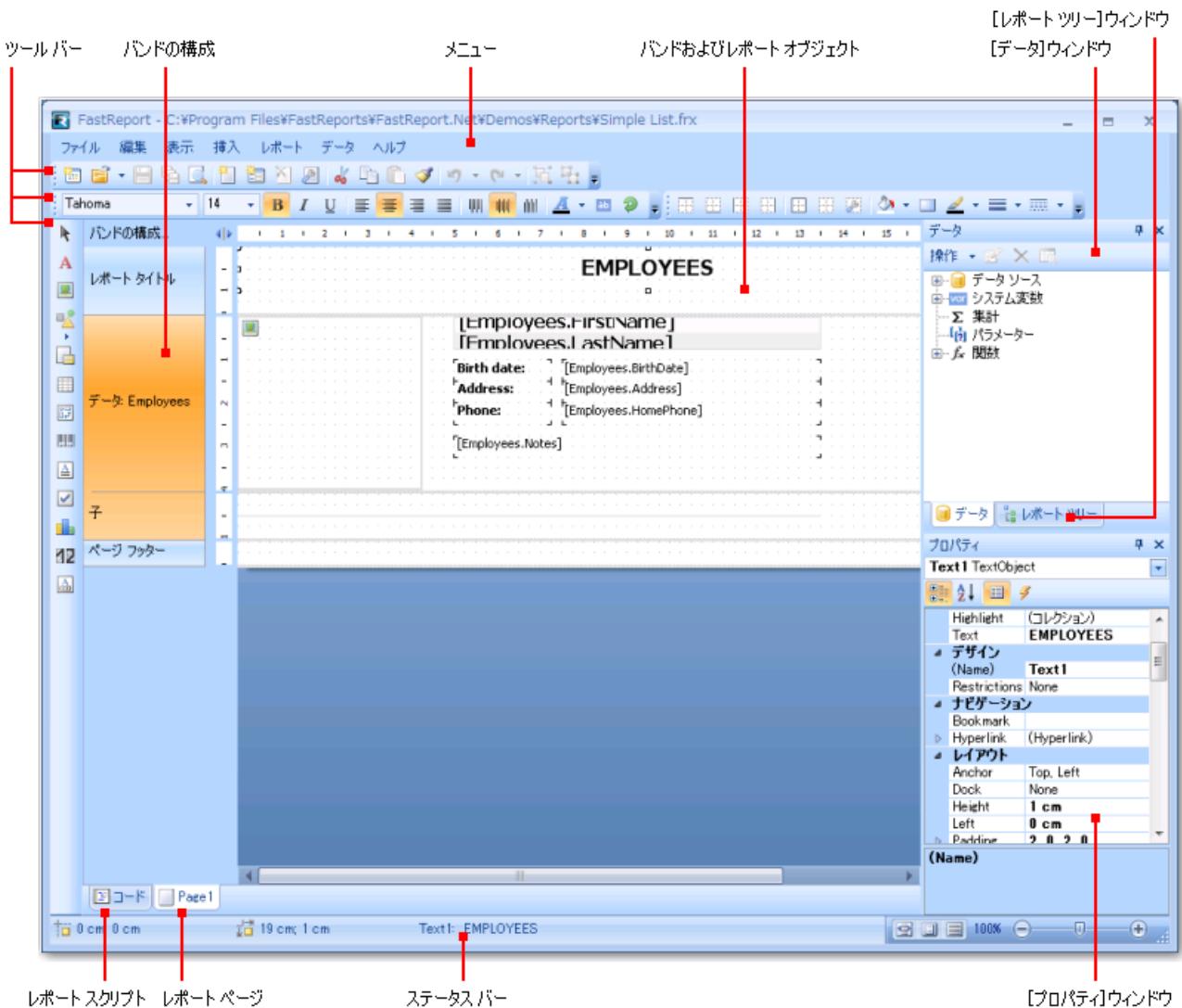
実行レポート- これはプレビュー ウィンドウに表示されるレポートです。実行レポートはプレビュー、印刷、サポートされる形式 (.doc, .xls, .pdf など)で保存、または電子メールで送信することができます。

## レポートデザイナー

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を使用します。レポートデザイナーを使用すれば、ユーザーは簡単な操作でレポート設計し、さらにそのレポートをプレビューすることができます。

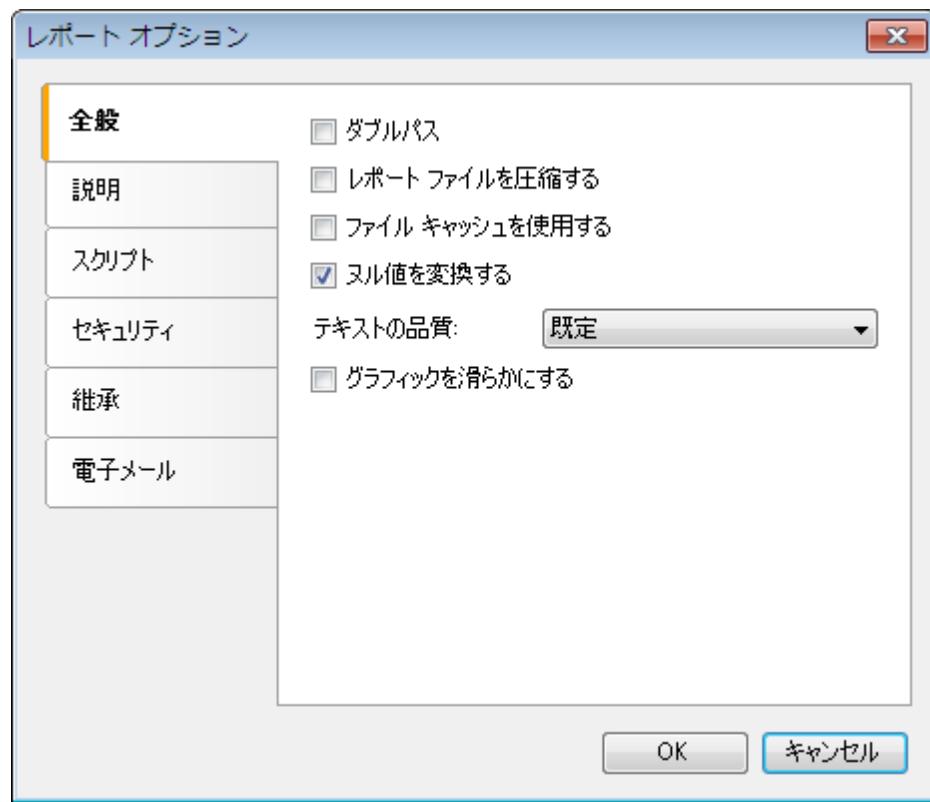
レポートデザイナーは FastReport に含まれる機能であり、開発環境 (MS Visual Studio など)には依存しません。ソフトウェア開発者

は、作成したアプリケーションにこのレポートデザイナーをインクリードすることができます。これにより、エンドユーザーが既存のレポートを変更したり、新しいレポートを作成したりすることができます。



## レポートオプション

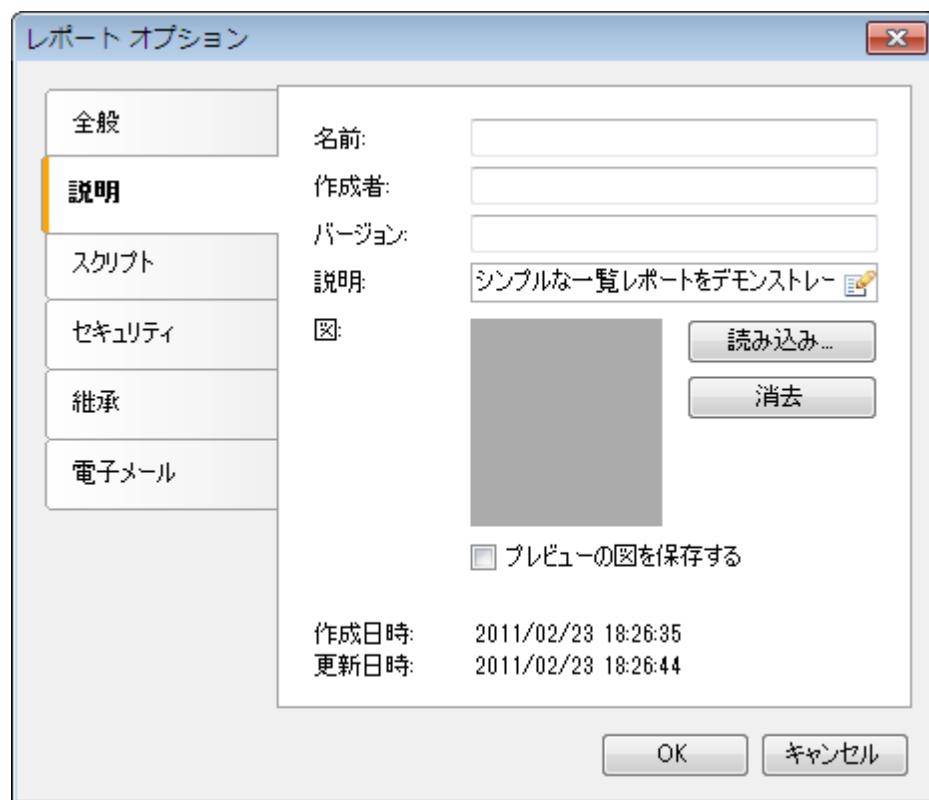
レポートオプションに関するウインドウは、[レポート]メニューの[オプション]を選択して呼び出すことができます。いくつかのタブを持つダイアログウインドウが表示されます。



[全般]タブでは、以下のレポートパラメーターを制御できます。

- [ダブルパス]パラメーターは2回のレポートパスを可能にします。これは、総ページ数に関するシステム変数を使用する際に必要です。
- [レポートファイルを圧縮する]パラメーターを使用すると、圧縮形式でレポートを保存することができます。圧縮にはzipアルゴリズムが使用されるので、そのアーカイブを使って元の内容を簡単に抽出することができます。
- [ファイルキャッシュを使用する]パラメーターを使用すると、レポートの作成時にメモリを節約することができます。レポートに大量のページが含まれる場合はこのパラメーターを使用します。
- [ヌル値を変換する]パラメーターを使用すると、ヌル値データ列を既定値へ変換することができます（既定値は列のデータ型に応じて0、空文字列、Falseのいずれか）。
- [テキストの品質]パラメーターでは、レポートに表示されるテキストのモードを選択できます。このモードはレポートの印刷には影響しません。
- [グラフィックを滑らかにする]パラメーターを使用すると、グラフィックオブジェクト（線、罫線、図）を描画する場合にスムーズモードを有効にすることができます。

[説明]タブでは、レポートに関する説明を入力することができます。これらのパラメーターの指定は必須ではありませんが、レポート情報として役立ちます。



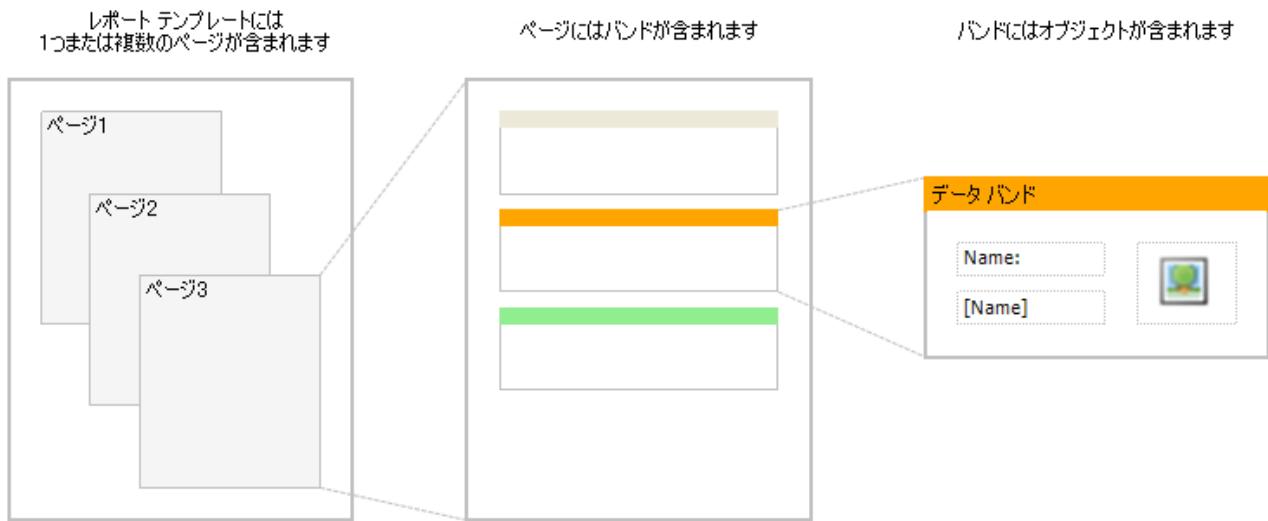
[スクリプト]タブでは、レポートのスクリプト言語を選択できます。スクリプトを使った作業の詳細については、[スクリプト](#)の章を参照してください。

[セキュリティ]タブでは、レポートを開くときに要求するパスワードを指定できます。パスワードを持つレポートはエンコードされた形式で保存されるので、パスワードを忘れないようにしてください。パスワードを忘れるときの復元は実質的に不可能です。

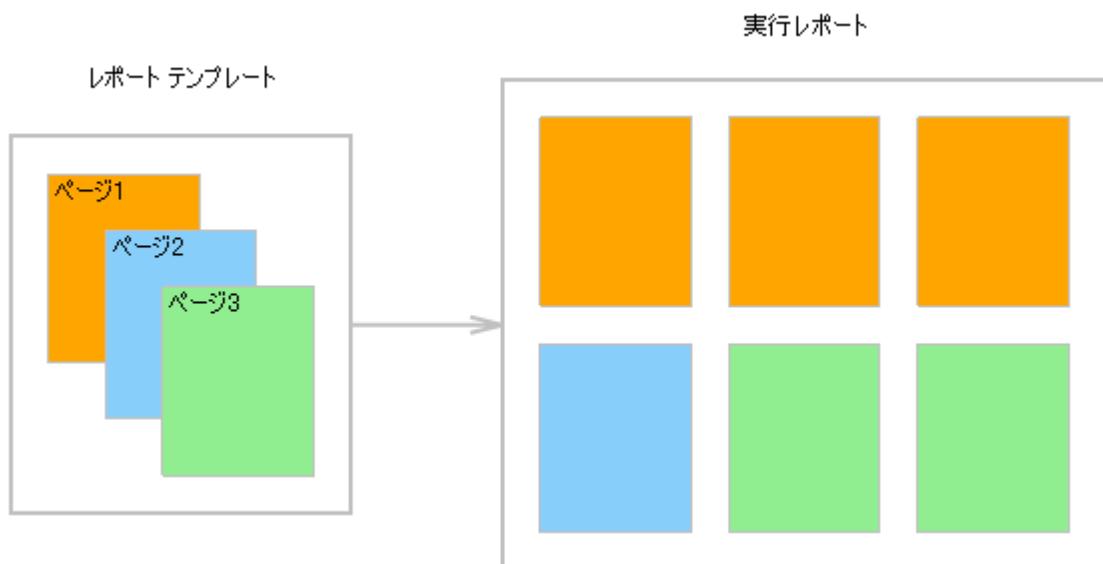
[継承]タブでは、レポートの継承を制御できます。この機能については、後程説明します。

## レポートページ

レポートテンプレートは(主に)1つまたは複数のレポートページで構成されます。レポートページには複数のバンドが含まれます。テキスト、図などのレポートオブジェクトはバンド上に配置されます。



レポートテンプレートは複数のページで構成することができます。たとえば タイトルページとデータページを含む1つのテンプレートを作成できます。そのようなレポートを作成する場合、最初のページが最初に出力され、次に2番目のページが出力されます。テンプレートの各ページがそれぞれ1ページまたは複数ページの実行レポートを生成できます。生成されるページ数はレポートに含めるデータに応じて異なります。



レポートページはサブレポートの作業時にも使用されます。ほかのレポートジェネレーターとは異なり FastReport のサブレポートは別のファイルではなく別のテンプレートページに保存されます。

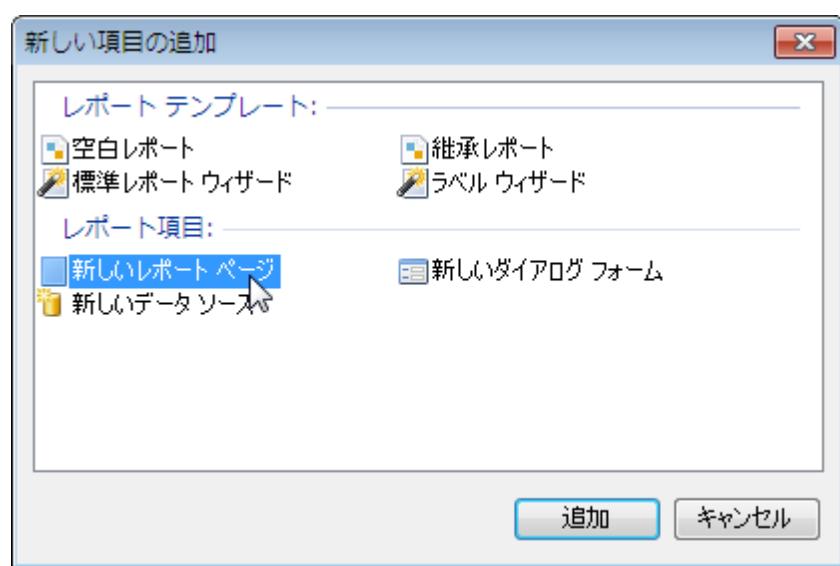
テンプレートには、レポートページ以外に1つまたは複数のダイアログフォームを含めることもできます。ダイアログフォームを使用すると、レポ

ートを作成する前にいくつかのパラメーターを照会することができます。ダイアログフォームを使った作業については、[ダイアログフォーム](#)の章（該当セクション）を参照してください。

## ページの管理

新規作成したレポートには、いくつかのバンドを含む1ページが既に存在します。新しいページを追加するには、 ボタンをクリックします。

または、 ボタンをクリックし、[新しい項目の追加] ウィンドウで [新しいレポートページ] を選択してページを追加することもできます。



ダイアログフォームもレポートに追加できます。これには、 ボタンを使用します。

テンプレートページはタブとしてデザイナーに表示されます。



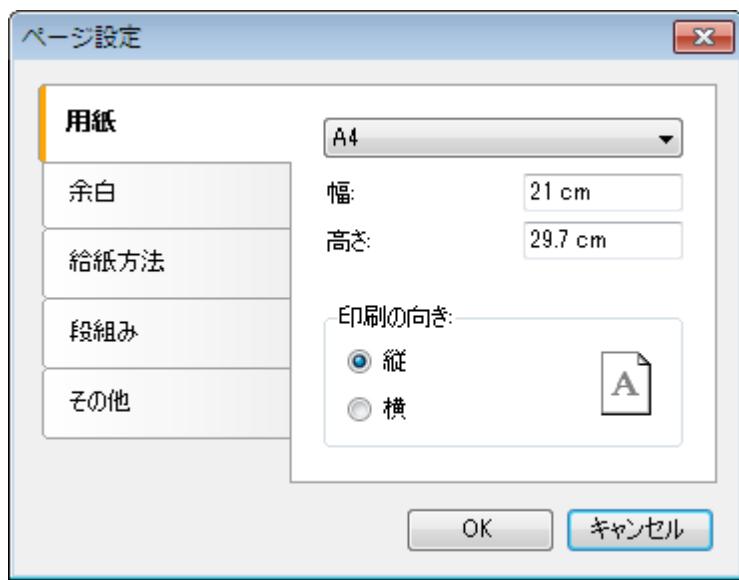
1番目(左端)のタブはレポートコードです。このタブは移動も削除もできません。

目的のページを表示するには、そのタブをクリックします。ページの順序は、マウスを使って変更することができます。移動対象のタブ上でマウスの左ボタンを押し、そのまま移動先の位置までマウスを移動させます。

ページを削除するには  ボタンをクリックします。レポートが1ページのみの構成であった場合、このボタンはアクティブではありません。

## ページ設定

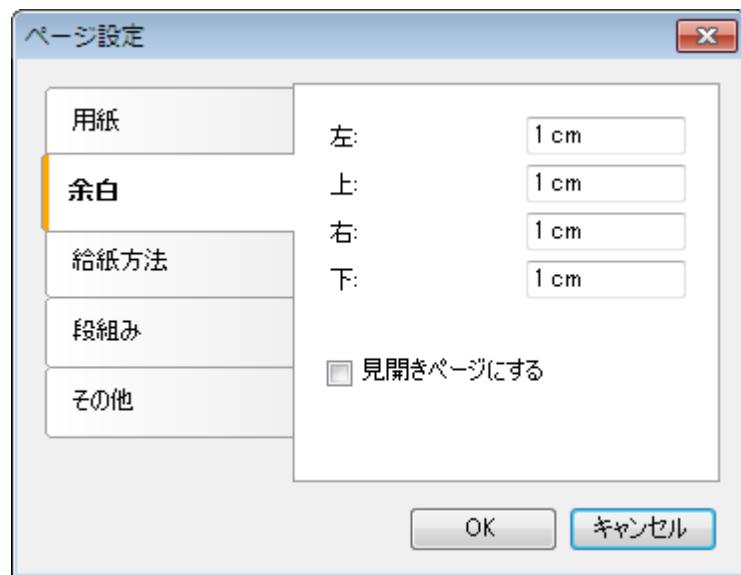
レポートページごとに、用紙サイズ、印刷の向き（縦または横）、余白、段組み、給紙方法などの設定を個別に持つことができます。レポートテンプレートには、印刷の向きや用紙サイズが異なる複数のページを含めることができます。ページ設定のウインドウは  ボタンをクリックするか、[ファイル]メニューの [ページ設定] を選択して呼び出すことができます。



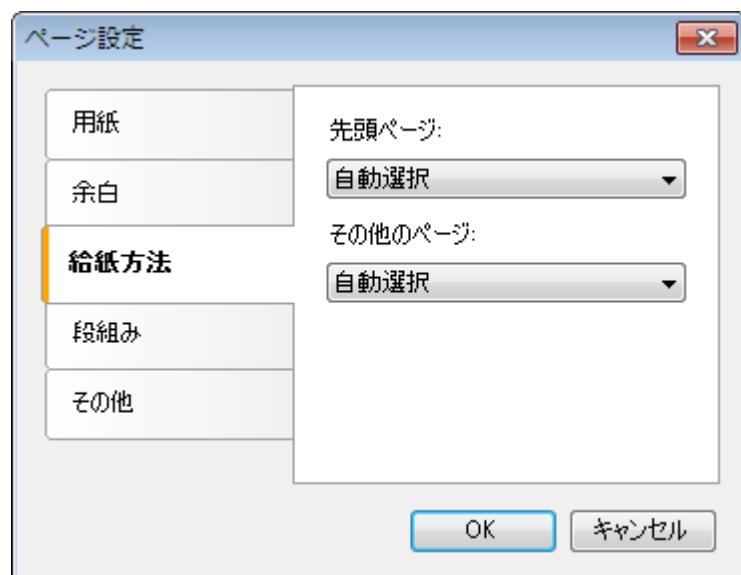
[用紙]タブでは、用紙サイズと印刷の向きを設定できます。用紙のサイズは、ドロップダウンリストのサイズの一覧から選択できます。この一覧には、現在使用中のプリンターでサポートされているすべての用紙サイズが含まれます。

現在使用中のプリンターは [ファイル]メニューの [プリンターの設定] を選択して設定できます。

[余白]タブでは、ページの余白を設定できます。[見開きページにする]オプションは小冊子を出力するときに利用できます。



[給紙方法]タブでは、用紙の給紙方法を選択できます。給紙方法は、実行レポートの先頭ページとその他のページで個別に指定できます。

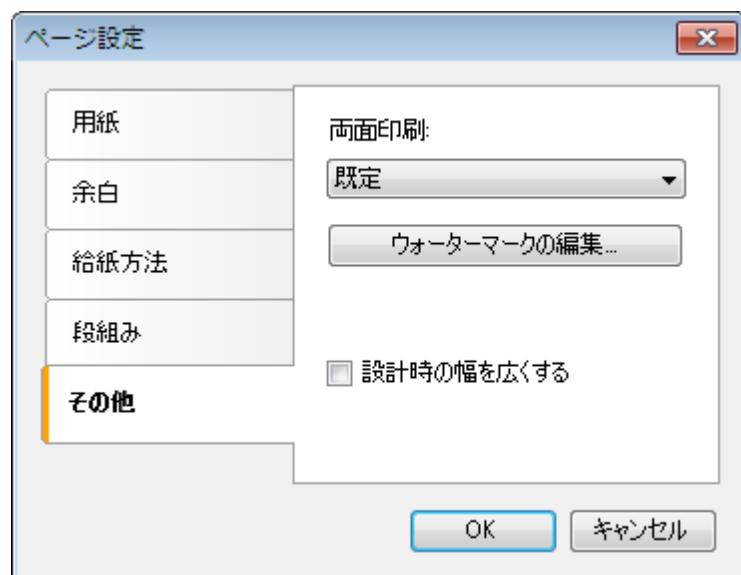


給紙方法は [印刷] ダイアログでも選択できます。

[段組み]タブでは、複数列のレポート向けに列パラメーターを設定することができます。このためには、段数を指定し、(任意で)段の幅と各段の位置を調整します。



[その他]タブでは、いくつかの有用なページオプションを設定できます。プリンターが両面印刷に対応している場合には、両面印刷モードを指定することができます。ウォーターマークも設定できます。これは実行したレポートページ上に出力されます。



[設計時の幅を広くする] チェックボックスを使用すると、デザインモードでページ幅を広くすることができます。このオプションは、表または行列のようなオブジェクトを扱う場合に役立つ可能性があります。

両面印刷は [印刷] ダイアログでも選択することができます。

## バンド

バンドとは、レポートページ上に直接配置されているオブジェクトで、その他のオブジェクト(テキスト、図など)用のコンテナでもあります。

FastReport には全部で 13 種類のバンドがあります。バンドはその種類に応じてレポート内の特定の場所に出力されます。

バンド	出力方法
レポートタイトル	レポートの冒頭に1回だけ出力されます。TitleBeforeHeader プロパティを使用すると出力の順序(ページヘッダー バンドの前または後)を選択することができます。このプロパティの変更は [プロパティ] ウィンドウで行うことができます。このプロパティの既定値は True で、レポートタイトルはページヘッダーの前に出力されます。
レポートの概要	レポートの最後で、最終データ行の後、かつページ フッター バンドの前に1回だけ出力されます。
ページ ヘッダー	レポートの各ページの最上部に出力されます。
ページ フッター	レポートの各ページの最下部に出力されます。
列 ヘッダー	このバンドは、複数列のレポート([ページ設定]の[段組み])で指定されている段数が2以上の場合)を出力する場合に使用します。ページ ヘッダー バンドの後で、各列の上部に出力されます。
列 フッター	ページ フッター バンドの前で、各列の下部に出力されます。
データ	このバンドは、データソースは接続され、そのデータソースに存在するデータ行がすべて出力されます。
データ ヘッダー	このバンドは、データバンドは接続され、データの先頭行の前に出力されます。
データ フッター	このバンドは、データバンドは接続され、データの最終行の後に出力されます。
グループ ヘッダー	グループ条件の値が変わったときに、各グループの先頭に出力されます。
グループ フッター	各グループの最後に出力されます。
子	このバンドは、どのバンドにも接続することができます。別の子バンドにも接続可能です。親となるバンドの直下に出力されます。
オーバーレイ	各レポートページ上で背景として出力されます。

## デザイナー内のバンド

デザイナーで、バンドは四角形の領域として現れます。バンドには、多くのレポートオブジェクトと同様に墨線や塗りつぶしの設定があります(既定では、どちらも設定されていません)。また、バンドにはグリッドが表示されます。グリッドモードを設定するには、[表示] メニューの [オプション] を選択し、[オプション] ダイアログで [レポートページ] を選択します。グリッドの表示/非表示の切り替えは、[表示] メニュー ([グリッ

ド])で直接行うともできます。

バンドの高さは以下の3つの方法で設定できます。

- マウスポインターをバンドの下端に置きます。カーソルの形状が上下分割に変わればバンドの高さを変更できます。
- 左側のルーラー上でバンドハンドルをドラッグします。
- [プロパティ]ウインドウで、バンドのHeightプロパティを設定します。

デザイナーには、バンドの表示に2つのモードがあります。モード間の切り替えは1つでも行えます。一方の表示モードでは、各バンドにヘッダーがあり、そのヘッダーにはバンドのタイトルやそのバンドに関する有用な情報(接続されるデータソースの名前など)が含まれます。



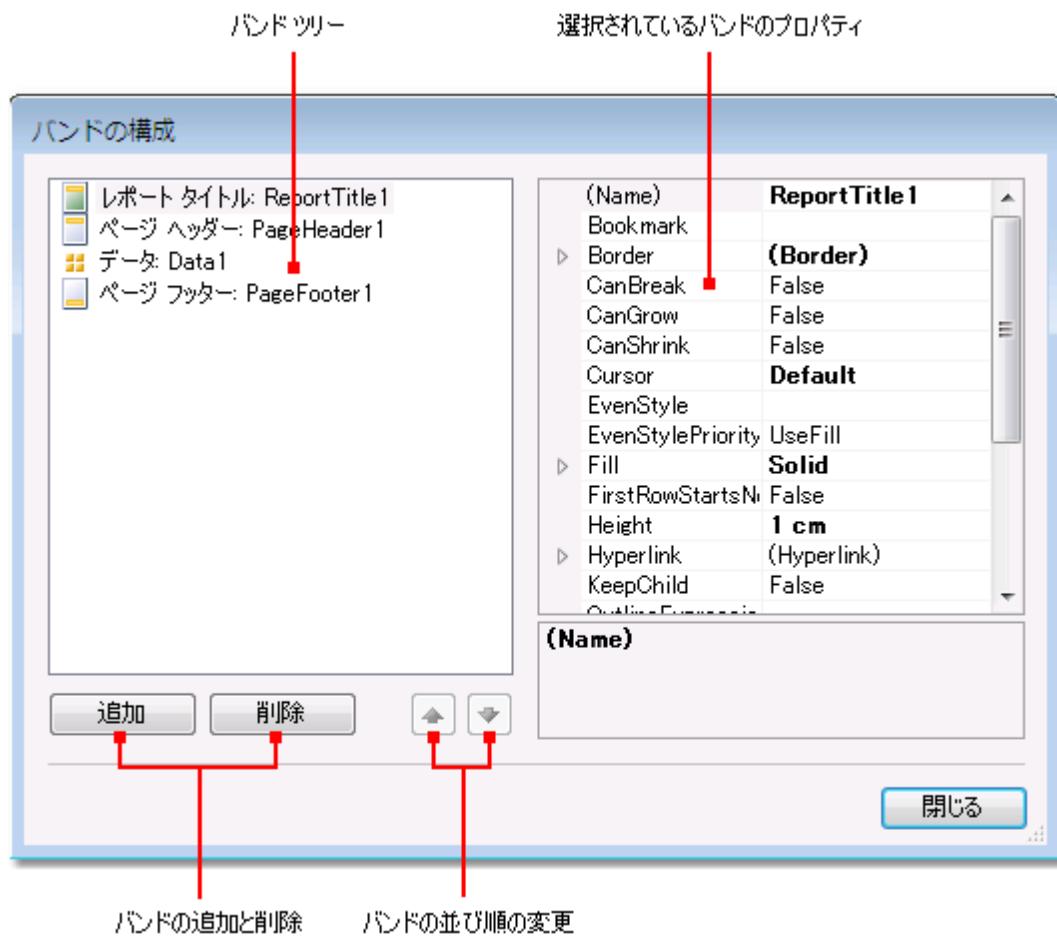
もう一方の表示モードでは、バンドにヘッダーがありません。その代わり、各バンドの左側にバンドの構造が表示されます。特にそのレポートを自分で作成しない場合には、この表示モードにすることでレポートの構造がわかりやすくなります。



これらの表示モードは ボタンで切り替えることができます。

## バンドの構成

バンドは [バンドの構成] ウィンドウで構成することができます。このウィンドウは、[レポート]メニューから [バンドの構成] を選択するか、またはバンド構成ツリー上 (バンドの左側上部) にある [バンドの構成] をクリックして呼び出すことができます。

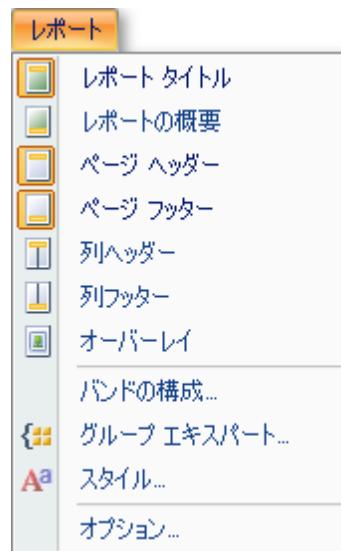


このウィンドウではレポートのバンドについて、追加、削除または出力順序の変更が行えます。バンドを追加するには [追加] ボタンをクリックするか、バンド構造ツリー上を右クリックします。コンテキストメニューには次のようにバンドの一覧が表示されます。追加できないバンドは淡色表示され選択できません。



追加操作はバンド構造ツリー内で選択したバンドに応じて異なります。たとえば、データ用のヘッダー バンドおよびフッター バンドは、事前にデータバンドが選択されていた場合にのみ追加することができます。

いくつかのバンドについては別 の方法で構成することができます。これは [レポート] メニューから行います。

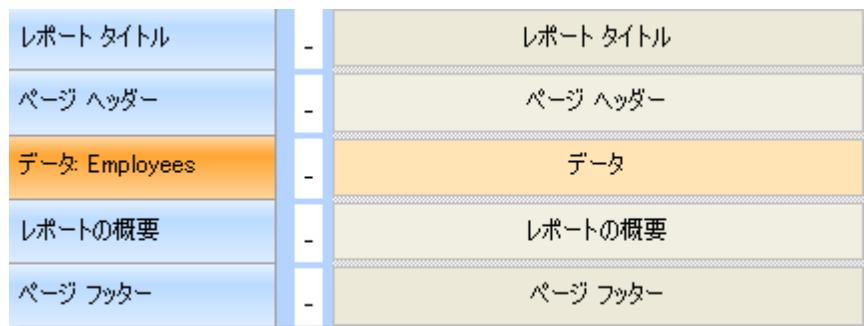


バンドを削除するには、そのバンドを選択して Delete キーを押します。

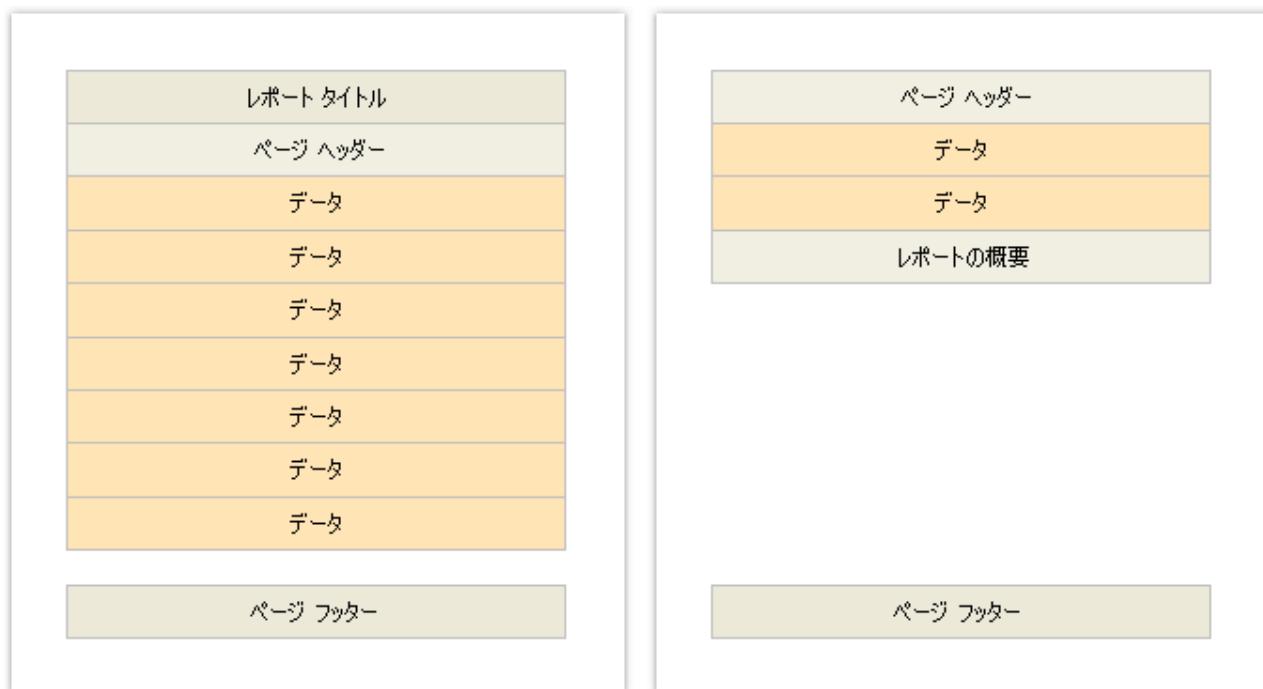
バンドを構成する場合、FastReport では不正なレポートテンプレートを作成するような操作を許可しません。たとえば、グループに接続されているデータバンドは削除できません。このデータバンドを削除するには、先にグループを削除してお必要があります。このほか、データバンドを削除すると、そのヘッダーとフッターは自動的に削除されます。また、ページ上にバンドが 1 つしかない場合は、そのバンドを削除することができません。

## バンドの出力

ページ上には複数のバンドが置かれます。FastReport はどのようにして実行レポートを構成するのでしょうか？次の例を見てみましょう



最初にレポートタイトルバンドが出力されます。その下にページヘッダーバンドが出力されます。さらにその下にはデータバンドが出力されます。このバンドは接続されているデータソースに存在するデータ行がすべて出力されます。データバンドの行がすべて出力された後に、レポートの概要バンドが出力され、ページの最後にはページフッターバンドが出力されます。これでレポートの出力は終了です。実行されたレポートは次のように表示されます。



出力の過程で、FastReport は実行レポートの現在のページ内にバンドを出力できる十分なスペースがあるかどうかをチェックします。十分なスペースがない場合は、以下のことが発生します。

- ページフッターが出力される
- 新しいページが追加される
- ページヘッダーが出力される
- 前ページに収まらなかったバンドの出力を続行する

## バンドのプロパティ

各バンドには出力処理に影響する有用なプロパティがあります。このプロパティはバンドのコンテキストメニューから設定できます。これを行うには、バンド上でオブジェクト以外の空白領域を右クリックします。このコンテキストメニューは、バンドヘッダー上 (クラシック表示モード使用時) を右クリック あるいは バンド構造上 (既定の非クラシック表示モード使用時) を右クリックした場合にも表示されます。別の方針 - バンドを選択し、[プロパティ] ウィンドウで対応するプロパティを変更します。

プロパティ	説明
CanGrow (拡張可能) / CanShrink (縮小可能)	これらのプロパティは、バンドに含まれるオブジェクトのサイズに応じてそのバンドを拡大または縮小させるかどうかを決定します。両方のプロパティが無効 (False) になると、バンドは常にデザイナーで指定されたサイズを維持します。この詳細については、「レポートの作成」の章の <a href="#">CanGrow, CanShrink プロパティ</a> を参照してください。
CanBreak (分断可能)	このプロパティを有効 (True) にすると、FastReport は出力可能なスペース上にバンドの内容の一部を出力します。つまり、バンドをブレークします。この詳細については、「レポートの作成」の章の <a href="#">データのブレークおよび同一ページに出力</a> を参照してください。
StartNewPage (新しいページを開始)	このプロパティを有効 (True) にすると、そのバンドは新しいページで出力を開始します。通常、このプロパティはグループを出力する際に使用されます。つまり、各グループは新しいページに出力されます。
PrintOnBottom (下部に出力)	このプロパティを有効 (True) にすると、そのバンドはページの下部でページ フッターの前に出力されます。これは、ページの下部に合計を出力させるような特定のドキュメントを出力するときに有用です。
RepeatOnEveryPage (ページごとに繰り返す)	このプロパティは、データヘッダー、データフッター、グループヘッダーおよびグループフッター バンドにあります。データの出力時、このタイプのバンドは新しいページ (改ページ) ごとに出力されます。この詳細については、「レポートの作成」の章の <a href="#">データのヘッダーおよびフッター</a> を参照してください。

## レポートオブジェクト

レポートではさまざまなオブジェクトが使用できます。

アイコン	名前	説明
	テキスト (TextObject)	1行または複数行のテキストを表示します。
	図 (PictureObject)	図を表示します。
	線 (LineObject)	線を表示します。縦線、横線または対角線が表示できます。
	図形 (ShapeObject)	四角形、楕円、三角形など幾何学的図形を表示します。
	リッチテキスト (RichObject)	書式付き (RTF 形式) のテキストを表示します。
	バーコード (BarcodeObject)	バーコードを表示します。
	チェックボックス (CheckBoxObject)	"有効" または "無効" の2つの状態を持つチェックボックスを表示します。
	表 (TableObject)	行、列およびセルを含む表を表示します。
	行列 (MatrixObject)	行列 (「クロス集計」としても知られる) を表示します。
	グラフ (MSChartObject)	グラフを表示します。
	郵便番号 (ZipCodeObject)	郵便番号を表示します。
	セル形式テキスト (CellularTextObject)	テキストの各文字を個々のセルに表示します。

オブジェクトを使用すれば、情報 (テキストオブジェクト) を表示したり、レポートの体裁 (図、線および図形オブジェクト) を向上させることができます。表や行列などの複雑なオブジェクトの場合、そのオブジェクト内にほかのシンプルなオブジェクトを含めることができます。

## オブジェクトの共通プロパティ

すべてのレポートオブジェクトは 1 つの基本クラス (ReportComponentBase) から継承され、特定の共通プロパティセットを持ちます。各オブジェクトについて説明する前に、これらの共通プロパティについて説明します。

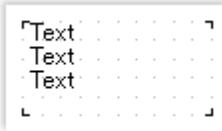
プロパティの値は [プロパティ] ウィンドウを使用して変更することができます。プロパティによっては、オブジェクトのコンテキストメニュー やツールバーから変更できるものもあります (罫線や塗りつぶしなど)。

プロパティ	説明
Left, Top, Width, Height	原則として、レポートオブジェクトは四角形です。オブジェクトは座標 (Left, Top プロパティ) とサイズ (Width, Height プロパティ) を持ちます。
Anchor	オブジェクトが置かれているコンテナが拡張または縮小されるときに、オブジェクトの位置やサイズを変更する方法を決定します。Anchor プロパティを使用することで、コンテナに同調してオブジェクトを拡張または移動させることができます。このプロパティの詳細については、動的なレイアウトセクションの <a href="#">Anchor プロパティ</a> をお読みください。
Dock	コンテナ内でオブジェクトがドッキングされる側面を決定します。このプロパティの詳細については、動的なレイアウトセクションの <a href="#">Dock プロパティ</a> をお読みください。
Border, Fill	オブジェクトの罫線と塗りつぶしを制御します。これらの設定は [罫線と塗りつぶし] ソールバーを使用して変更できます。
CanGrow, CanShrink	オブジェクト内にテキスト全体が収まるように、そのオブジェクトの高さを調整します。このプロパティの詳細については、動的なレイアウトセクションの <a href="#">CanGrow, CanShrink プロパティ</a> をお読みください。
ShiftMode	このプロパティを有効にしたオブジェクトは、拡張または縮小できる場合には上または下へ移動します。このプロパティの詳細については、動的なレイアウトセクションの <a href="#">ShiftMode プロパティ</a> をお読みください。
GrowToBottom	このプロパティを有効 (True) にしたオブジェクトは、バンドの下端まで引き伸ばされます。このプロパティの詳細については、動的なレイアウトセクションの <a href="#">GrowToBottom プロパティ</a> をお読みください。
CanBreak	このプロパティはテキストオブジェクトおよびリッチテキストオブジェクトにあります。オブジェクトのコンテンツを分割するかどうかを決定します。
PrintOn	オブジェクトを出力するページを決定します。このプロパティの詳細については、[小冊子タイプのレポート] セクションの <a href="#">奇数/偶数ページ上の印刷</a> をお読みください。
Cursor	オブジェクト上にマウスカーソルを置いたときに表示されるカーソルの種類を決定します。これはプレビュー ウィンドウでのみ動作します。
Visible	オブジェクトをレポートに表示するかどうかを決定します。オブジェクトを非表示にすると、その

<p>オブジェクトはプレビュー ウィンドウに表示されず、プリンター印刷時にも出力されません。</p>	
Printable	プリンター印刷時にそのオブジェクトを出力するかどうかを決定します。このプロパティを無効にした場合、そのオブジェクトはプレビュー ウィンドウには表示されますが、印刷時には出力されません。
Hyperlink	レポートオブジェクトを対話型にすることができます。このプロパティの詳細については <a href="#">対話型レポート</a> をお読みください。
Bookmark	このプロパティはHyperlink プロパティと共に使用します。このプロパティには式を含めることができます。式はレポートに対する作業時に計算され、その値はブックマーク名として使用されます。
Restrictions	オブジェクトの移動、サイズ変更、削除などの特定の操作を制限します。
Style	このプロパティにスタイル名を割り当てることができます。スタイル名を割り当てると、オブジェクトはそのスタイルで表示されるようになります。スタイルのパラメーターを変更すると、オブジェクトの外観も変わります。

## テキストオブジェクト

テキストオブジェクトは、使用頻度の高い主要なオブジェクトです。これは次のように表示されます。

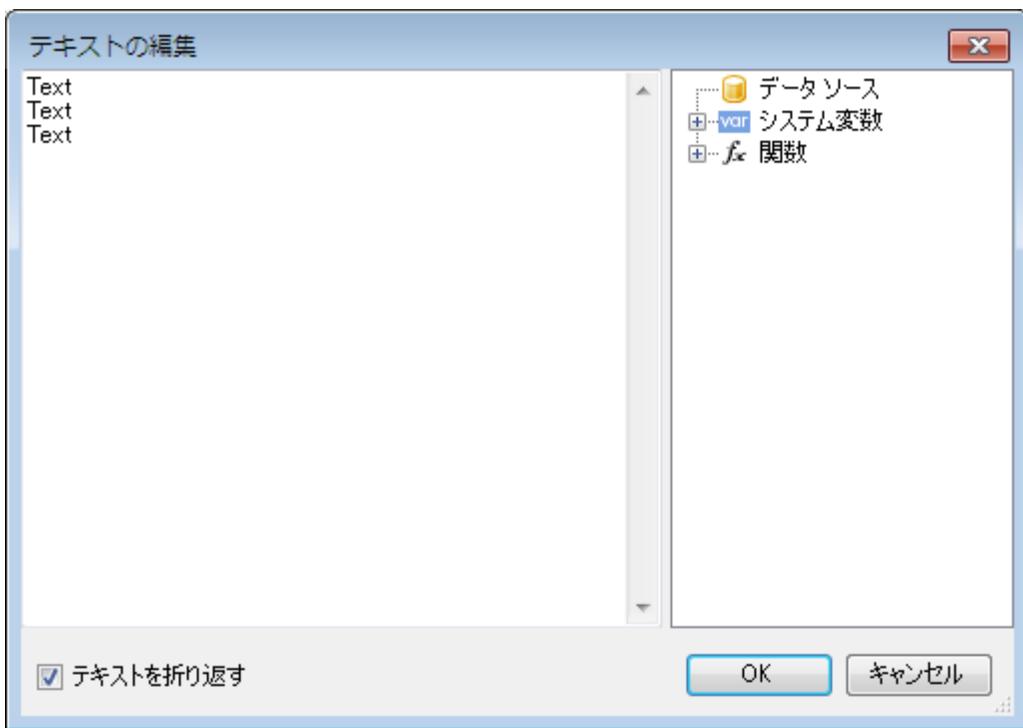


このオブジェクトはあらゆるテキストデータを表示できます。具体的には以下のよびなテキストが挙げられます。

- 1行または複数行のテキスト
- データ列
- レポートパラメーター
- 集計値
- 式
- 上記の項目の組み合わせ

## テキストの編集

オブジェクトのテキストを編集するには、そのオブジェクトをダブルクリックします。次のようなテキストエディターが表示されます。



エディターの右側にはデータツリーがあり、要素をテキストへ追加することができます。要素を追加するには、マウスを使用してその要素を目的の位置にドラッグします。要素をテキストへ追加する別の方法、要素をダブルクリックすると、その要素はカーソルの現在の位置に追加されます。

変更を保存してエディターを閉じる場合は、[OK]ボタンをクリックするか、Ctrl+Enterキーを押します。

別のテキスト編集方法 - インプレース(直接)編集。これを行うには、テキストオブジェクトを選択してEnterキーを押します。編集を終了するには、オブジェクト領域外の任意の場所をクリックするか、Ctrl+Enterキーを押します。Escキーを押すと変更が取り消されます。

オブジェクトを直接編集する場合、そのサイズはマウスを使用して変更することができます。

## 式の表示

テキストオブジェクトには、ブレーンテキストと式を混在させることができます。たとえば、次のように指定します。

今日は[Date]です。

このようなオブジェクトを出力する場合、テキストに含まれる式はすべて計算されます。結果は次のようになります。

今日は2010/09/12 です。

ご覧のとおり 式は角かっこによって識別されます。このかっこは Brackets プロパティで設定されます。このプロパティには既定で[]という文字列が含まれます。必要であれば <,> や<!,!> など別の記号セットを使用することもできます。たとえば最後に挙げた記号セット<!,!>を使用した場合、テキスト内の式は次のようにになります。

今日は<!Date!> です。

これと別に すべての式を無効にすることもできます。これを行うには AllowExpressions プロパティに **False** を設定します。この場合、テキストは記述されたとおりに表示されます。

角かっこ内では 有効な式がすべて使用できます。式の詳細については [式](#)の章をお読みください。たとえば 次のようなテキストを持つオブジェクトがあるとします。

2 \* 2 = [2 \* 2]

これは次のように出力されます。

2 \* 2 = 4

よくある間違いとして、角かっこ外側に式が記述されることがあります。角かっこ内に置かれているものが式と判断され実行されることを覚えておいてください。たとえば 次のように指定します。

2 \* 2 = [2] \* [2]

このテキストは次のように出力されます。

2 \* 2 = 2 \* 2

式内には自身に角かっこを必要とする要素があります。たとえば システム変数を参照することもあります(詳細については [式](#)の章の [データ要素の参照](#)を参考してください)。次の例を見てみましょう

次のページ [[Page] + 1]

このテキストには[Page] + 1 という式が含まれます。このPage は、現在のレポートページ番号を返すシステム変数です。この変数自身がかっこで囲まれています。これらのかっこはテキストオブジェクトの設定にかかわらず、角かっこでなければなりません。

厳密に言えば 上で挙げた例の Date システム変数を使用する場合は、次のように一対の角かっこを使用することになります。

今日は[[Date]] です。

しかし FastReport では式内に存在するメンバーが1 つのみの場合は、不要なかっここのペアを除外することができます。

## データ列の表示

以下の方法でデータ列を出力することができます。

[データソース名.列名]

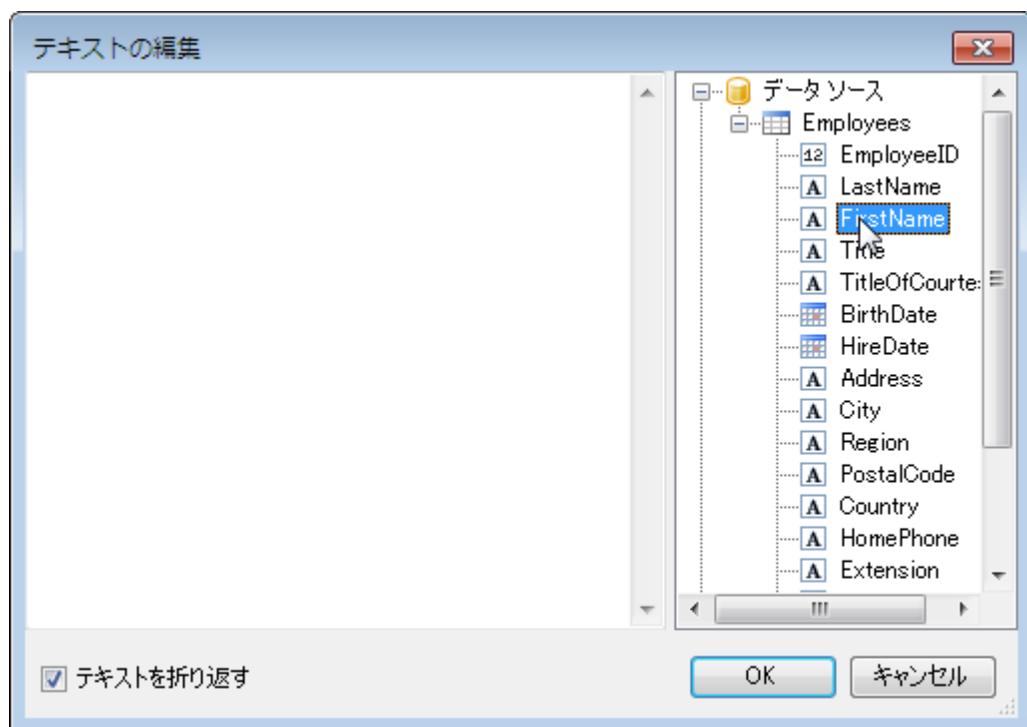
ここでは、ご覧のように角かっこを使用します。データソース名とデータ列名はピリオドで区切ります。たとえば、次のように指定します。

[Employees.FirstName]

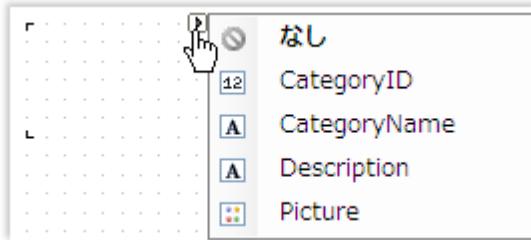
データ列の使用の詳細については、8章のデータソースの参照をお読みください。

テキストオブジェクトへデータ列を挿入する方法はいくつかあります。

1. テキストオブジェクトのエディターで、データ列の名前を手動で入力します。これは入力ミスが起こりやすいので最も不便な方法です。
2. テキストオブジェクトのエディターで、必要なデータ列を選択し、テキストフィールドへドラッグアンドドロップします。



3. テキストオブジェクトの右上隅にある小さなボタンをクリックし、一覧からデータ列を選択します。



4. [データ] ウィンドウからデータ列をレポートページへドラッグアンドドロップします。この場合、列へのリンクを含むテキストオブジェクトが作成されます。

## HTML タグ

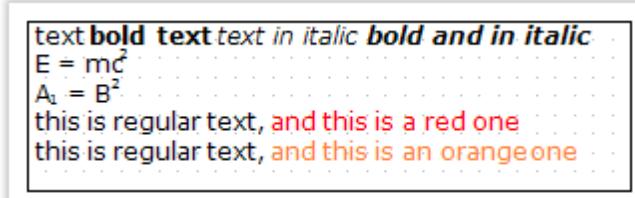
テキストオブジェクトでは、いくつかのシンプルなHTMLタグを使用することができます。既定では、HTMLタグは使用不可です。このタグを使用できるようにするには、[プロパティ] ウィンドウで HTMLTags プロパティを True に設定します。サポートされるタグの一覧を次に示します。

タグ	説明
<b>...</b>	太字
<i>...</i>	斜体
<u>...</u>	下線
<strike>...</strike>	取り消し線
<sub>...</sub>	下付き
<sup>...</sup>	上付き
<font color=...>...</font>	フォントの色。フォントの色は、色の名前 (DarkGrayなど) またはRGB形式の16進コード (#FF8030など) で指定できます。

これらのタグの使用方法を次の具体例で示します。

```
text <b>bold text</b> <i>text in italic</i> <b><i>bold and in italic</i></b></i>
E = mc<sup>2</sup>
A<sub>1</sub> = B<sup>2</sup>
this is regular text, <font color=red>and this is a red one</font>
this is regular text, <font color=#FF8030>and this is an orange one</font>
```

このテキストは、次のように表示されます。



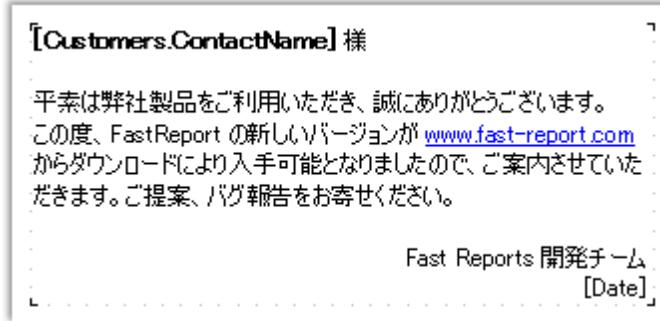
## オブジェクトのプロパティ

プロパティ	説明
AllowExpressions	式による処理を有効または無効にすることができます。既定では有効 (True)です。
Angle	テキストの回転を角度で示します。
AutoShrink	フォントサイズまたはフォント幅をテキストに合わせて自動的に縮小させることができます。
AutoShrinkMinSize	AutoShrink プロパティを使用した場合には、フォントサイズまたはフォント幅の最小値を設定します。
AutoWidth	オブジェクトの幅を自動的に計算します。
Brackets	式の識別に使用する一対の記号が含まれます。
BreakTo	あるテキストオブジェクトから別のテキストオブジェクトへのテキストフローを有効にします。たとえば、A および B というテキストオブジェクトがあるとします。A オブジェクトには、そのサイズ範囲内に収まらない長いテキストが含まれています。A.BreakTo プロパティに B に設定すると、B オブジェクトには A オブジェクトに収まらなかったテキスト部分が表示されるようになります。
Clip	オブジェクトのサイズ範囲内に収まらないテキストを切り抜く(クリッピングする)かどうかを決定します。既定では有効 (True)です。
Duplicates	重複する値の出力方法を決定します。このプロパティの詳細については、書式設定「セクションの重複する値を非表示にする」を読みください。
FirstTabOffset	最初のタブストップのオフセットをピクセル単位で設定します。
FontWidthRatio	フォントの幅を広くまたは狭くします。既定では、このプロパティには 1 が設定されます。フォントの幅を広げるには 1 より大きい値を設定します。フォントの幅を狭くするには、0 から 1 までの範囲の値を設定します。
HideValue	これは文字列型のプロパティです。このプロパティの値と同じ値を非表示にすることができます。このプロパティの詳細については、書式設定「セクションのゼロの値を非表示にする」を読みください。
HideZeros	ゼロの値を非表示にすることができます。このプロパティの詳細については、書式設定「セクションのゼロの値を非表示にする」を読みください。
Highlight	強調表示の条件を設定できます。この詳細については、書式設定「セクションの条件付き強調表示」を読みください。

HorzAlign、VertAlign	テキストの配置を決定します。
HtmlTags	オブジェクトのテキスト内でシンプルなHTMLタグを使用できるようになります。このプロパティの詳細については、「テキストオブジェクト」セクションの <a href="#">HTMLタグ</a> をお読みください。
LineHeight	テキスト行の高さを明示的に設定することができます。既定値は0に設定されているので、既定の行間が使用されます。
NullValue	ヌル値の代わりに出力されるテキストです。[レポートオプション]ダイアログ([レポート]メニューで[オプション]を選択)で[ヌル値を変換する]オプションのチェックをオフにすることも必要です。
Padding	パディング(オブジェクト内余白)をピクセル単位で設定できます。
RightToLeft	テキストを右から左に向かって表示させるかどうかを示します。
TabWidth	タブストップの幅をピクセル単位で設定します。
Text	オブジェクトのテキストが含まれます。
TextFill	テキストの塗りつぶしを設定します。このプロパティエディターを使用して、塗りつぶしのタイプを選択します。
Trimming	オブジェクトのサイズ範囲内に収まらないテキストのトリミング方法を決定します。これはWordWrapプロパティにFalseが設定されている場合にのみ使用します。
Underlines	各テキスト行の下にグラフィカルな線を表示できます。このプロパティは、テキストが上揃えになっている場合にのみ使用できます。
WordWrap	テキストを単語単位で折り返す(改行)するかどうかを決定します。
Wysiwyg	画面と印刷結果が一致するようテキストオブジェクトの表示モードを変更します。この表示モードは両端揃え、または非標準の行の高さを使用する場合にも使用されます。

## リッチ テキストオブジェクト

このオブジェクトは、書式設定されたテキスト(RTF形式)を表示します。これは次のように表示されます。

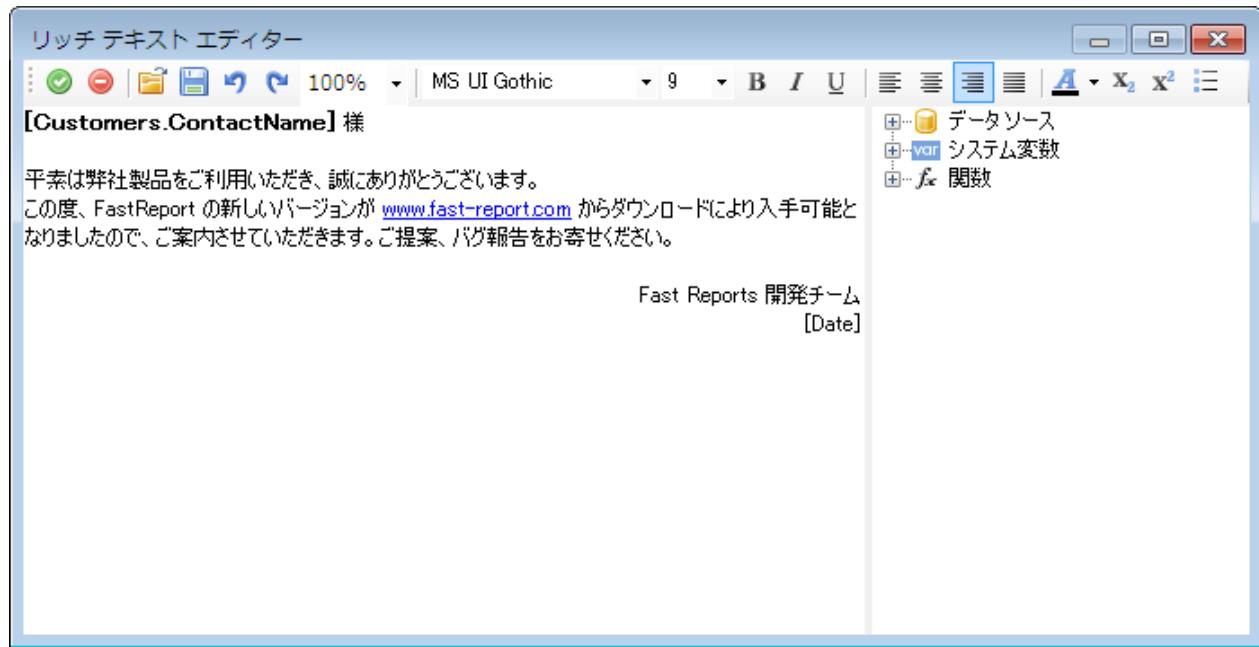


単なるテキストを表示する場合はテキストオブジェクトを使用するようこして下さい。レポートをエクスポートする場合、そのエクスポート先のド

キュメント形式によっては、リッチテキストオブジェクトが図としてエクスポートされることがあります。

このオブジェクトは、純色の塗りつぶしのみをサポートしています。グラデーション、ハッチ、ガラスの塗りつぶしはサポートしていません。

オブジェクトを編集するには、そのオブジェクトをダブルクリックします。エディターウィンドウが表示されます。



テキストは Microsoft Word を使用して作成することもできます。Word でテキストを作成したら、そのテキストを .RTF 形式で保存します。

次に、リッチテキストエディターを開き、 ボタンをクリックしてその .RTF ファイルを読み込みます。

リッチテキストオブジェクトは Microsoft Word のすべての書式設定機能をサポートしているわけではありません。

以下の方法で、このオブジェクトにデータを表示できます。

- テキストオブジェクトの場合と同様に、オブジェクトのテキスト式を挿入できます。必要なデータ列をテキストに挿入します。
- DataRow プロパティを使用してオブジェクトを列にバインドします。

このオブジェクトには以下のプロパティがあります。

プロパティ	説明
AllowExpressions	式による処理を有効または無効にすることができます。既定では有効 (True) です。
Brackets	式の識別に使用する一対の記号が含まれます。

DataColumn	このオブジェクトがバインドするデータ列です。
Text	RTF テキストが含まれます。
Padding	パディング (オブジェクト内余白) ピクセル単位で指定します。

## 図オブジェクト

図オブジェクトは、BMP、PNG、JPG、GIF、TIFF、ICO、EMF および WMF 形式の図を表示することができます。この図オブジェクトを使用して、会社のロゴ、従業員の写真のほかあらゆるグラフィカル情報を出力できます。このオブジェクトは次のように表示されます。



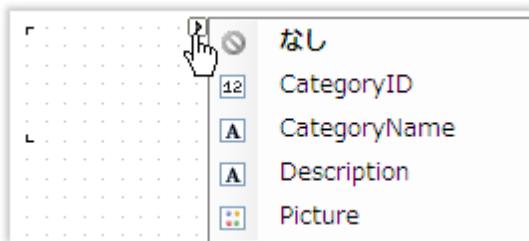
オブジェクトは以下のソースからデータを表示できます。

ソース	説明
図のファイル	図はファイルから読み込まれ、レポート内に保存されます。図は Image プロパティに格納されます。
データ列	データ列の図。列の名前は DataColumn プロパティに格納されます。
ファイル名	図は指定した名前のファイルから読み込まれます。ファイルの名前は ImageLocation プロパティに格納されます。図はレポート内に保存されません。図はファイルとしてレポートと一緒に配布する必要があります。
URL	図はレポートが作成されるたびにインターネットから読み込まれます。図はレポート内に保存されません。URL は ImageLocation プロパティに格納されます。

図エディターを呼び出すには、その図オブジェクトをダブルクリックします。そのエディターで、図のデータソースを選択できます。



オブジェクトをデータ列へバインドするには、オブジェクトの右上隅にある小さなボタンをクリックし、一覧からデータ列を選択します。

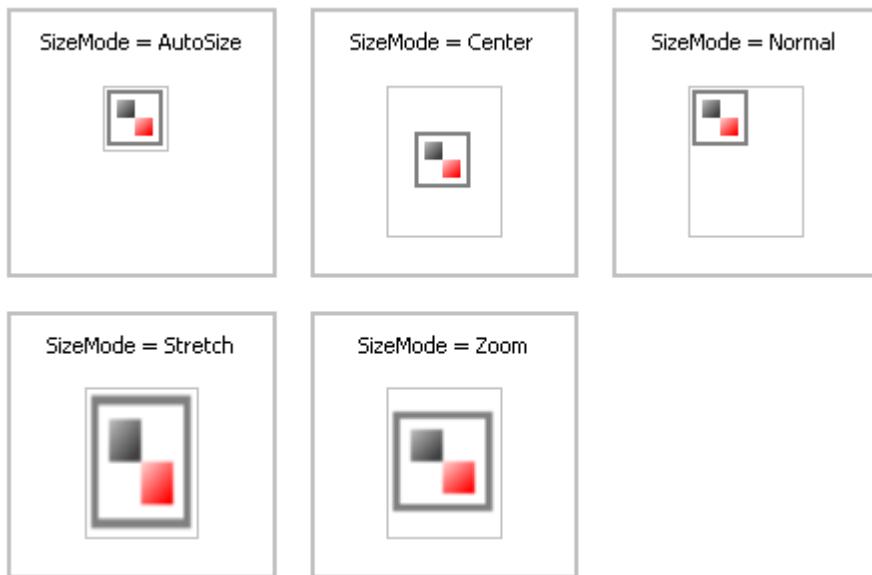


[データ] ウィンドウから [データ列] をレポートページへドラッグアンドドロップします。この場合、列へのリンクを含む図オブジェクトが作成されます。ドラッグする列は Byte[] データ型である必要があります。

図オブジェクトのコンテキストメニューの [サイズ モード] から、以下のモードのいずれかを選択することができます (SizeMode プロパティに対応)。

- ・自動サイズ調整 (AutoSize) : オブジェクトは図のサイズを取得します。
- ・イメージを真中に置く (CenterImage) : 図はオブジェクト内の中間に配置されます。
- ・標準 (Normal) : 図はオブジェクトの左上隅に表示されます。
- ・イメージを引き伸ばす (StretchImage) : 図はオブジェクトのサイズと同等になるよう引き伸ばされます。
- ・拡大 (Zoom) : 図は縦横比を維持しながら拡大されます。

各サイズモードの違いを以下の図で示します。



図オブジェクトには以下のプロパティがあります。

プロパティ	説明
Angle	回転角度(度単位)。このプロパティに設定できる値は、0、90、180、270です。
SizeMode	サイズモード。
Transparency	図の透過度。値は0～1の間で設定できます。値0(既定値)は図が不透明であることを意味します。
TransparentColor	図を表示するときに透過される色。
Image	図。
DataColumn	このオブジェクトがペイントするデータ列です。
ImageLocation	ファイル名またはURLを指定することができます。図はレポートの作成時にこの場所から読み込まれます。
Padding	パディング(オブジェクト内余白)。ピクセル単位で指定します。
ShowErrorImage	図が空の場合に、図がないことを表す図を表示します。このプロパティは、図がインターネットからダウンロードされる場合などの使用に適しています。

## 線オブジェクト

線オブジェクトは、横線、縦線、対角線を表示することができます。このオブジェクトは次のように表示されます。



可能であれば、線オブジェクトではなくオブジェクトの墨線を使用してください。これによってレポートが単純化され、異なる形式へエクスポートする際に起こり得る問題を回避することができます。

FastReport デザイナーには線を描画するための便利なツールがあります。レポートに線を挿入するには、オブジェクトツールバーで ボタンをクリックし、一覧から"線"または"対角線"を選択します。線の開始位置にマウスカーソルを置きます。次に、マウスの左ボタンを押したまま移動して線を描画します。その後も続けて線を描画することができます。すべての線が描画できたら、オブジェクトツールバーの ボタンをクリックします。

対角線とは異なり、普通の線で描画できるのは縦線または横線のみです。

このオブジェクトには二重線を適用するスタイルを選択しないでください。このスタイルが適用されるのはオブジェクトの墨線のみです。

線オブジェクトには以下のプロパティがあります。

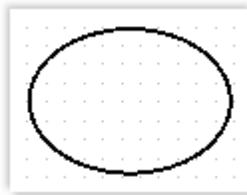
プロパティ	説明
Diagonal	線が対角線かどうかを決定します。普通の線に対してこのプロパティを有効 (True) にすると、その線を対角線にすることができます。
StartCap、EndCap	線のキャップスタイルを設定することができます。以下のキャップスタイルのいずれかを使用できます。 <ul style="list-style-type: none"><li>円 (Circle)</li><li>正方形 (Square)</li><li>ひし形 (Diamond)</li><li>矢印 (Arrow)</li></ul> キャップのサイズはそのキャップの Width、Height プロパティで設定できます。キャップは線の両端それぞれに設定できます。

## 図形オブジェクト

図形オブジェクトは、以下の図形のいずれかを表示します。

- 四角形
- 角丸四角形
- 楕円
- 三角形
- ひし形

このオブジェクトは次のように表示されます。



レポートに図形を挿入するには、オブジェクトツールバーで  ボタンをクリックし、一覧から希望の図形タイプを選択します。

この図形は、ほかのレポートオブジェクトと同様に塗りつぶしや罫線を設定できます。テキストオブジェクトとは逆に、各罫線を制御することはできません。また、二重線を適用するスタイルも使用しません。

可能であれば、四角形図形ではなくオブジェクトの罫線を使用してください。

図形オブジェクトには以下のプロパティがあります。

プロパティ	説明
Shape	図形の種類を決定します。
Curve	角丸四角形 (RoundRectangle) 図形で使用されます。これは曲線を設定できます。

## バーコードオブジェクト

レポート内にバーコードを表示するオブジェクトです。これは次のように表示されます。



このオブジェクトは、以下のバーコードタイプをサポートします。

コード	長さ	使用可能なシンボル
Interleaved (インターリード) 2of5		0 ~ 9
Industrial (インダストリアル)2of5		0 ~ 9
Matrix (マトリックス)2of5		0 ~ 9
Codabar		0 ~ 9、-\$:/.+
Code128		ASCII 128 文字
Code39		0 ~ 9、A ~ Z、-.*\$/+%
Code39 Extended		ASCII 128 文字
Code93		0 ~ 9、A ~ Z、-.*\$/+%
Code93 Extended		ASCII 128 文字
EAN8	8	0 ~ 9
EAN13	13	0 ~ 9
MSI		0 ~ 9
PostNet		0 ~ 9
UPC A	12	0 ~ 9
UPC E0	6	0 ~ 9
UPC E1	6	0 ~ 9
2 桁の補足コード	2	0 ~ 9

5 桁の補足コード	5	0 ~ 9
PDF417		すべて
Datamatrix		すべて

バーコードの種類の詳しい説明については、インターネットで情報を参照してください。たとえば、次のようなサイト(英語)があります。

<http://www.barcodeisland.com>

オブジェクト内のバーコードデータは文字列型です。この文字列には、選択したバーコードのタイプで使用可能なシンボルを含めることができます。バーコードオブジェクトのコンテキストメニューでバーコードの種類を選択できます。

以下の方法のいずれかを使用してオブジェクトをデータに接続できます。

- Text プロパティにバーコードデータを設定する
- DataColumn プロパティを使用してオブジェクトをデータ列へバインドする
- Expression プロパティにバーコードデータを返す式を設定する

バーコードオブジェクトには以下のプロパティがあります。

プロパティ	説明
Barcode	バーコード固有の設定が含まれています。このプロパティを展開して、それらの設定を指定します。
Angle	バーコードの回転角度(度単位)を決定します。0、90、180、270 の値のいずれかを使用できます。
Zoom	バーコードを拡大/縮小することができます。このプロパティはAutoSize プロパティの設定に応じて使用します。
AutoSize	このプロパティをTrue にすると、バーコード全体をそのまま表示するようにオブジェクトが拡張されます。このプロパティをFalse にすると、オブジェクトの境界内に収まるようバーコードが表示されます。
ShowText	人間が判読できるテキストを表示するかどうかを決定します。
DataColumn	オブジェクトがバインドするデータ列です。
Expression	バーコードデータを返す式。
Text	バーコードデータ
Padding	パディング(オブジェクト内余白)、ピクセル単位で指定します。

以下のプロパティは、バーコードの種類に応じて設定可能なプロパティです。これらを変更するには、[プロパティ] ウィンドウで Barcode プロパティを(ダブルクリックして)展開します。

プロパティ	説明
WideBarRatio	これは1D(1 次元)バーコードタイプ固有のプロパティです。バーの幅の比率を決定します。ほとんどのバーコードに対し、このプロパティの値は2 から3 の間で指定してください。
CalcCheckSum	これは1D(1 次元)バーコードタイプ固有のプロパティです。チェックサムを自動的に計算するかどうかを決定します。このプロパティをFalse にする場合は、バーコードデータにチェックデジットを提供してください。
AutoEncode	これはCode128 バーコード固有のプロパティです。このコードにはABC という3 種類のエンコードがあります。バーコードデータに明示的にエンコードを設定するか、このプロパティにTrue を設定します。この場合、エンコードは自動的に選択されます。

バーコードデータでは次の制御コードを使用します。

コード	意味
&A;	START A / CODE A
&B;	START B / CODE B
&C;	START C / CODE C
&S;	SHIFT
&1;	FNC1
&2;	FNC2
&3;	FNC3
&4;	FNC4

AutoEncode プロパティにTrue を設定すると すべての制御コードが無視されます。

制御コードの使用例 :

&C;1234&B;ABC

AspectRatio	これはPDF417 バーコード固有のプロパティです。縦横比を決定します。この値によってバーコードサイズが自動的に計算されます(Columns およびRows プロパティが設定されていない場合)。
CodePage	これは PDF417 およびDatamatrix バーコード固有のプロパティです。非 ASCII 文字への変換に使用するコードページを決定します。たとえば Windows の既定のコードページは1251 です。

Columns、Rows	これらはPDF417 バーコード固有のプロパティです。バーコード内の列数および行数を決定します。両方のプロパティに0 を設定すると、バーコードのサイズは自動的に計算されます。この場合、AspectRatio プロパティが使用されます。
CompactionMode	これはPDF417 バーコード固有のプロパティです。PDF417 データの圧縮モードを決定します。
ErrorCorrection	これはPDF417 バーコード固有のプロパティです。エラーの修正レベルを決定します。
PixelSize	これはPDF417 バーコード固有のプロパティです。バーコード要素のサイズをピクセル単位で決定します。一般に、要素の高さまたは幅の3倍以上になるよう設定します。
Encoding	これはDatamatrix バーコード固有のプロパティです。Datamatrix データのエンコードを決定します。
PixelSize	これはDatamatrix バーコード固有のプロパティです。バーコード要素のサイズをピクセル単位で決定します。
SymbolSize	これはDatamatrix バーコード固有のプロパティです。バーコードシンボルのサイズを決定します。

## チェックボックス オブジェクト

レポート内にチェックボックスを表示するオブジェクトです。これは次のように表示されます。



このオブジェクトはチェックの"オン" と"オフ" という2つの状態を表示することができます。以下の方法を使用して、オブジェクトの状態を処理します。

- Checked プロパティで状態を設定する
- DataColumn プロパティを使用してオブジェクトをデータ列へバインドする
- Expression プロパティにTrue またはFalse を返す式を設定する

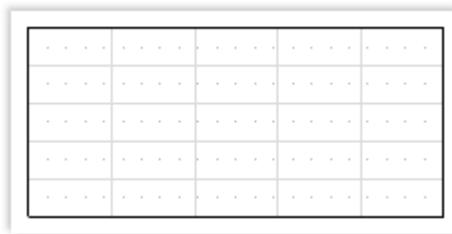
CheckBox オブジェクトには以下のプロパティがあります。

プロパティ	説明
CheckedSymbol、 UncheckedSymbol	オブジェクトの状態に応じて、オブジェクト内に表示される記号を決定します。
CheckColor	チェック記号の色を決定します。
CheckWidthRatio	チェック記号の幅の比率を設定します。チェック記号の幅はオブジェクトのサイズによって決

まします。値は0.2 ~ 2 の範囲で指定できます。	
HideIfUnchecked	チェックがオフになった場合はオブジェクトを非表示にすることができます。
Checked	オブジェクトの状態を制御します。
DataTable	オブジェクトがバインドするデータ列です。列のデータ型はBool またはInt になるようにしてください。
Expression	True またはFalse を返す式です。

## 表オブジェクト

表オブジェクトは行、列およびセルで構成されています。このオブジェクトはMicrosoft Excel の表を簡略化したものです。これは次のように表示されます。



このオブジェクトの詳細については、「レポートの作成」の章の[表タイプのレポート](#)を参照してください。

表オブジェクトには以下のプロパティがあります。

プロパティ	説明
ColumnCount	列数をすばやく設定する場合に使用します。表の列数を必要に応じて追加または削除できます。
RowCount	行数をすばやく設定する場合に使用します。表の行数を必要に応じて追加または削除できます。
FixedColumns	表内で固定にする列数を決定します。固定された列は表ヘッダーになります。ヘッダーの出力はRepeatHeaders プロパティによって制御されます。
FixedRows	表内で固定にする行数を決定します。固定された行は表ヘッダーになります。ヘッダーの出力はRepeatHeaders プロパティによって制御されます。
RepeatHeaders	ページが変わったときに表ヘッダーを出力できるようにします。このプロパティは動的に形成される表に対してのみ動作します。

## 行列オブジェクト

行列オブジェクトは表オブジェクトのように行、列およびセルで構成されています。しかし、行列内に入る行数と列数は事前にわかつません。これはこのオブジェクトが接続するデータによって決まります。

このオブジェクトは次のように表示されます。

Employee	[Year]	集計
[Name]	[Revenue]	
集計		

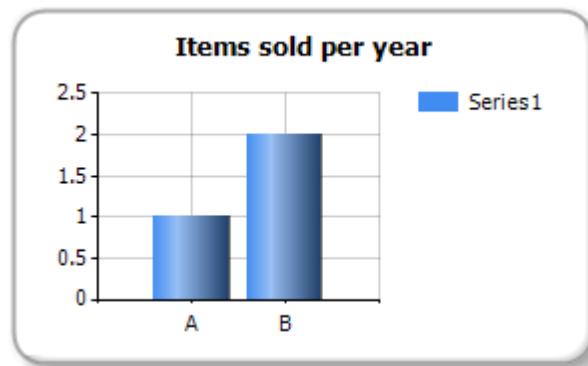
このオブジェクトの詳細については、「レポートの作成」の章の[行列タイプのレポート](#)を参照してください。

行列オブジェクトには以下のプロパティがあります。

プロパティ	説明
RepeatHeaders	行列が複数ページに渡って出力される場合、各ページごとに行列のヘッダーを出力します。
CellsSideBySide	行列が複数のデータセルレベルを持つ場合に、行列のセルの配置方法を決定します。指定できる配置方法は次のとおりです。 <ul style="list-style-type: none"><li>セルは横に並んで表示されます。</li><li>セルは縦に並んで表示されます。</li></ul>
Style	行列全体のスタイルを設定できます。定義済みのスタイルの中から1つを選択できます。
AutoSize	行列のサイズを自動的に計算することができます。オブジェクトのサイズを手動で設定する場合はこのプロパティを無効 (False)にしてください。
DataSource	行列をデータソースへ接続できます。データ列を行列へドラッグすると、このプロパティは自動的に設定されます。ただし、セル式を使用する場合は、このプロパティが正しく設定されていることを確認してください。
Filter	このプロパティには、行列のデータソースに対して適用されるデータフィルター処理の式が含まれています (DataSource プロパティを参照)。

## MS Chart オブジェクト

このMS Chart オブジェクトを使用すると、グラフを表示することができます。30種類以上の系列タイプ(棒、列、領域、折れ線、バブル、円、円形、財務、ピラミッド、範囲)が使用できます。このオブジェクトは次のように表示されます。



このオブジェクトの詳細については、「レポートの作成」の章の[「グラフ付きのレポート」](#)を参照してください。

MS Chart オブジェクトには以下のプロパティがあります。

プロパティ	説明
Chart	Microsoft Chart オブジェクトへの参照。
AlignXValues	このプロパティは系列数が異なる場合に(空の値を挿入することによって)X値の数を揃えることができます。これはグラフに2つ以上の系列が含まれる場合に使用します。
AutoSeriesColumn、 AutoSeriesColor、 AutoSeriesSortOrder	これらのプロパティは自動的に作成される系列を設定します。この詳細については、「レポートの作成」の章の <a href="#">「系列の自動作成」</a> を参照してください。
DataSource	このプロパティを使用すると、グラフをデータソースへ接続できます。
Filter	このプロパティには、グラフのデータソースに対して適用されるデータフィルター処理の式が含まれています (DataSource プロパティを参照)。

## 郵便番号オブジェクト

郵便番号オブジェクトを使用すると、封筒に郵便番号を印刷することができます。数字(0 ~ 9)を表示することができます。

このオブジェクトは次のように表示されます。



以下の方法のいずれかを使用してオブジェクトをデータと接続できます。

- Text プロパティに郵便番号データを設定する

- DataColumn プロパティを使用してオブジェクトをデータ列へバインドする
- Expression プロパティに郵便番号データを返す式を設定する

郵便番号オブジェクトには以下のプロパティがあります。

プロパティ	説明
SegmentCount	番号のセグメント数。既定値は6です。
SegmentWidth、 SegmentHeight	1つの番号セグメントのサイズ。既定のサイズは0.5cm 1cm。
Spacing	2つのセグメントの原点間の距離を決定します。既定値は0.9cmです。
ShowGrid	グリッドを表示するかどうかを決定します。
ShowMarkers	マーカー(郵便番号の各番号上部に表示される太字の横線)を表示するかどうかを決定します。
DataColumn	オブジェクトがバインドするデータ列です。
Expression	郵便番号データを返す式。
Text	郵便番号が含まれるテキスト。

## セル形式テキストオブジェクト

このオブジェクトは、テキストの各文字を個々のセルに表示します。これは財務アプリケーションで定型フォームを出力するためによく使用されます。

このオブジェクトは次のように表示されます。

S	i	m	p	l	e		t	e	x	t				
---	---	---	---	---	---	--	---	---	---	---	--	--	--	--

実際には、このオブジェクトはテキストオブジェクトから直接継承されたものです。同じ方法でオブジェクトをデータへ接続することができます。たとえば、オブジェクトのエディターを起動して次のようなテキストを入力することができます。

[Employees.FirstName]

セル形式テキストオブジェクトには以下のプロパティがあります。

プロパティ	説明
CellWidth、CellHeight	これらのプロパティは1つのセルのサイズを決定します。両方のプロパティが0(既定値)の場

合、セルのサイズは使用されているフォントに応じて自動的に計算されます。

HorzSpacing、VertSpacing

これらのプロパティは、隣接するセル間の横間隔および縦間隔を決定します。

## Fast Report でレポートを作成する

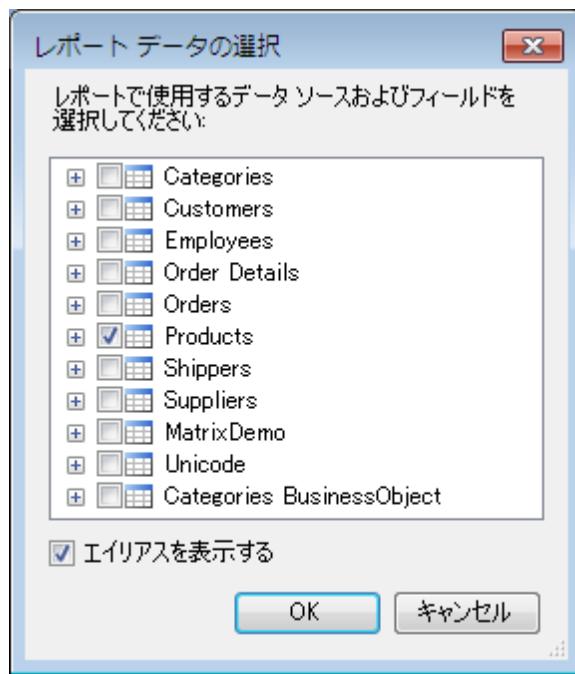
製品の一覧を出力するシンプルなレポートを作成してみましょう。Products というテーブルを使用します。このテーブルはデータソースとして本製品に付属するデモ用データベースに存在します。

これはレポートデザイナーを呼び出すことができるデモプログラム Demo.exe で、以降のトピックで記述されている手順を実行することを前提としています。

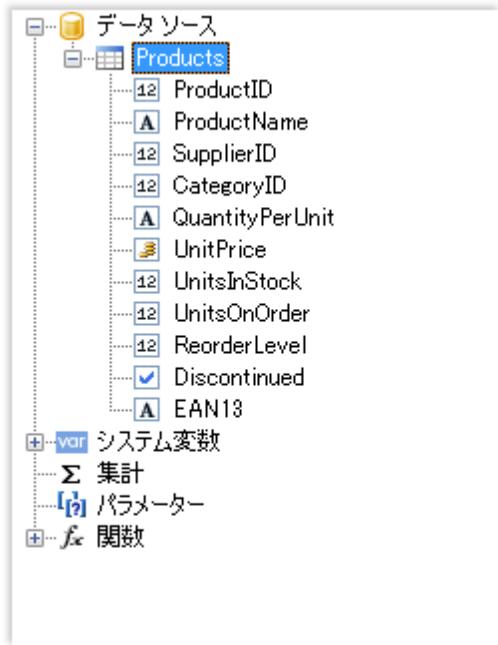
### 例 1 : レポートの手動作成

この例では、レポートを手動で作成します。このため、以下のことを行います。

- ツールバー上の  ボタンをクリックし、[新しい項目の追加] ウィンドウで [空白レポート] を選択します。
- [データ] メニューで [レポートデータの選択] を選択し、[レポートデータの選択] ダイアログで "Products" データソースのチェックをオンにして [OK] をクリックします。



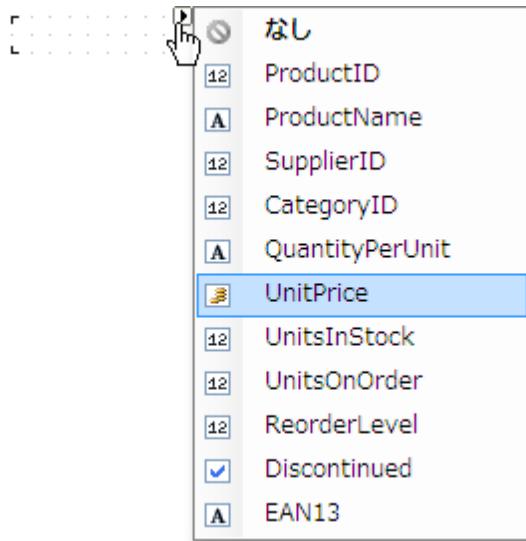
- [データ] ウィンドウに移ります (画面上に表示されていない場合は、[データ] メニューの [データウィンドウの表示] を選択して表示します)。"データソース" 項目を展開し、さらに "Products" 項目を展開します。



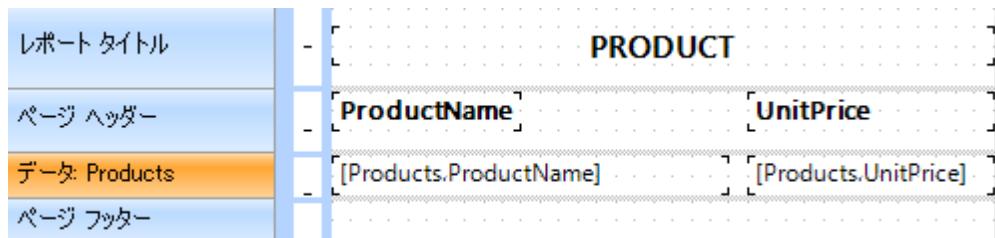
- 「ProductName」データ列をデータバンドへドラッグします。FastReport によってテキストオブジェクトが作成されます。このオブジェクトはこの「ProductName」データ列とそのヘッダーに接続されます。
- 別の方法で、「UnitPrice」データ列を作成してみましょう。オブジェクトツールバーにある「テキスト」ボタンをクリックします。



- マウスポインターをデータバンド上にドラッグします。FastReport によってオブジェクトの挿入が提示されることがわかります。挿入したい位置を選び、マウスをクリックするとオブジェクトが挿入されます。
- そのオブジェクト上にマウスポインターを置き、オブジェクトの右上隅にある小さなボタンをクリックします。データ列の一覧が表示されます。一覧から「UnitPrice」項目を選択します。



- 「UnitPrice」列のヘッダー用のテキストオブジェクトを作成します。これはページヘッダーバンド上に置きます。オブジェクトをダブルクリックし、「Unit Price」というテキストを入力します。
- レポートタイトル用のテキストオブジェクトを作成します。レポートタイトルバンド上に置き、「PRODUCT」というテキストを入力します。
- ページヘッダーおよびレポートタイトルバンドに置かれているすべてのオブジェクトのフォントスタイルを太字に設定します。これを設定するには、Shiftキーを押したまま各オブジェクトを選択し、テキストツールバー上の **B** ボタンをクリックします。これで、レポートは次のように表示されます。



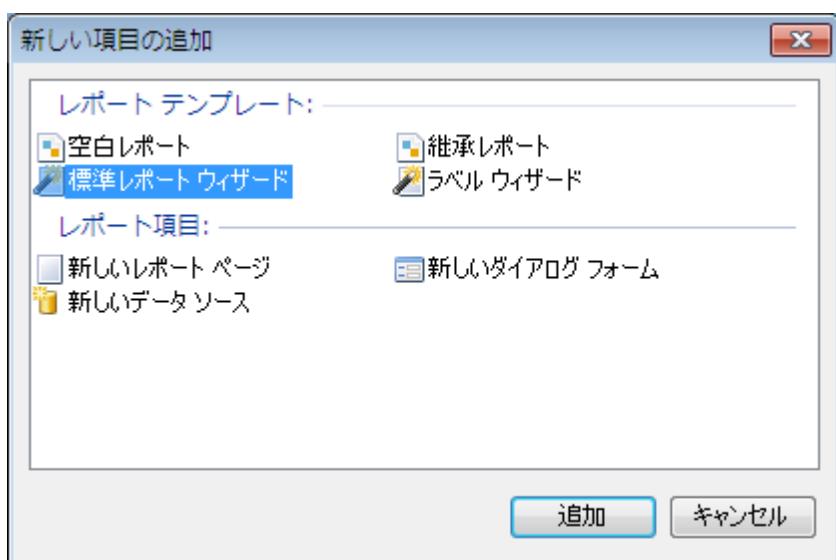
レポートを実行するには、ツールバー上の  ボタンをクリックします。レポートが実行され、プレビュー ウィンドウに表示されます。

PRODUCT	
Product Name	Unit Price
Chai	1,800
Chang	1,900
Aniseed Syrup	1,000
Chef Anton's Cajun Seasoning	2,200
Chef Anton's Gumbo Mix	2,135
Grandma's Boysenberry Spread	2,500
Uncle Bob's Organic Dried Pears	3,000
Northwoods Cranberry Sauce	4,000
Mishi Kobe Niku	9,700
Ikura	3,100

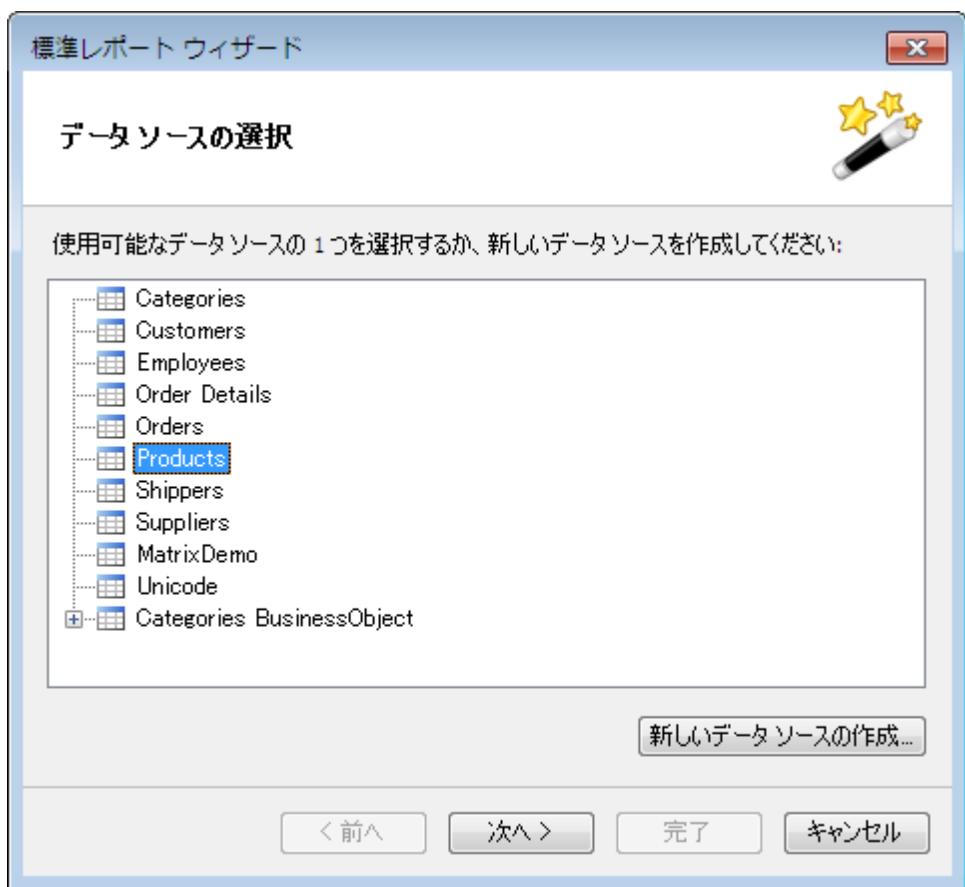
## 例 2 : ウィザードを使ったレポートの作成

この例では、"標準レポートウィザード" を使用してレポートを作成します。このため、以下のことを行います。

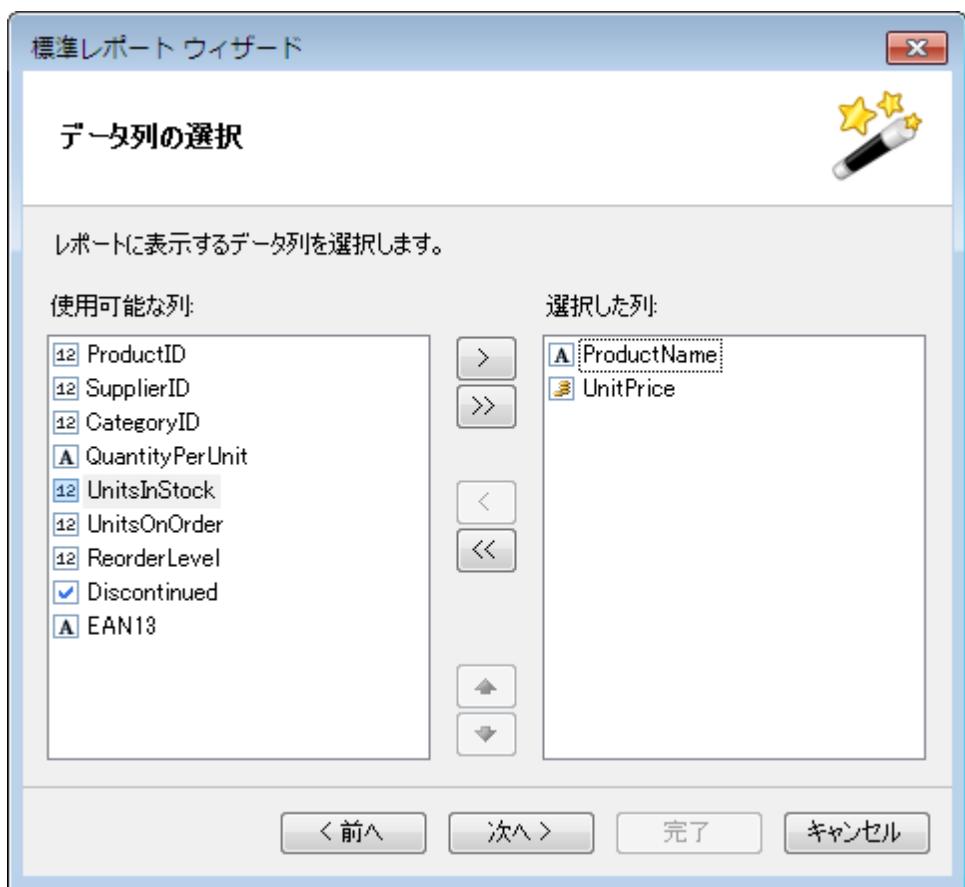
- ツールバー上の  ボタンをクリックし、[新しい項目の追加] ウィンドウで "標準レポートウィザード" を選択します。



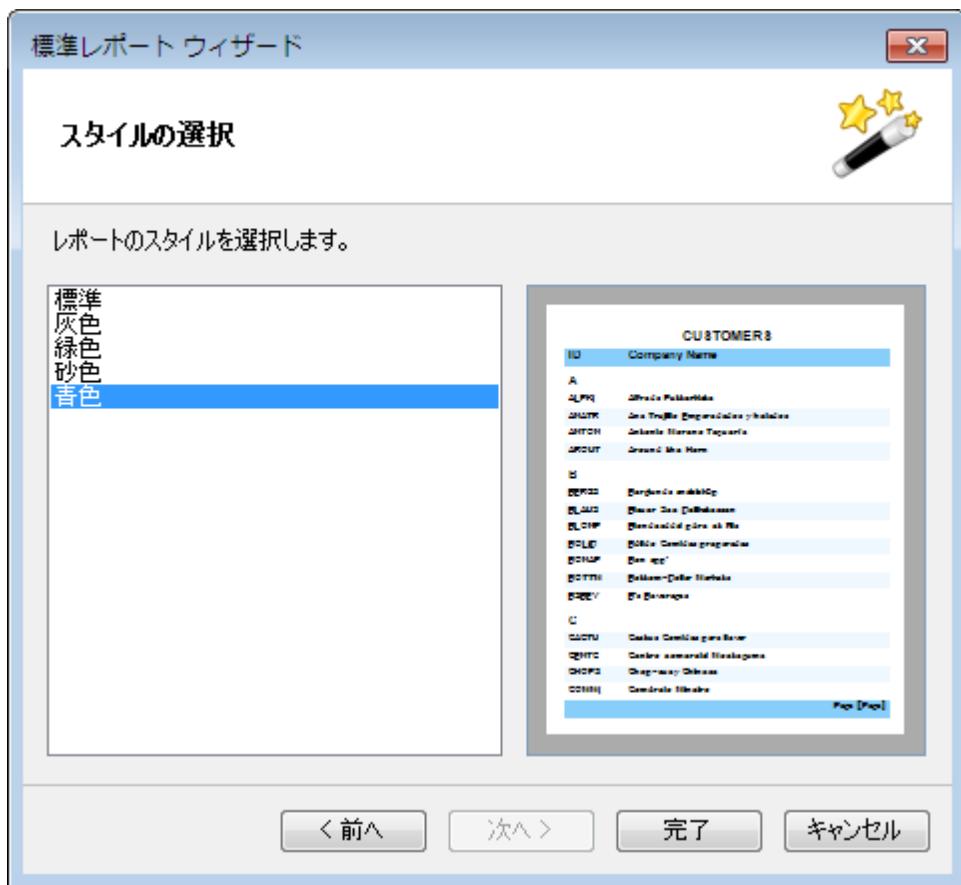
- ウィザードの最初のステップで、"Products" テーブルを選択して [次へ] ボタンをクリックします。



- ウィザードの2番目のステップで、"ProductName" および "UnitPrice" データ列を選択します。



- 以降のステップは既定の設定のまま、[次へ]ボタンをクリックして います。
- ウィザードの最後のステップ([スタイルの選択])では、"青色"スタイルを選択し、[完了]ボタンをクリックします。



FastReport では次のようなレポートを作成します。

レポートを実行するには、ツールバー上の  ボタンをクリックします。レポートが実行され、プレビュー ウィンドウに表示されます。

Products	
ProductName	UnitPrice
Chai	1800
Chang	1900
Aniseed Syrup	1000
Chef Anton's Cajun Seasoning	2200
Chef Anton's Gumbo Mix	2135
Grandma's Boysenberry Spread	2500
Uncle Bob's Organic Dried Pears	3000
Northwoods Cranberry Sauce	4000
Mishi Kobe Niku	9700
Ikura	3100
Queen Cabraler	2100



## 第2章

# レポートの作成

## レポートの作成

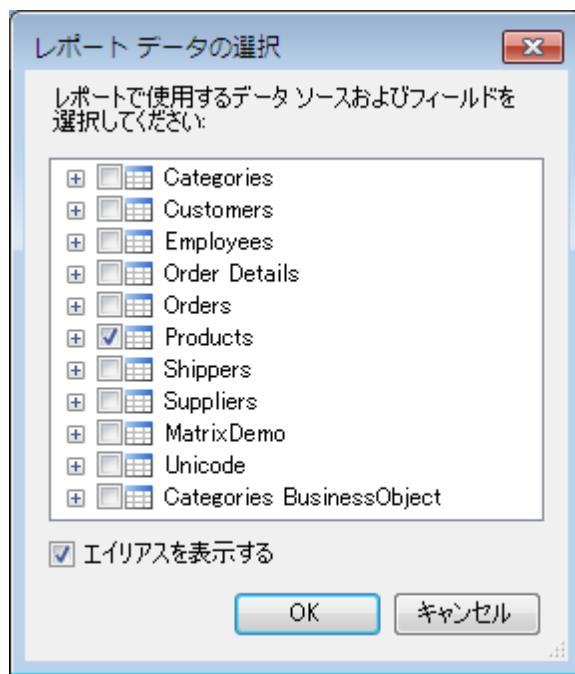
この章では、一般的なレポートの作成方法について見てみましょう。原則として、レポートを作成するには以下のことを行います。

1. レポートで使用するデータを選択または作成します。
2. 必要なバンドをレポートに追加してレポート構造を作成します。
3. バンドをデータソースへ接続します。
4. バンド上に"テキスト"オブジェクトを配置しデータを出力します。
5. 外観、書式を設定します。

## レポート用データの選択

レポートの作成を始める前に、レポートに出力するデータを選択する必要があります。これには2つ方法があります。

- 1つのデータソースを選択できます。これはプログラミングによってレポートに登録されたものです。[データ]メニューの[レポートデータの選択]を選択して表示されるダイアログで、目的のデータソースを選択します。



- [データ]メニューの[データソースの追加]を選択して新しいデータソースを作成します。

データソースの詳細については、[データ]の章の[データソース](#)をお読みください。

選択されたデータソースは[データ]ウィンドウに表示されます。これでこのソースをレポートで使用できるようになります。レポートの多くは1つのデータソースのみを使用します。"マスター/詳細"タイプのレポートの場合は、相互に関係を持つ2つのデータソースを使用する必要があります(関係の詳細については、[データ]の章の[関係](#)をお読みください)。また、関連する複数のソースからデータを出力するレポートでは、いくつものデータソースが必要となることがあります。

## 動的なレイアウト

レポートの作成時には、サイズが不明なテキストを出力しなければならない状況も多いでしょう。たとえば、商品の説明などがこれに該当します。これを解決するには、以下の作業が必要です。

- テキスト全体がオブジェクト内に収まるように、オブジェクトの高さを計算する
- テキストサイズが可変のオブジェクトがバンド内に収まるように、バンドの高さを計算する
- レポートの全体的なデザインに支障が出ないように、バンド上に含まれるその他のオブジェクトを移動または高さを変更する

これらの作業は、以下に挙げるオブジェクトやバンドのプロパティを使用して達成できます。

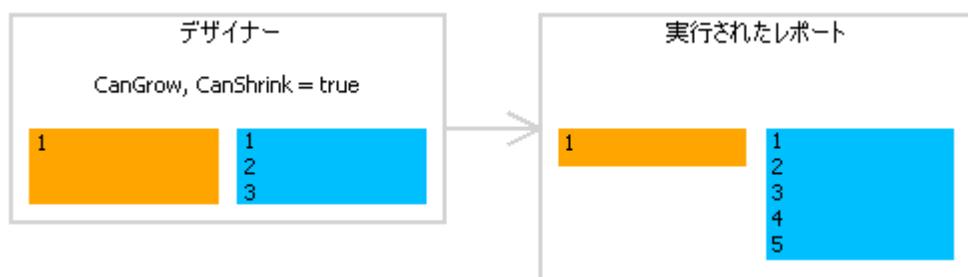
- CanGrow および CanShrink プロパティを使用すると、オブジェクトの高さを自動的に計算できます。
- ShiftMode プロパティを使用すると、オブジェクトを、その上にあるオブジェクトの拡張または縮小に応じて上または下へ移動させることができます。
- GrowToBottom プロパティを使用すると、オブジェクトをバンドの下端まで拡張します。
- Anchor および Dock プロパティを使用すると、バンドのサイズに応じてオブジェクトのサイズを制御します。

次のセクションではこれらのプロパティについて説明します。

### CanGrow, CanShrink プロパティ

バンドおよびレポートオブジェクトにはこれらのプロパティがあります。オブジェクトのサイズを、その内容のサイズに応じて拡張または縮小するかどうかを決定します。両方のプロパティを無効 (False) にすると、オブジェクトは常にデザイナーで指定されたサイズになります。

デザイン時にはサイズが不明なテキストを出力する必要がある場合は、これらのプロパティが役立ちます。オブジェクト内にテキスト全体が収まるようにするには、CanGrow および CanShrink プロパティを有効 (True) にする必要があります。

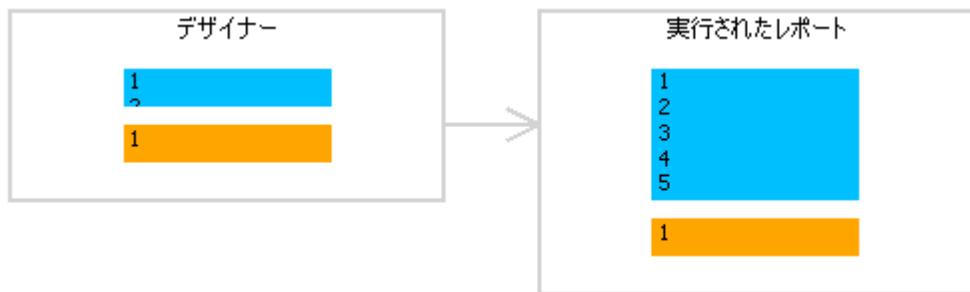


以下のオブジェクトはバンドの高さによって影響を受けます。

- テキスト
- リッチテキスト
- 図 (SizeMode プロパティに"AutoSize" が設定済み)
- 表

## Shift Mode プロパティ

すべてのレポートオブジェクトにこのプロパティがあります。このプロパティが設定できるのは [プロパティ] ウィンドウのみです。ShiftMode プロパティを有効にしたオブジェクトは、その上にあるオブジェクトが拡張または縮小される場合には上または下へ移動します。



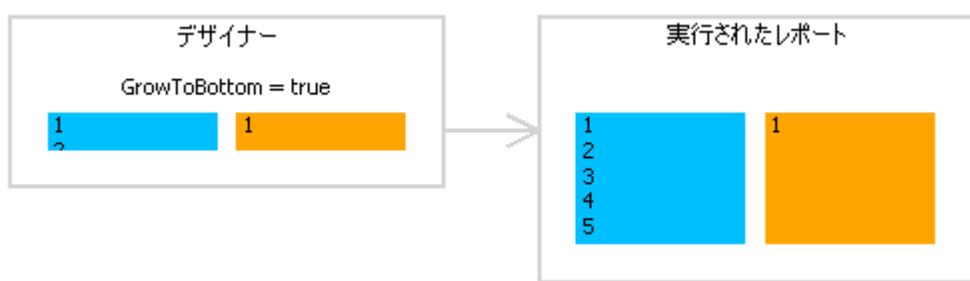
ShiftMode プロパティには次のいずれかの値を設定できます。

- Always (既定値) : オブジェクトは常に移動します。
- Never : オブジェクトは移動しません。
- WhenOverlapped : オブジェクトは、その上にあるオブジェクトが拡張され自身が覆われてしまう(両オブジェクトが水平に重なる)場合に移動します。

このプロパティは、情報を表のフォームで出力するときに、その表のいくつかのセルが互いに重なり合うように配置され、かつセル内のテキストの量が可変である場合に利用できます。

## GrowToBottom プロパティ

すべてのレポートオブジェクトにこのプロパティがあります。このプロパティを使ってオブジェクトを出力する場合、そのオブジェクトをバンドの下端まで引き伸ばします。



これは情報を表のフォームで出力する場合などに必要です。表の1行内には、拡張する可能性のあるオブジェクトがいくつも存在することがあります。このプロパティは、すべてのオブジェクトの高さをバンドと同じ高さに設定することができます。

## Anchor プロパティ

すべてのレポートオブジェクトにこのプロパティがあります。オブジェクトが置かれているコンテナのサイズが変更されたときに、オブジェクトの位置やサイズを変更する方法を決定します。Anchor を使用することで、そのコンテナに同調してオブジェクトを拡張または移動させることができます。

多くの場合、コンテナは"バンド"を指しています。しかし、これは"表"や"行列"オブジェクトを指すこともあります。

Anchor プロパティは以下の値のいずれか、または値の組み合わせを持ちます。

値	説明
Left	オブジェクトの左端にアンカーを固定します。コンテナのサイズが変更された場合、オブジェクトは左右に移動しません。
Top	オブジェクトの上端にアンカーを固定します。コンテナの高さが変更された場合、オブジェクトは上下に移動しません。
Right	オブジェクトの右端にアンカーを固定します。コンテナの幅が変更された場合、オブジェクトの右端とコンテナの右端との距離は変わりません。さらにオブジェクトの左端にアンカーが固定された場合は、オブジェクトはコンテナと同期して拡張および縮小します。
Bottom	オブジェクトの下端にアンカーを固定します。コンテナの高さが変更された場合、オブジェクトの下端とコンテナの下端との距離は変わりません。さらにオブジェクトの上端にアンカーが固定された場合は、オブジェクトはコンテナと同期して拡張および縮小します。

このプロパティの既定値はLeft, Topです。これは、コンテナのサイズが変更されてもオブジェクトは変更されないことを意味します。次の表では、よく使用される値の組み合わせを示します。

値	説明
Left, Top	既定値。コンテナのサイズが変更されてもオブジェクトは変更はありません。
Left, Bottom	コンテナの高さが変更された場合、オブジェクトは上または下に移動します。コンテナの下端に対するオブジェクトの位置は変わりません。
Left, Top, Bottom	コンテナの高さが変更された場合、それに応じてオブジェクトの高さが変わります。
Left, Top, Right, Bottom	コンテナの幅と高さが変更された場合、それに応じてオブジェクトも拡張または縮小します。

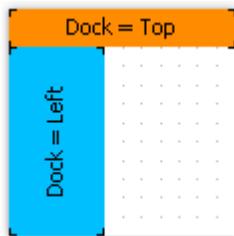
## Dock プロパティ

すべてのレポートオブジェクトにこのプロパティがあります。このプロパティは、オブジェクトがコンテナ内でドッキングされる位置を決定します。

Dock プロパティには次のいずれかの値を設定できます。

値	説明
None	既定値。オブジェクトはドッキングされません。
Left	オブジェクトはコンテナの左端にドッキングされます。オブジェクトの高さはコンテナと同じ高さになります。*
Top	オブジェクトはコンテナの上端にドッキングされます。オブジェクトの幅はコンテナと同じ幅になります。*
Right	オブジェクトはコンテナの右端にドッキングされます。オブジェクトの高さはコンテナと同じ高さになります。*
Bottom	オブジェクトはコンテナの下端にドッキングされます。オブジェクトの幅はコンテナと同じ幅になります。*
Fill	オブジェクトはコンテナの空き領域全体を占めるようにドッキングされます。

\* 複数のオブジェクトが同時にドッキングされている場合は、コンテナの高さや幅と同じではありません。次の図は2つのオブジェクトを表しています。1つはコンテナの上端にドッキングされ、もう1つは左端にドッキングされています。



このように、2つ目のオブジェクトの高さは、1つ目のオブジェクトがドッキングされた後に残った空き領域の高さと同じになります。

ドッキングの動作はオブジェクトの作成順序によって異なります。この順序は、オブジェクトのコンテキストメニューで変更することができます。これをを行うには、メニューから [最前面に移動] または [最背面に移動] を選択します。

## 書式設定

ここでは、以下の事項について説明します。

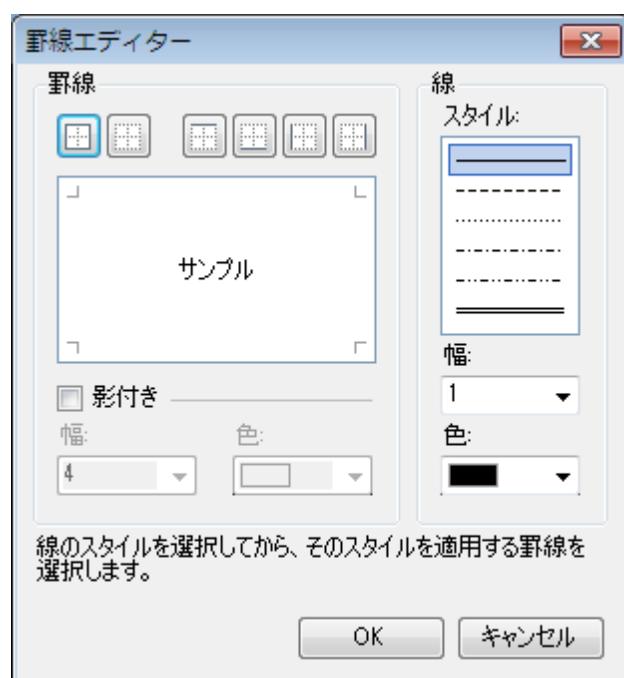
- オブジェクトの外観を変更する
- 出力する値の書式を変更する
- ある条件を満たす場合にオブジェクトの外観を自動的に変更する
- 不要な値を非表示にする
- 1行おきに異なる色で強調表示する

### 罫線と塗りつぶし

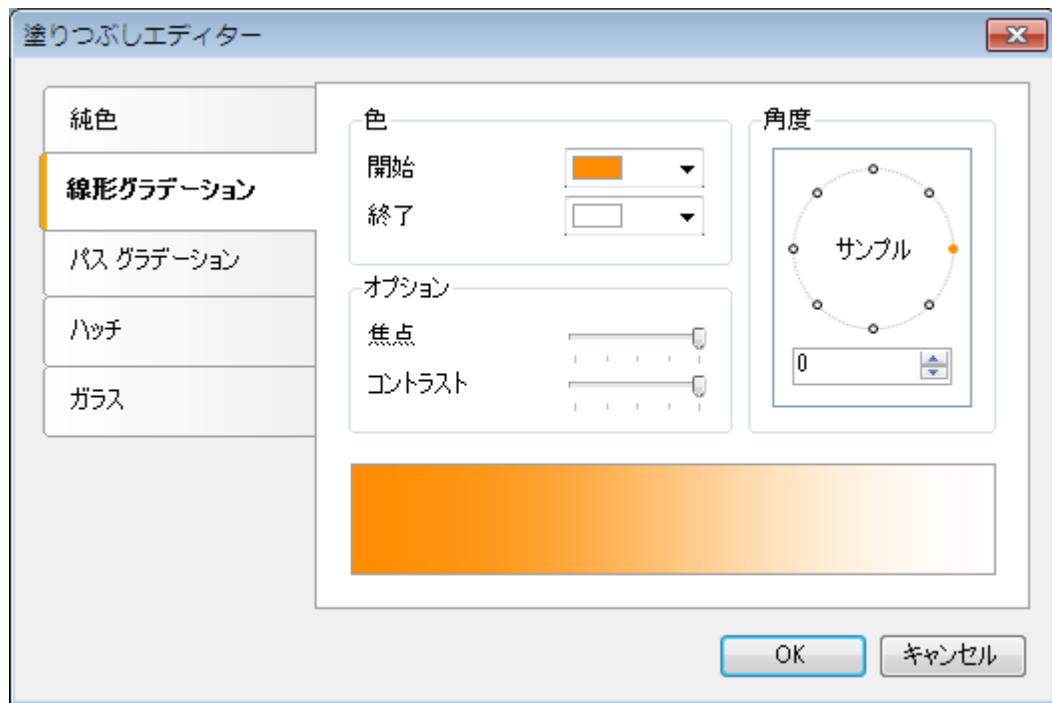
ほとんどのレポートオブジェクトで、罫線や塗りつぶしを設定することができます。これらのプロパティを設定するには、**罫線と塗りつぶしツールバー**を使用します。



オブジェクトの罫線は4つの線で構成されます。各線に異なる幅、色およびスタイルを指定することができます。ツールバーのボタンがすべての枠線に作用します。  ボタンをクリックすると、各線を個別に設定できるダイアログが表示されます。

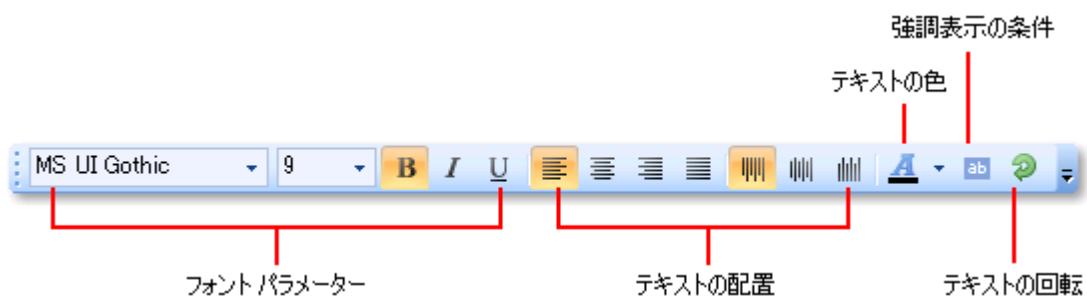


塗りつぶしの設定を行なう場合、ツールバーには2つのボタンがあります。① ボタンをクリックすると、純色の塗りつぶしタイプの色を選択できます。② ボタンをクリックすると、純色以外の塗りつぶしタイプを選択できるダイアログが表示されます。



## テキストの書式設定

テキストオブジェクトの外観を変更するには、[テキスト]ツールバーを使用します。



## スタイル

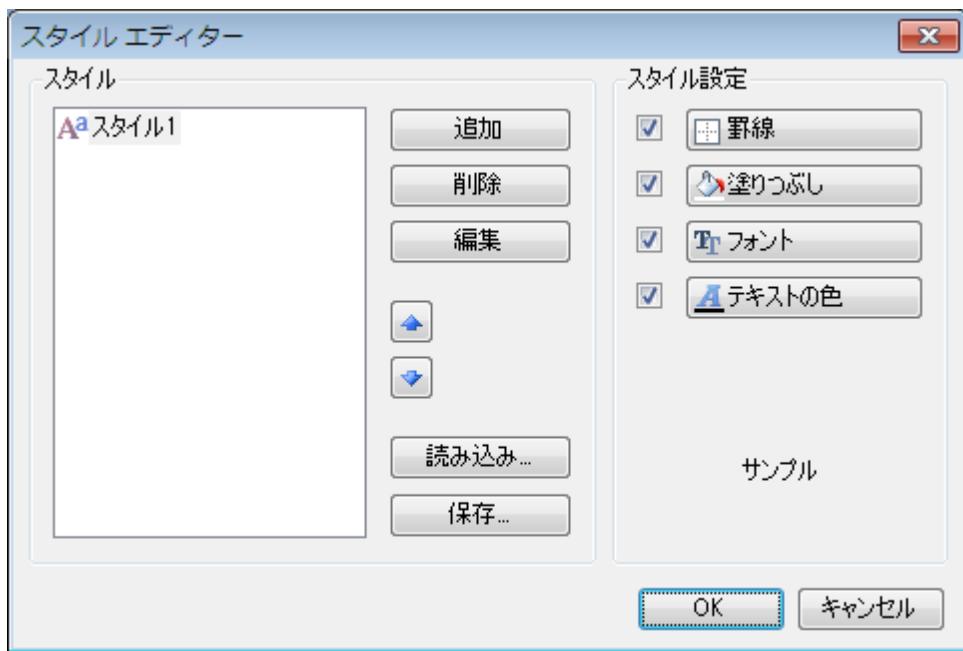
オブジェクトの外観を設定する場合はスタイルを使用します。スタイルは以下のプロパティがセットになったものです。

- 罫線
- 塗りつぶし

- フォント
- テキストの色

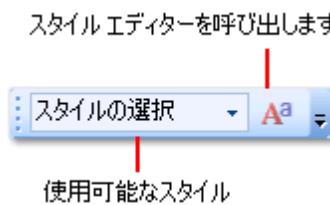
スタイルのリストはレポートに保存されます。スタイルの制御は、[レポート]メニューで[スタイル]を選択、あるいは[スタイル]ツールバーで

**A<sup>a</sup>** ボタンをクリックして行います。



オブジェクトのスタイルは以下の方法で設定することができます。

- [プロ/パティ] ウィンドウで Style プロパティを設定する
- [スタイル]ツールバーを使用する



[スタイル]ツールバーが表示されてない場合は、[表示]メニューの[ツールバー | スタイル]を選択すると表示されます。

オブジェクトにスタイルを設定した場合、オブジェクトの外観はそのスタイルの設定に従って変更されます。スタイルの設定を変更した場合、そのスタイルが適用されているオブジェクトには自動的に変更が反映されます。

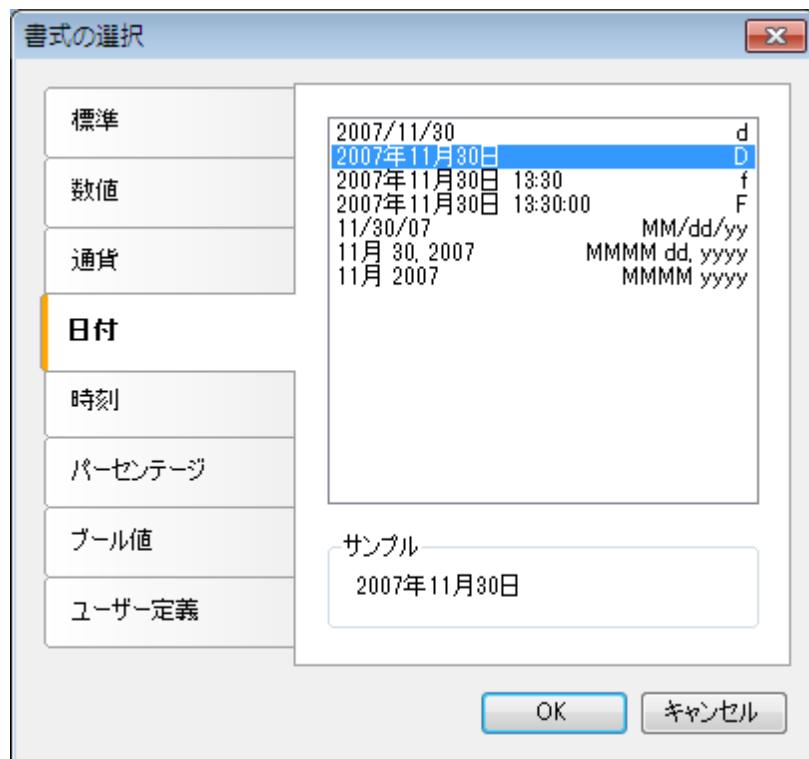
注意: オブジェクトのスタイルを"スタイルなし"に設定した場合、現在適用されているスタイル設定がクリアされるわけではありません。これはオブジェクトとそのスタイルとの関連付けを解除し、それ以後、そのスタイルの設定変更が適用されないようにするものです。

## データの書式設定

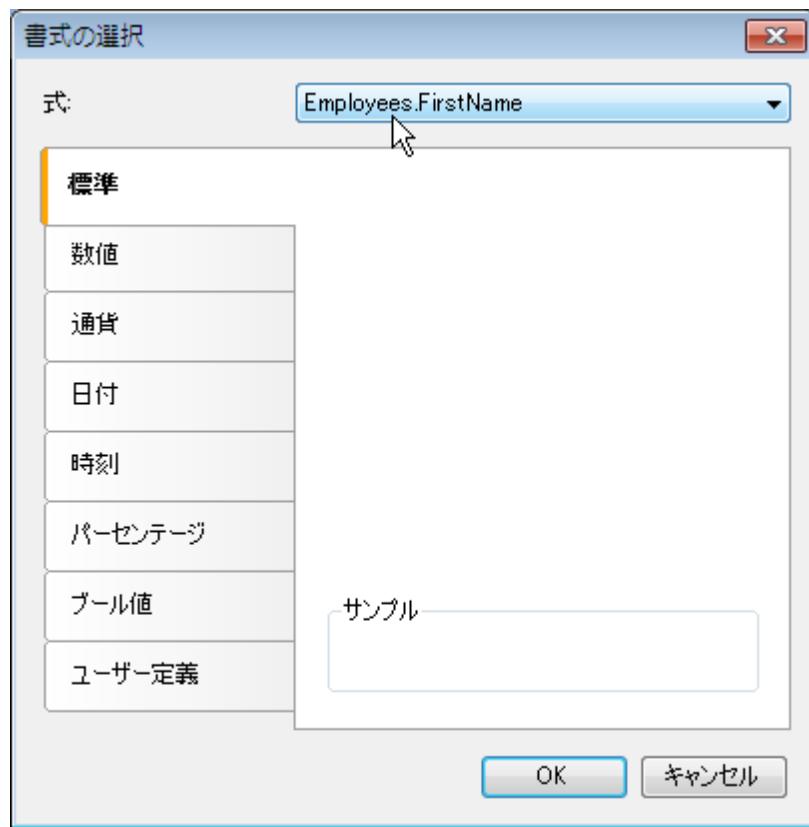
テキストデータをレポートに出力する場合はテキストオブジェクトを使用します。データソースからのデータには既定の書式設定が適用されます。たとえば、データソースの列のデータ型が日付時刻型の場合、次のように出力されます（これは、お使いのシステムの「地域」の設定によって異なります）。

11.10.2008 18:04:52

日付部分のみを出力する必要がある場合、データの書式を設定する必要があります。これを行うには、まずテキストオブジェクトを右クリックしてコンテキストメニューを表示します。このメニューから「書式」を選択します。次のような書式エディターのウインドウが表示されます。



使用可能な書式設定のタイプを選択するか、独自の書式文字列を設定することができます。これを行うには、「ユーザー定義」を選択します。テキストオブジェクトに複数のデータ列または式が含まれる場合は、それぞれの項目に対して適切な書式を選択することができます。これを行うには、エディターの上部で式を選択し、書式を選択します。

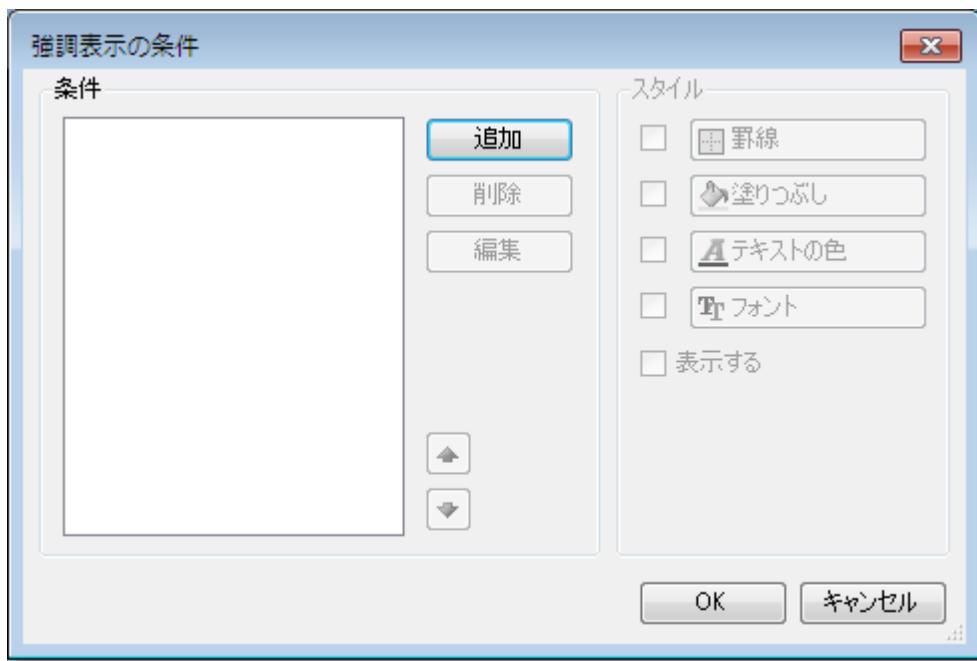


String.Format メソッドを使用して次のようにデータの書式を設定することもできます。このメソッドについては MSDN のヘルプを参照してください。

今日は[String.Format("{0:d}", [Date])] です。

## 条件付き強調表示

テキストオブジェクトの外観を、指定した条件に応じて変更することができます。たとえば、オブジェクトの値が負の値の場合には、そのオブジェクトを赤色で強調表示させることができます。この機能を"条件付き強調表示"と言います。これを設定するには、テキストオブジェクトを選択し、[テキスト]ツールバーの  ボタンをクリックします。次のようなダイアログが表示されます。

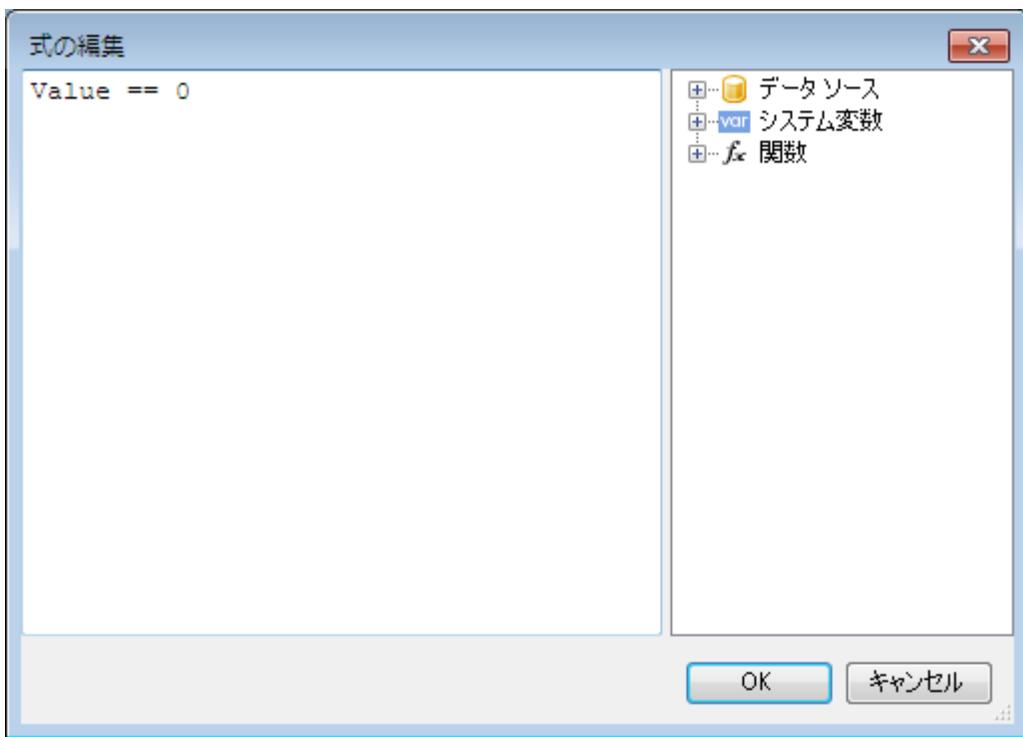


1つまたは複数の条件を定義し、条件ごとにスタイルを設定することができます。スタイルには、以下の設定を含めることができます。

- 塗りつぶし
- テキストの色
- フォント
- オブジェクトの表示/非表示

条件と一致した場合に変更する設定項目を指定できます。[スタイル]カテゴリのチェックボックスを使用して必要な設定項目をチェックします。新しいスタイルには既定で1つの設定 "テキストの色" が含まれます。

新しい条件を作成するには、[追加]ボタンをクリックします。次のような式エディターが表示されます。



ここで、ブール型の結果を返す式を記述することができます。ほとんどの場合、Value 変数を使用します。これには現在出力している値が含まれます。

次の例を見てみましょう。製品の在庫数を出力する1つのテキストオブジェクトがあります。

[Products.UnitsInStock]

製品の在庫数が0のときはそのオブジェクトを赤色で出力するようになります。このため、次の条件を作成します。

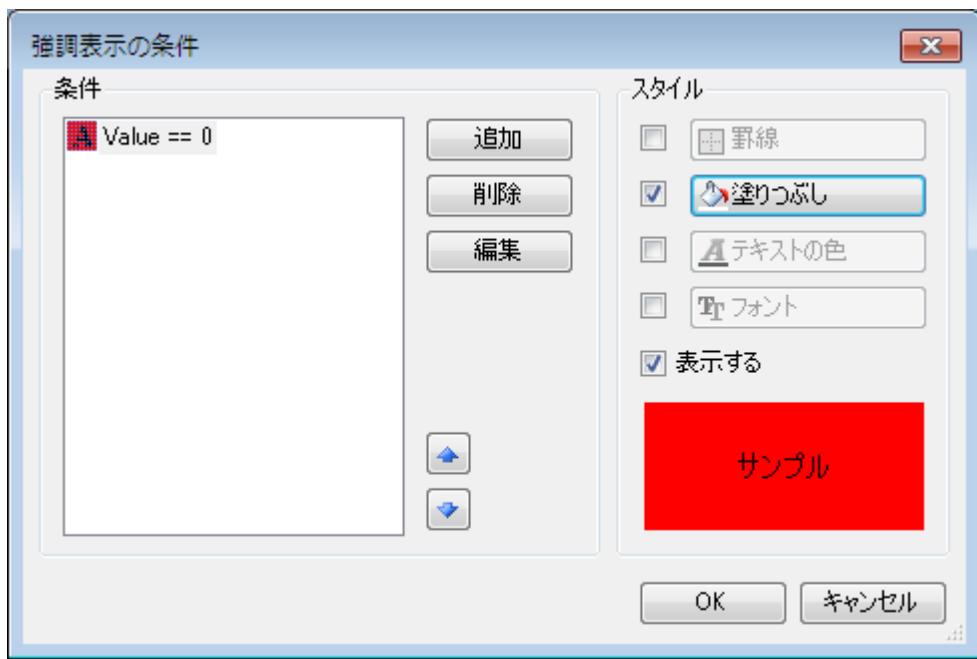
Value == 0

指定した例では、出力される1つの値を持つValue変数を使用しています。オブジェクト内で複数の式がある場合、この変数は最後の式の値を持ちます。Valueの代わりにデータ列を使用できます。

[Products.UnitsInStock] == 0

この式はC#スタイルで記述されています。これは、レポートで選択されている言語がC#である場合を想定しています。VisualBasic.NETの場合は"=" (等号記号)は1つのみ使用してください。レポートの言語は、[レポート]メニューの[オプション]を選択して変更することができます。

次のダイアログは、指定した条件に対して塗りつぶしのみを使用し、赤色を選択するスタイルの設定を表しています。



ゼロの値を持つオブジェクトを出力する場合、そのオブジェクトは赤色で出力されます。さらに、より複雑な例として条件をもう1つ追加します。在庫数が10未満になつたら黄色で出力するようにします。これには、条件エディターを開き、[追加] ボタンをクリックします。2番目の条件は次のようになります。

Value < 10

複数の条件が指定された場合、FastReportでは最初の条件から順にチェックされます。ある条件と一致するとFastReportはその条件に応じたスタイル設定をオブジェクトに適用し、処理を停止します。このため、条件を適切な順序で指定することが重要です。次の例は正しい順序です。

1. Value == 0  
2. Value < 10

条件の順序を入れ替えた場合、強調表示は誤って動作します。

1. Value < 10  
2. Value == 0

この例は、値が0の場合、最初の条件と一致してしまうため、"Value==0"が実行されません。条件の順序を変更するには および ボタンを使用します。

## ゼロの値を非表示にする

テキストオブジェクトには、ゼロの値を非表示にするために使用する HideZeros プロパティがあります。次の内容が含まれるオブジェクトについて見てみましょう。

総要素数 {CountOfElements}

変数 CountOfElements の値が0、かつ HideZeros プロパティに True が設定された場合、オブジェクトは次のように出力されます。

総要素数 :

また、テキストオブジェクトには、指定した値と式の値が等しい場合にその式の値を非表示にする HideValue プロパティがあります。たとえば、このプロパティの値が0 の場合、ゼロのフィールドはすべて非表示になります。このプロパティを使用すると、ゼロの日付を非表示にすることもできます。一般に、これは "1.1.0001" または "1.1.1900" のような日付です。この場合、HideValue プロパティの値は次のようにする必要があります。

1.1.1900 0:00:00

上記のとおり、日付だけではなく時刻も示す必要があります。.Net では日付の値に時刻も含まれるため、この指定は必要です。

**重要:** このメカニズムは、コントロールパネルで設定できるシステムの「地域」の設定に依存します。これは FastReport が ToString() メソッドを使用して文字列を比較するからです。このメソッドは式の値を文字列に変換します。これに関連し、異なる地域設定を持つコンピューター上で実行できるレポートの作成時にも注意してください。

さらに、テキストオブジェクトの NullValue プロパティを使用すれば、ヌル値ではなくテキストを出力することもできます。多くの場合、これはヌル値ではなくダッシュを出力するために使用されます。次の内容が含まれるオブジェクトについて見てみましょう。

総要素数 {CountOfElements}

変数 CountOfElements の値がヌル、かつ NullValue プロパティに -- が設定された場合、オブジェクトは次のように出力されます。

総要素数 :-

## 重複する値を非表示にする

テキストオブジェクトには、重複する値の出力方法を制御できる Duplicates プロパティがあります。このプロパティは、テキストオブジェクトがデータバンド上にある場合に使用できます。データ行で同一の値が近くに出力される場合、それらの値は重複していると見なされます。

Duplicates プロパティには次のいずれかの値を設定できます。

- Show - 重複する値を持つオブジェクトをそのまま表示します(既定値)。
- Hide - 重複する値を持つオブジェクトを非表示にします。
- Clear - 重複する値はクリアしますが、オブジェクト自身は表示します。
- Merge - 重複する値を持つオブジェクトを結合します。

これらのモードの違いを以下の図で示します。



## データの偶数行を強調表示する

レポートを見やすくするために、データの偶数行を別の色で強調表示することができます。これはバンドまたはそのバンド上のオブジェクトの EvenStyle プロパティを使用して設定できます。このプロパティには、データ行の強調表示に使用されるスタイル名を指定します。

バンドではなくオブジェクトの EvenStyle プロパティを使用することをお勧めします。これにより、レポートのエクスポート時に発生する可能性がある問題が回避されます。

強調表示を設定するには、以下の手順に従います。

1. 行の強調表示に使用するスタイルを定義します。これは[レポート]メニューの[スタイル]を選択して行います。
2. その定義した新しいスタイル名を、バンドまたはそのオブジェクトの EvenStyle プロパティに指定します。

既定では、オブジェクトが使用するのは EvenStyle プロパティに指定されたスタイルの塗りつぶし属性のみです。この動作は EvenStylePriority プロパティで定義されます。このプロパティの既定値は "UseFill" です。残りのスタイルパラメーターも使用する必要がある場合は、このプロパティに "UseAll" を設定します。

この技術を使用したレポートは次のようになります。

Product name	Unit price
Chai	1,800
Chang	1,900
Chartreuse verte	1,800
Côte de Blaye	26,350
Guaraná Fantástica	450
Ipoh Coffee	4,600

## 1 つのデータ バンドを持つレポート

このタイプのレポートは頻繁に使用されます。これはデータソースの行をリスト出力することができます。たとえば、顧客リストがこれに該当します。

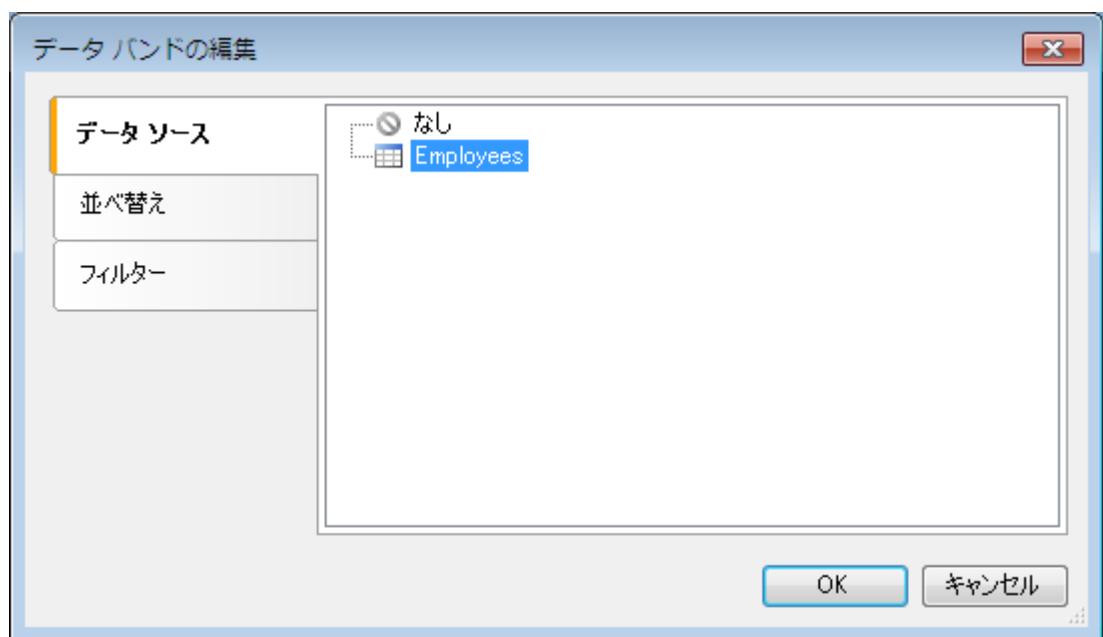
### バンドをデータ ソースへ接続する

データソースのデータを出力するには、データソースへ接続されるデータバンドが必要です。データソースに存在する行数と同数のデータバンドが出力されます。

データバンドがデータソースに接続されてない場合は 1 回 (1 行)だけ出力されます。

新しいレポートを作成する場合、そのレポートにはあらかじめデータバンドを含めいくつかの空のバンドが含まれています。バンドの追加は、[レポート]メニューの [バンドの構成] を選択して表示される [バンドの構成] ウィンドウで行うともできます。

バンドをデータへ接続するには、そのバンドをダブルクリックします。エディター ウィンドウで次のようにデータソースを選択し、[OK] をクリックします。



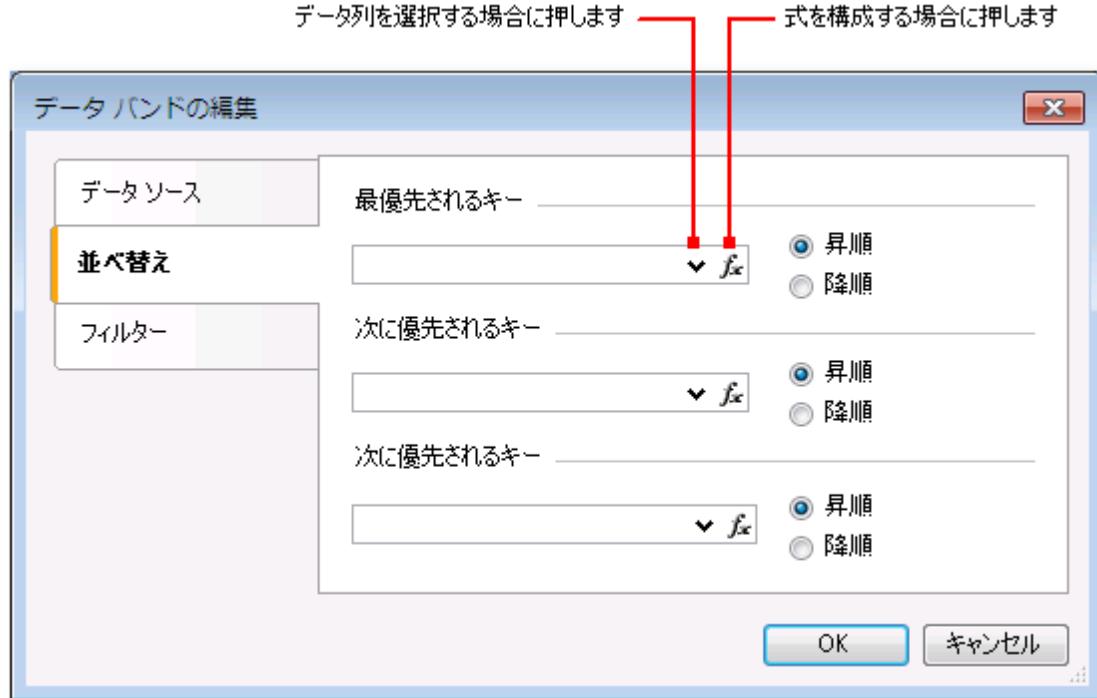
### テキストの出力

バンドをデータソースへ接続したら、データ列の情報を表示するテキストオブジェクトをバンド上に配置することができます。これを最も速く行うには、[データ] ウィンドウからバンド上へデータ列をドラッグアンドドロップします。テキストオブジェクトの詳細については、[テキストオブジェクト](#)を参照してください。

## データの並べ替え

既定では、データバンドはデータを自然な順序で出力します。出力する前に、データの並び替えが必要となることがあります。たとえば、顧客の一覧をアルファベット順に並べ替えることによって、わかりやすく提示することができます。

データバンドエディターでは並べ替えを制御できます。このエディターを呼び出すには、バンド上でオブジェクト以外の領域をダブルクリックします。

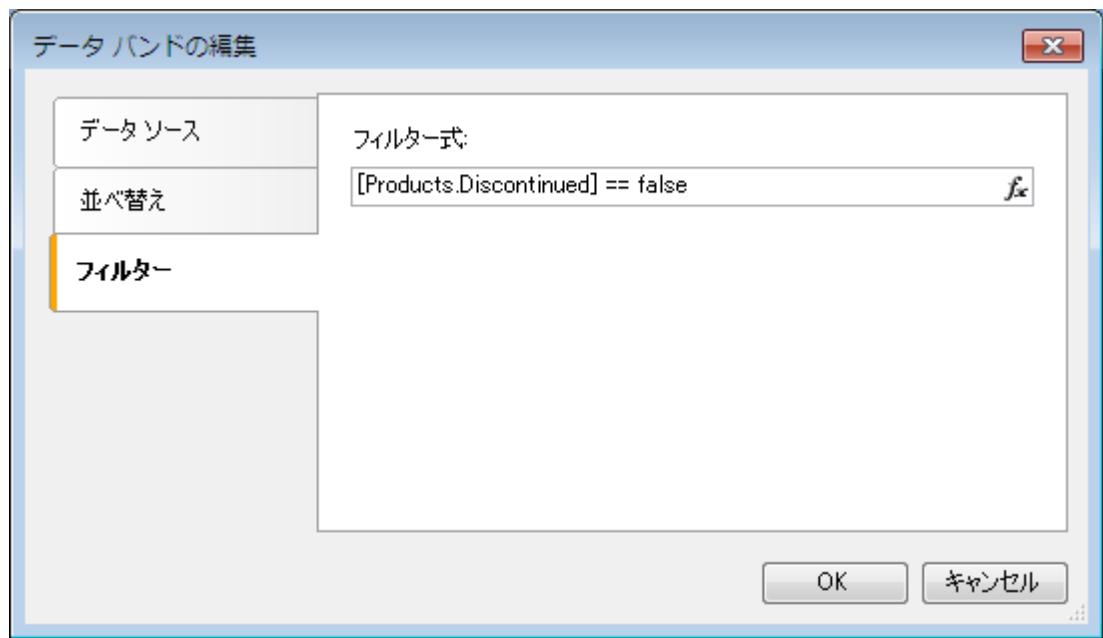


並べ替え条件として、データ列または式を使用できます。複数(3つ以下)の並べ替え条件を指定することができます。これは、たとえば顧客の一覧を最初に「市」で並べ替え、次に「顧客名」で並べ替えるという場合に必要です。条件ごとに、並べ替えの順序(昇順または降順)を選択できます。

別 の方法でデータを並べ替える場合 - SQL クエリをデータソースとして使用します。SQL クエリはデータサーバー上で実行され、並べ替えられた行が返されます。

## データのフィルタリング

データバンドに output する行をフィルター処理するには、そのバンドのエディターを呼び出して [フィルター] タブを表示します。



フィルター式として適切な式を指定できます。式の詳細については [式](#) を参照してください。

上図の例では次のフィルターが使用されます。

```
[Products.Discontinued] == false
```

これは Disconnected フラグが false であるデータ行がすべて選択されることを意味します。

次のように複雑なフィルター条件を使用することができます。

```
[Products.Discontinued] == false && [Products.UnitPrice] < 10
```

これは Disconnected フラグが false であり かつ 価格 (UnitPrice) が 10 未満であるデータ行が選択されることを意味します。

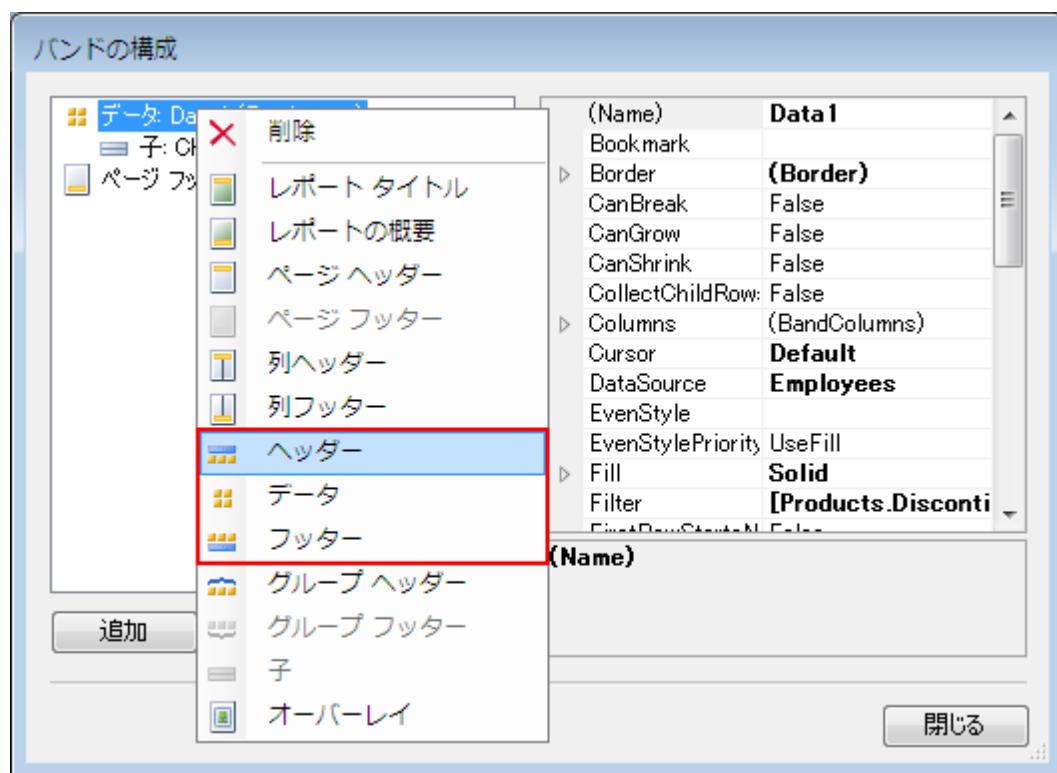
このフィルター処理方法は、データソースにすべての行が含まれ、その一部が除外されることを想定しています。データソースに大量の行が含まれる場合は、レポートの実行速度が著しく低下する可能性があります。この場合は、SQL クエリをデータソースとして使用し、そこから必要なフィルター処理を実行することができます。SQL クエリはデータサーバー上で実行され、レポートで必要な行のみを返します。

ダイアログ フォームを使用してデータのフィルター処理を実行することもできます。詳細については [ダイアログ フォーム](#) を参照してください。

## データのヘッダーおよびフッター

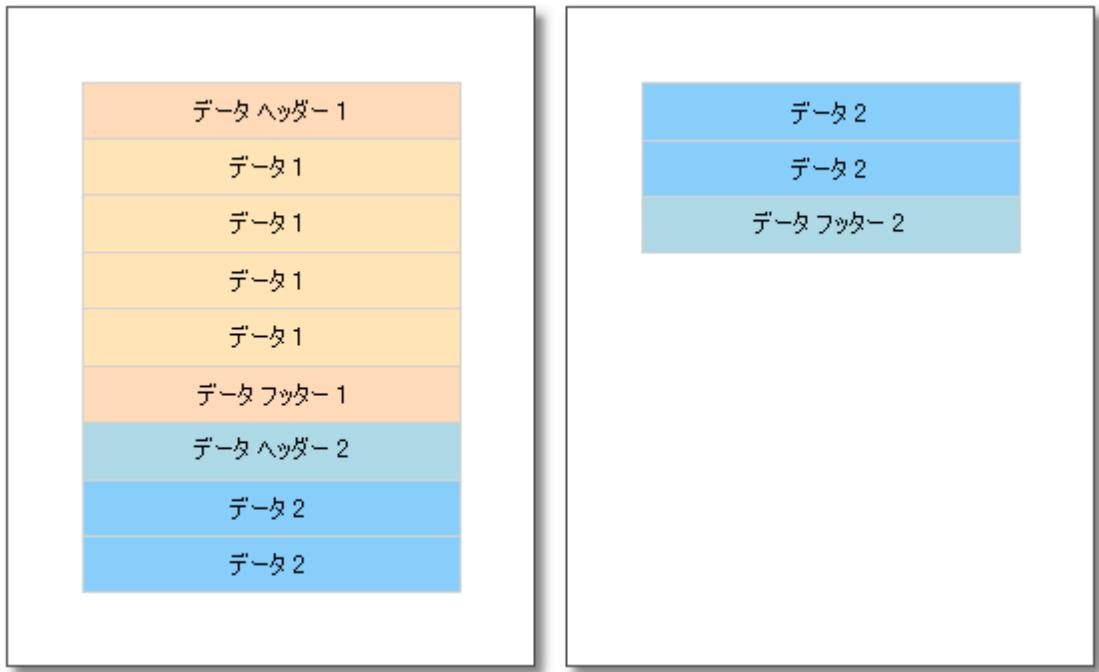
データバンドにはヘッダーとフッターを含めることができます。ヘッダーはデータの前に1回出力され、フッターはすべてのデータが出力された後に出力されます。

データバンドにヘッダーとフッターを追加するには、[レポート]メニューの [バンドの構成] を選択します。[バンドの構成] ウィンドウで、バンドの一覧から "データ" を選択して右クリックします。コンテキストメニューから "ヘッダー" または "フッター" を選択します。

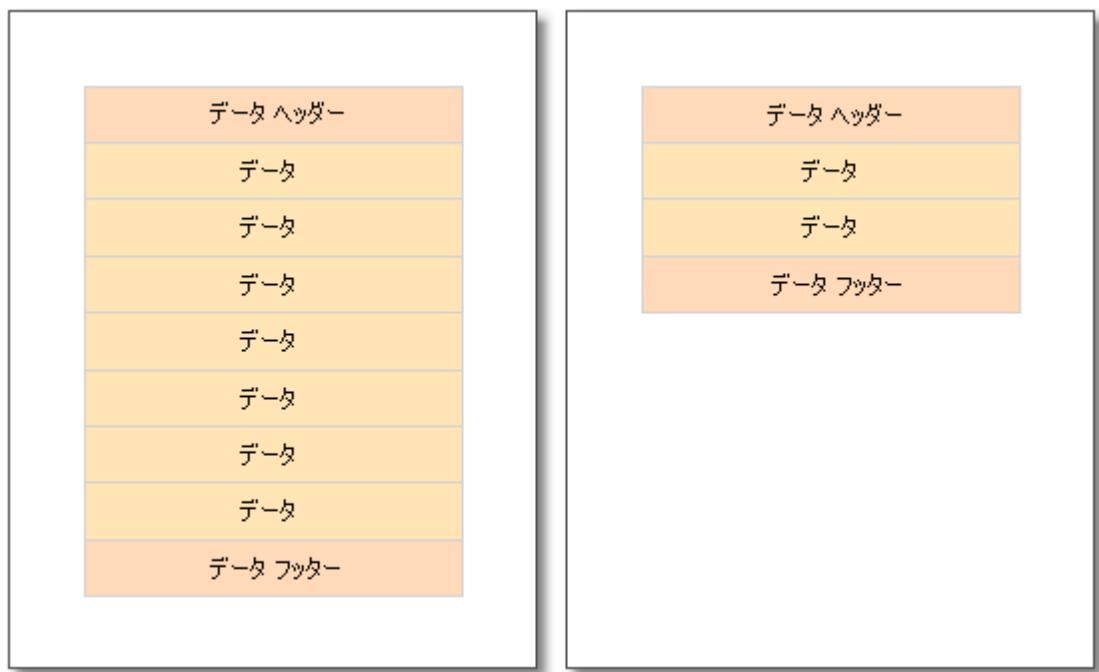


これらのバンドは以下のようないくつかの状況で役立ちます。

- 1ページに複数のリストを出力する場合 ([マスター/マスター レポート](#))。この場合、各データバンドがヘッダーおよびフッターを個別に持つます。



- 1 つのリストを出力するときに、そのリストが実行レポートの1 ページ内に収まらない場合。データのヘッダー/フッターバンドの RepeatOnEveryPage (ページごとに繰り返す) プロパティを使用すると、レポートの各ページにそのヘッダー/フッターバンドを出力することができます。



## データのブレークおよび同一ページに出力

このセクションでは、データ出力時の2つのモード"ブレーク"および"同一ページに出力"について説明します。

標準のバンド出力モードの場合、FastReportは現在のページに1つのバンドをまるごと出力できる十分なスペースがあるかどうかをチェックします。十分なスペースがない場合は、そのバンド全体が次のページに出力されます。バンドのCanBreakプロパティを有効(True)になると、FastReportは出力可能な領域にバンドの内容の一部を出力しようとします。つまり、バンドをブレークします。

バンドのブレークは、成功する場合もあれば、失敗する場合もあります。これは、バンド上に置かれているオブジェクトやその設定によります。以下のオブジェクトはブレークすることができます。

- テキスト
- リッチテキスト
- 表

これらのオブジェクトにはCanBreakプロパティがあります。このプロパティを有効にすると、そのオブジェクトはブレークすることができます。ブレークできないオブジェクトは、そのオブジェクト全体を表示する十分なスペースがある場所に表示されます。

次の図で、バンドのブレーク方法を示します。



ブレークアルゴリズムは常に正しく動作するとは限りません。バンド上にある複数のオブジェクトがそれぞれ異なるフォントサイズを使用している場合は、不正な結果をもたらす可能性があります。

バンドを"ブレーク"する目的は、出力シートのスペースを節約することです。逆に"同一ページに出力"の目的は、1つの出力シートにバンドセット全体を表示することです。この場合、シート上に多くの未使用スペースが存在する可能性がありますが、出力されるデータは見やすくなります。

"同一ページに出力"メカニズムを使用すると、一連のバンドをまとめて1ページ(または列 - レポートに列がある場合)に出力することができます。出力時にデータがページの最後まで到達すると、FastReportは既に新しいページに出力されているすべてのデータを移動させます。

以下の場合は"同一ページに出力"を使用できます。

- データバンドのすべての行をまとめて出力する

- グループの全要素 (ヘッダー、データ フッター)をまとめて出力する
- (マスター/詳細レポートの場合)マスター データソースの行と明細行を一緒にまとめて出力する
- データヘッダーを 少なくとも1 つのデータ行と一緒にまとめて出力する
- レポートの概要またはデータフッターを 少なくとも1 つのデータ行と一緒にまとめて出力する
- 親バンドと子バンドと一緒に出力する

"同一ページに出力" メカニズムの使用について説明します。

すべてのデータ行またはグループ要素 (ヘッダー、データ フッター)をまとめて同一ページに出力するには KeepTogether プロパティを有効 (True) にします。このプロパティは データバンドおよびグループヘッダーバンドで使用されます。次の図では データに"同一ページに出力" を適用した場合と 適用しない場合の出力方法を示します。



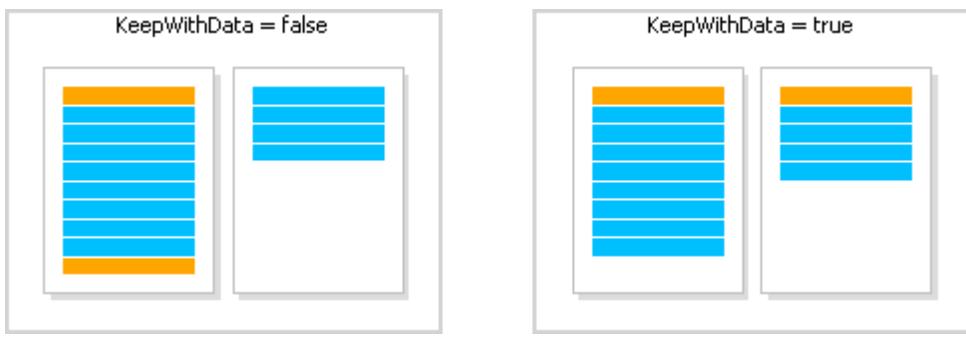
マスター データ行と詳細データ行と一緒にまとめて1 ページに出力するには データバンドの KeepDetail プロパティを有効 (True) にします。このプロパティは マスター/詳細タイプのレポートで使用されます。



ヘッダーおよびフッターの"ぶら下げ" を避けるには KeepWithData プロパティを使用します。以下のバンドにはこのプロパティがあります。

- データヘッダー
- データフッター
- グループヘッダー
- グループフッター
- レポートの概要

このプロパティを使用すると ヘッダー/フッターを少なくとも1 つのデータ行と結びつけます。



バンドとその子バンドを同一ページに出力させる場合は、KeepChild プロパティを有効 (True) にします。

## 空のデータ行の出力

既定の出力フォーム上にデータ行を出力する場合、一定数のデータ行が outputされることが想定するでしょう。実際のデータが必要なデータ行よりも少ない場合、空のデータ行を出力する必要があります。これを行うには、データバンドへ追加できる子バンドを使用します。

子バンドにはCompleteToNRows (最大 N データ行まで満たす) プロパティがあります。このプロパティに 0 より大きい値を設定すると、そのバンドを使用して、プロパティに指定された数までデータ行を追加します。たとえば、15 行を出力する必要があるが、データソースには 8 行しかなかったとします。この場合、子バンドは 7 回 (7 行分) 出力されます。



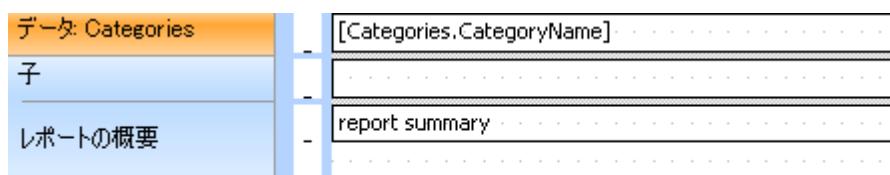
実行されたレポートは次のようになります。

1	Beverages
2	Condiments
3	Confections
4	Dairy Products
5	Grains/Cereals
6	Meat/Poultry
7	Produce
8	Seafood
9	
10	
11	
12	
13	
14	
15	

データソースに CompleteToNRows (最大 N データ行まで満たす) に指定した数より多くの行がある場合、空の行は出力されません。

ページの空き領域を埋めるもう別の方法もあります。この場合、子バンドはデータのフッターやクリープフッターやいずれかのバンドに追加され、ページ上の空き領域を埋めます。フッターバンドはページの最下部に出力されます。

このように空の行を出力するには、子バンドをフッターバンドに追加し、そのバンドの FillUnusedSpace (未使用の空間を埋める) プロパティを有効 (True) にします。これで、その子バンドは、追加しているバンドの上に表示されるようになります。下図では、子バンドはレポートの概要バンドに追加されています。



このレポートを実行すると、次のように表示されます。



## "データなし" テキストの出力

データバンドが空のデータソースが接続された場合、そのバンドは出力されません。場合によっては、単なる空のページではなく"データなし"のよなテキストを出力する必要があります。これは次の手順で行います。

- データバンドへ子バンドを追加します。
- その子バンドのPrintIfDataIsEmpty プロパティにTrue を設定します（[プロパティ]ウインドウで設定できます）。
- 子バンドにテキストオブジェクトを置き、表示するデータはありません」というテキストを入力します。



このレポートは、次のように出力されます。

- データソースにデータ行がある場合は、データバンドおよびそれに関連するすべてのバンド（データのヘッダー/フッター）が表示されます。
- データソースが空の場合は、"表示するデータはありません" というテキストを表示する子バンドのみ出力されます。

## 階層の出力

データバンドでは、階層化リストを出力することができます。この出力には、1つのバンドと1つのデータソースが使用されます。階層は、データソースで次の2つのデータ列を用いて定義されている必要があります。

1. キー列 :これはデータ行の識別子です。
2. 列 :この項目の親のキーが含まれる列です。

このようなソースを階層形式で出力するには、以下に示すデータバンドのプロパティを設定する必要があります。これは[プロパティ]ウインドウで設定できます。

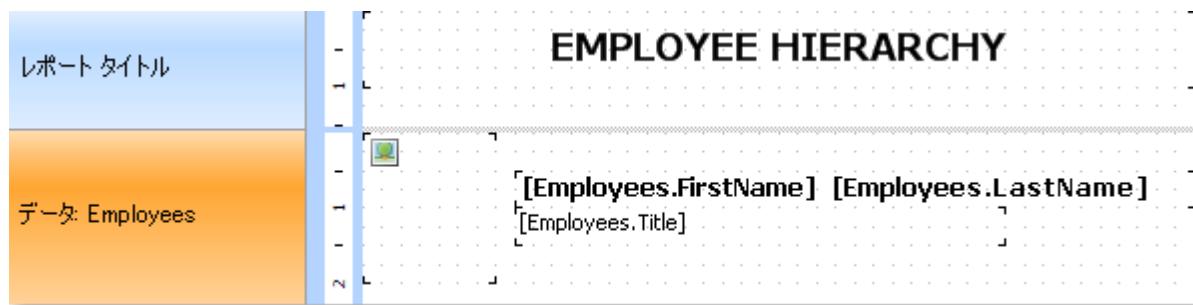
階層	
IdColumn	
Indent	1 cm
ParentIdColumn	

- IdColumn プロパティにキー列を指定する
- ParentIdColumn プロパティに親の値が含まれる列を指定する
- Indent プロパティに階層のインデントを指定する

Employees デモテーブルから階層形式で従業員を出力する方法の例を見てみましょう。このテーブルには、次のような2つの列があります。

- EmployeeID 列 :キーであり、従業員 ID を含みます。
- ReportsTo 列 :親となる従業員の ID を含みます。

次のようなレポートを作成します。

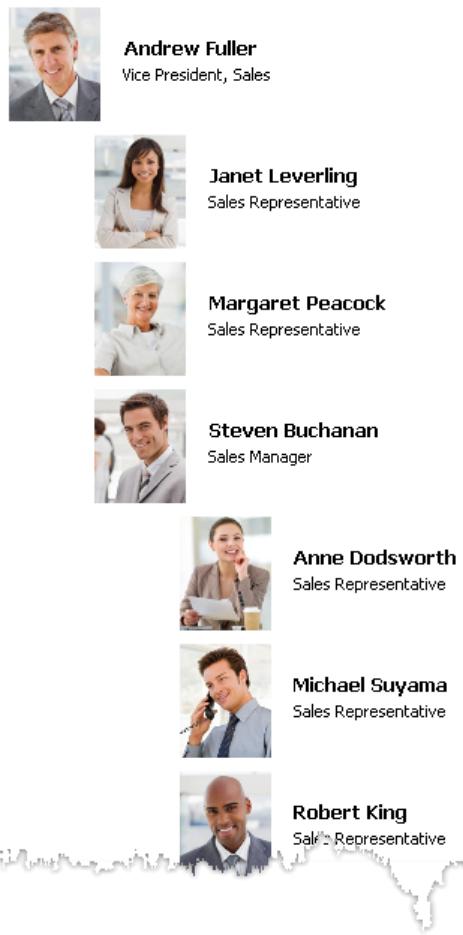


データバンドのプロパティを以下のように設定します。これらは階層に関するプロパティです。

階層	
IdColumn	Employees.EmployeeID
Indent	1.5 cm
ParentIdColumn	Employees.ReportsTo

レポートを実行すると、次のように表示されます。

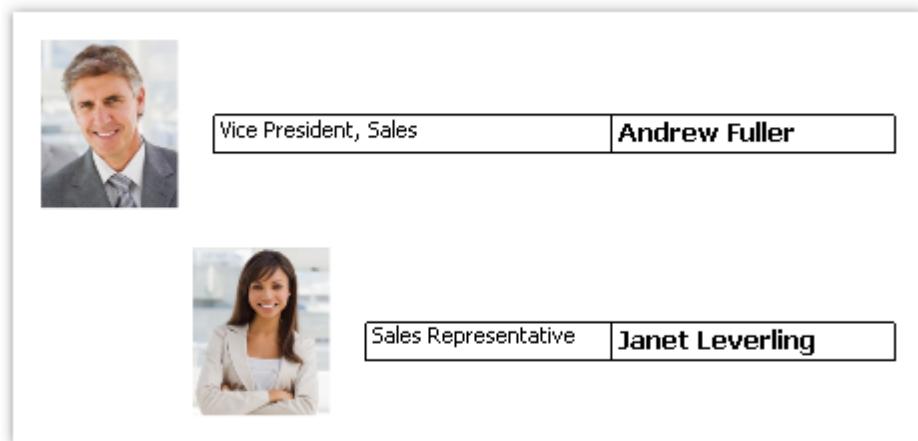
## EMPLOYEE HIERARCHY



階層を出力する場合、FastReport はバンドを (Indent プロパティで指定された値に従って) 右へシフトし、またその同じ値分だけバンド幅を狭めます。これによりバンドのオブジェクトの Anchor プロパティを使用できます。この場合、このプロパティに指定可能な値は次のとおりです。

- Top, Left (既定値) - オブジェクトはバンドと共に移動します。
- Top, Right - オブジェクトは元の位置から移動しません。
- Top, Left, Right - オブジェクトの右側は元の位置から移動しません。オブジェクトの左側はバンドと共に移動します。

この設定を使用すると有用な効果を得ることができます。



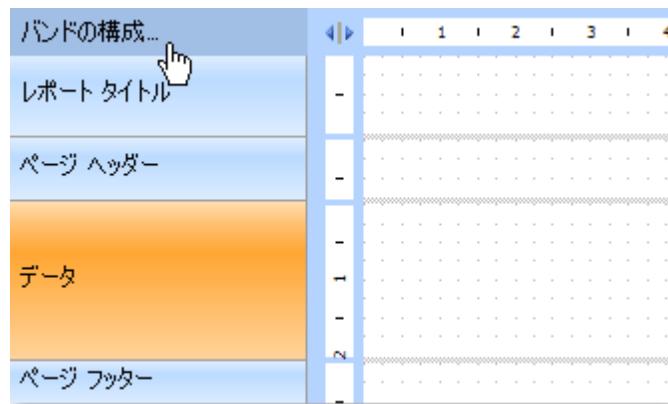
この例では、図オブジェクトのAnchorプロパティには"Left, Top"、役職(Title)用オブジェクトのAnchorプロパティには"Top, Left, Right"、氏名(FastName/LastName)用オブジェクトのAnchorプロパティには"Top, Right"が設定されています。

## マスター/詳細レポート

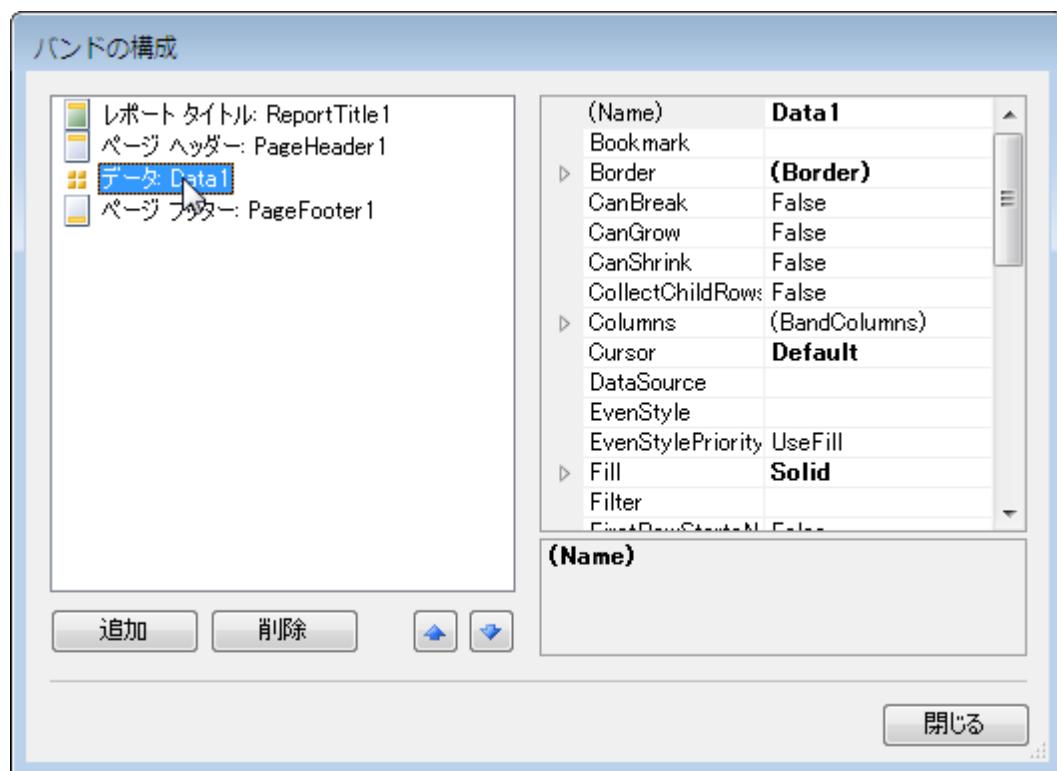
2つのデータバンドを使用することで、マスター/詳細タイプのレポートを簡単に作成することができます。このレポートでは、関連性がある2つのデータソースを使用します。マスター用データソースにある1つの行を、詳細用データソースの複数の行と対応付けることができます。関係の詳細については、[「データ」の章の「関係」](#)を参照してください。

レポートには1つのデータバンドを置く必要があります。このバンドをマスター バンドとし、その内部に詳細バンドが含まれるようにします。これは、[レポート]メニューの[バンドの構成]を選択して呼び出すことができる[バンドの構成]ウィンドウで行います。

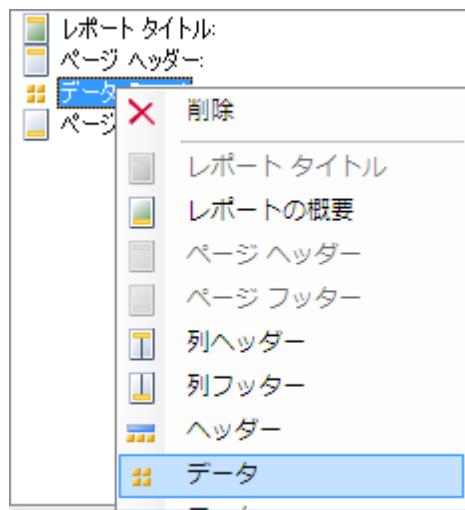
マスター/詳細レポートを一から作成する手順を見てみましょう。まず、レポートデザイナーを実行し、新しい空白レポートを作成します。新しいレポートにはあらじめ1つのデータバンドが含まれています。



詳細データバンドを追加するには、[バンドの構成]ウィンドウを呼び出します。これは上の図で示すように[バンドの構成]ボタンをクリックするか、あるいは[レポート]メニューの[バンドの構成]を選択して呼び出します。このウィンドウではバンドの構成が表示されます。



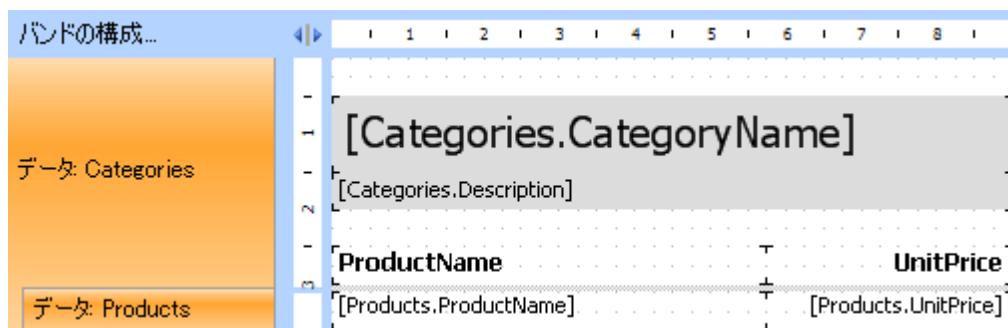
上の図で示すように「データ バンドを選択して右クリックし、コンテキストメニューを表示します（またはウインドウの下部にある「追加」ボタンをクリックします）。メニューから「データ バンドを選択します。



これで、選択したバンドに入れ子のデータバンドが追加されます。閉じるボタンをクリックしてウインドウを閉じます。次のように変更されたレポートテンプレートが表示されます。



入れ子のデータバンドは、ウインドウの左側に現れるバンド構成列でわかりやすく表示されます。次に、バンドを対応するデータソースは接続し、そのバンド上にデータ列を配置する必要があります。ここでは、FastReport に付属のデモデータベースからCategories およびProducts という2つのデータソースを使用します。



レポートを実行すると 次のように出力されます。

## Beverages

Soft drinks, coffees, teas, beers, and ales

ProductName	UnitPrice
Chai	1,800
Chang	1,900
Guaraná Fantástica	450
Sasquatch Ale	1,400
Steeleye Stout	1,800
Côte de Blaye	26,350
Chartreuse verte	1,800
Ipoh Coffee	4,600
Laughing Lumberjack Lager	1,400
Outback Lager	1,500
Rhönbräu Klosterbier	775
Lakkalikööri	1,800

## Condiments

Sweet and savory sauces, relishes, spreads, and seasonings

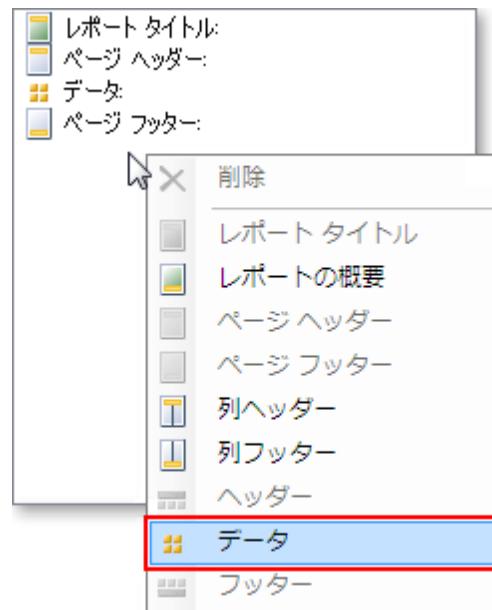
ProductName	UnitPrice
Aniseed Syrup	1,000
Chef Anton's Cajun Seasoning	2,200
Chef Anton's Gumbo Mix	2,135

このように、入れ子 (レベルは無制限) のデータによるマスター/詳細 レポートを作成することができます (マスター/詳細/サブ詳細...など)。マスター/詳細 レポートタイプの作成に使用する方法がもう1つあります。これは入れ子のレポートの使用が関係しています。入れ子のレポートについては、「レポートの作成」の章の[サブレポート](#)で説明しています。

## マスター/マスター レポート

レポートページには、複数の単一リストを出力することができます。これは、ページ上に2つ以上のデータバンドを置くことで出力可能です。バンドが互いに入れ子になり、関連するソースからデータを出力するようなマスター/詳細レポートとは異なり、このタイプのレポートは1つのデータソースも互いに依存関係がありません。

Categories テーブルおよびCustomers テーブルによる2つのリストを1ページに出力するレポートの作成方法を例に説明します。レポートを新規作成し、必要なデータソースを追加します。2つ目のデータバンドを追加するには、[ バンドの構成 ] ウィンドウを呼び出します。



下図で示すように、バンド一覧の空白領域を右クリックして、コンテキストメニューから「データ バンド」を選択します。これで、独立したデータバンドが新たに作成されます。レポートテンプレートは次のようになります。



これで、バンドをデータソースに接続し、複数のデータ列をそのバンド上に置くことができます。

データ: Categories	[Categories.CategoryName]
データ: Customers	[Customers.CompanyName]

レポートを実行すると、次のように出力されます。

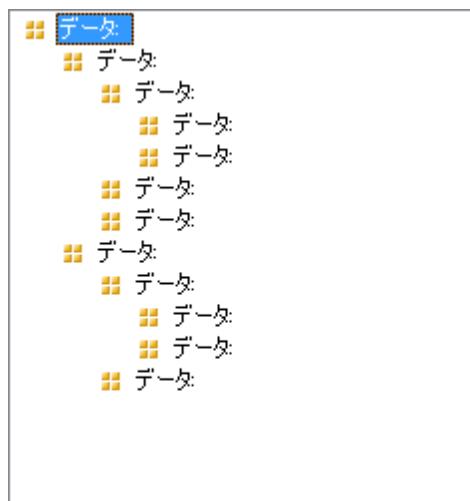
Beverages  
Condiments  
Confections  
Dairy Products  
Grains/Cereals  
Meat/Poultry  
Produce  
Seafood  
Alfreds Futterkiste  
Ana Trujillo Emparedados y helados  
Antonio Moreno Taquería  
Around the Horn

## マスター/ 詳細/ 詳細レポート

データバンドには1つまたは複数の入れ子のデータバンドを含めることができます。これにより マスター/詳細/詳細レポートを作成することができます。これを行うには、[バンドの構成] ウィンドウを開き、マスターのデータバンドを右クリックし、詳細のデータバンドを追加します。この手順を繰り返して詳細バンドを追加していきます。



このように、マスター データバンドには詳細バンドを無制限に追加することができます。レポート構築の例は次のようにになります（これは1つの例で、FastReport の能力を具体的に示した過ぎません）。



## 複数列のレポート

一般的なレポートでは、ページの末尾までデータが出力されます。出力がページの末尾に達すると、次のページが新たに形成され、出力を続行します。レポートに列を用いると、データを複数の列に出力することができます。データの出力がページの末尾に達すると、後続データは同じページ内で新しい列に出力されます。この意味では、一般的なレポートは1つの列を持つレポートと見ることができます。

FastReport では列の出力方法が2つあります。

## ページの段組み

複数列のレポートを出力する1つの方法は、レポートページへ段組みを設定することです。この設定は [ページ設定] ウィンドウの [段組み] タブで行います。



以下の段組みパラメーターを設定できます。

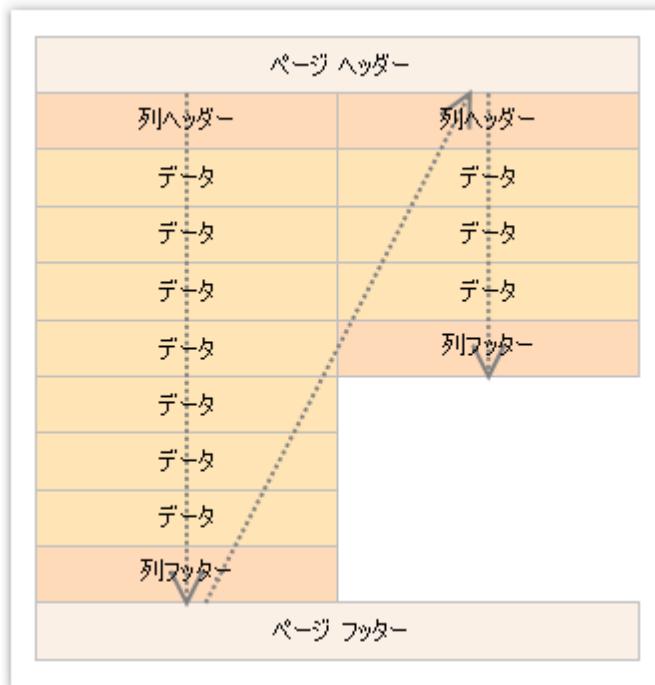
- 段数
- 段の幅
- 各段の位置

一般的なレポートを段組みのレポートに変換するには、このタブの段数を設定するだけです。それ以外のパラメーターは FastReport が独自に計算します。段組みを有効にすると、デザイナーのバンドのモードは次のように変更されます。



灰色で示される領域にはオブジェクトを配置することはできません。これは次の段のオブジェクトを出力するために使用されます。

段組みを扱う場合は、列ヘッダーおよび列フッター バンドが使用されます。バンド名からわかるように、これらはそれぞれ各段の先頭と末尾に 出力されます。次の図は段組み設定のレポートにおけるバンドの出力順序を示しています。



## データ バンドの列

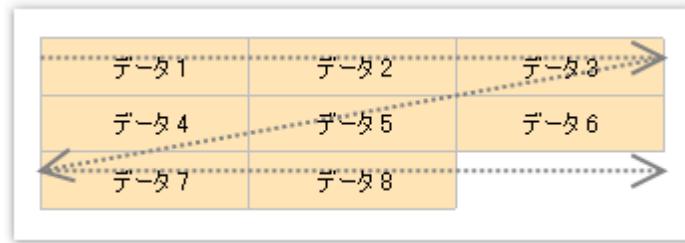
複数列のレポートを出力するもう1つの方法は、データバンドの列を使用することです。それ以外のバンドは1つの列で出力されます。

列のパラメーターは、[プロパティ]ウインドウで利用可能な **Columns** プロパティを使用して設定できます。以下のパラメーターを設定できます。

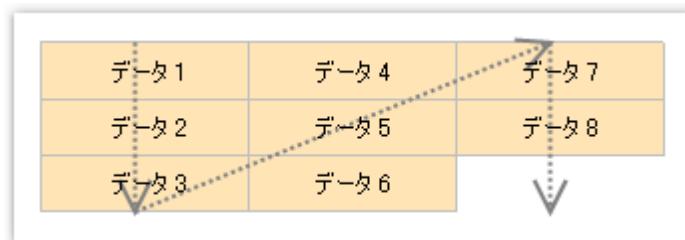
- 列数
- 列の幅
- 列の出力モード。 **AcrossThenDown** および **DownThenAcross** という2つのモードからいずれかを選択できます。

- 1列内の最小行数 - DownThenAcross モードを選択した場合

列のバンドは2つのモードのいずれかで出力できます。AcrossThenDown モード(既定)の場合、列は次のように出力されます。



DownThenAcross モードの場合、列は次のように出力されます。



このモードでは、FastReport が1列内のデータ行数を計算し、列が等しく配置されるようにします。Columns.MinRowCount プロパティを使用して、1列内の最小行数を設定することもできます。

## 小冊子タイプのレポート

小冊子フォームのレポートを出力する場合は、以下のことを考慮する必要があります。

- レポートページの区別 - 表紙、目次、レポート内容、裏表紙
- 偶数ページと奇数ページで異なるページ余白
- 偶数ページと奇数ページで異なるヘッダー/フッター

## レポートへのページの追加

レポートテンプレートはページをいくつでも追加することができます。ページごとに異なるレポートを設定することもできます。新しいページを

追加するには、ツールバー上の  ボタンをクリックします。または、[新しい項目の追加] ボタンをクリックし、[新しい項目の追加] ウィンドウで "新しいレポートページ" 項目を選択して追加することもできます。

目次を作成する場合は、対話型レポートセクションの例 4: 目次、ナビゲーションおよびアウトライン付きのレポートで説明している技術を使用できます。

## ページ設定

[ページ設定] ウィンドウでは、ページ構成を見開きページに設定するオプションがあります。



ページを奇数ページから開始する必要がある場合は、StartOnOddPage プロパティに True を設定します。FastReport は必要に応じて、指定されたページの出力を開始する前に空のページを出力します。

## 奇数/偶数ページ上の印刷

すべてのレポートオブジェクトにPrintOn プロパティがあります。このプロパティを使用すると、奇数ページと偶数ページに異なるオブジェクトを出力できます。

このプロパティは[プロパティ]ウインドウで設定できます。

オブジェクトを出力するページを決定します。このプロパティは以下の値のいずれか、または値の組み合わせを持ちます。

- FirstPage
- LastPage
- OddPages
- EvenPages
- RepeatedBand (この値は RepeatOnEveryPage プロパティがTrue に設定されているバンドを対象とします)

このプロパティの既定値は、"FirstPage, LastPage, OddPages, EvenPages, RepeatedBand" です。これは、そのオブジェクトがレポートの全ページに出力されることを意味します。以下に、このプロパティの典型的な使用例をいくつか挙げます。

プロパティの値	オブジェクトが出力される場所
FirstPage	先頭ページのみ
LastPage, OddPages, EvenPages, RepeatedBand	先頭ページ以外の全ページ
FirstPage, OddPages, EvenPages, RepeatedBand	最終ページ以外の全ページ
RepeatedBand	RepeatOnEveryPage プロパティがTrue に設定されているバンドのみ
FirstPage, LastPage, OddPages, EvenPages	繰り返されるバンド以外の全バンド
FirstPage, LastPage, OddPages, RepeatedBand	奇数ページのみ
FirstPage, LastPage, EvenPages, RepeatedBand	偶数ページのみ

たとえば、奇数ページと偶数ページで異なるテキストを出力する場合は、バンド上に2つのテキストオブジェクトを置き、各オブジェクトを次のように設定します。

- 1つ目のオブジェクトは奇数ページに出力します。このオブジェクトのPrintOn プロパティには"FirstPage, LastPage, OddPages, RepeatedBand" (つまりEvenPages 以外の値すべて)を設定します。

- 2つ目のオブジェクトは偶数ページに出力します。このオブジェクトのPrintOn プロパティには"FirstPage, LastPage, EvenPages, RepeatedBand" (つまりOddPages 以外の値すべて)を設定します。

これらのオブジェクトは 同時に出力されることはありません。これらのオブジェクトは奇数および偶数ページの上部に交互に置かれます。

すべてのバンドにも同じプロパティがあります。奇数ページと偶数ページに異なるバンドを出力するには Child バンドを使用します。子バンドはどのバンドにも追加できます。この追加は [バンドの構成] ウィンドウで行えます。以下のようにしてメインのバンドとその子バンドを設定します。

- メインのバンドは奇数ページに出力します。このオブジェクトのPrintOn プロパティには"FirstPage, LastPage, OddPages, RepeatedBand" (つまりEvenPages 以外の値すべて)を設定します。
- 子バンドは偶数ページに出力します。このオブジェクトのPrintOn プロパティには"FirstPage, LastPage, EvenPages, RepeatedBand" (つまりOddPages 以外の値すべて)を設定します。

バンドはそれぞれ異なる高さ 外観および内容を持ちます。次の図は 異なるページヘッダーを出力する例です。



## グループ化と集計

先のセクションでは、2つの関連するデータソースからデータを出力するマスター/詳細レポートタイプについて説明しました。FastReportでは、同様の方法でレポートを作成することができます。ただし、使用するデータソースは1つです。この場合はグループ化が使用されます。

1つのグループは、グループヘッダー、データ、およびグループフッタの3つのバンドでセットになります。これはデザイナーで次のように表示されます。



グループには必ずヘッダーとデータが含まれます。グループフッターは任意で削除することができます。

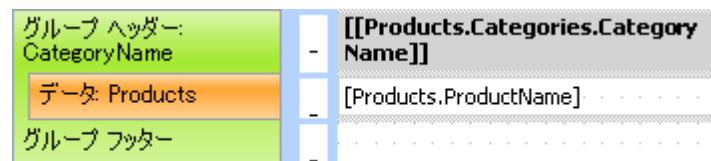
グループを使用するには、グループヘッダーに対してグループ条件を設定し、データソースをデータバンドへ接続します。条件にはどのような式も指定できますが、一般的にはデータソースの列の1つが設定されます。グループの出力は次のように行われます。

1. グループヘッダーが出力される
2. データ行が出力される
3. グループ条件が変更されたかどうかをチェックする
4. 条件が変わらなければ、次のデータ行が出力される(手順2)
5. 条件が変わったら、グループフッターを出力し、新しいグループを出力する(手順1)

以下のようなデータを持つProductsテーブルがあるとします。

CategoryName	ProductName
Beverages	Côte de Blaye
Beverages	Chartreuse verte
Beverages	Steeleye Stout
Beverages	Guaraná Fantástica
Beverages	Sasquatch Ale
Beverages	Rhönbräu Klosterbier
Beverages	Lakkalikööri
Beverages	Outback Lager
Beverages	Ipoh Coffee
Beverages	Laughing Lumberjack Lager
Beverages	Chang
Beverages	Chai
Condiments	Original Frankfurter grüne Soße
Condiments	Sirop d'érable
Condiments	Chef Anton's Gumbo Mix
Condiments	Northwoods Cranberry Sauce
Condiments	Grandma's Boysenberry Spread
Condiments	Chef Anton's Cajun Seasoning
Condiments	Aniseed Syrup
Condiments	Louisiana Hot Spiced Okra
Condiments	Vegie-spread
Condiments	Louisiana Fiery Hot Pepper Sauce
Condiments	Gula Malacca
Condiments	Genen Shouyu

データはCategoryName列でグループ化できます。この列はグループヘッダーに出力されます。データ自身はProductNameフィールドによって表します。レポートデザインは次のようにになります。



レポートを実行すると、次のように表示されます。

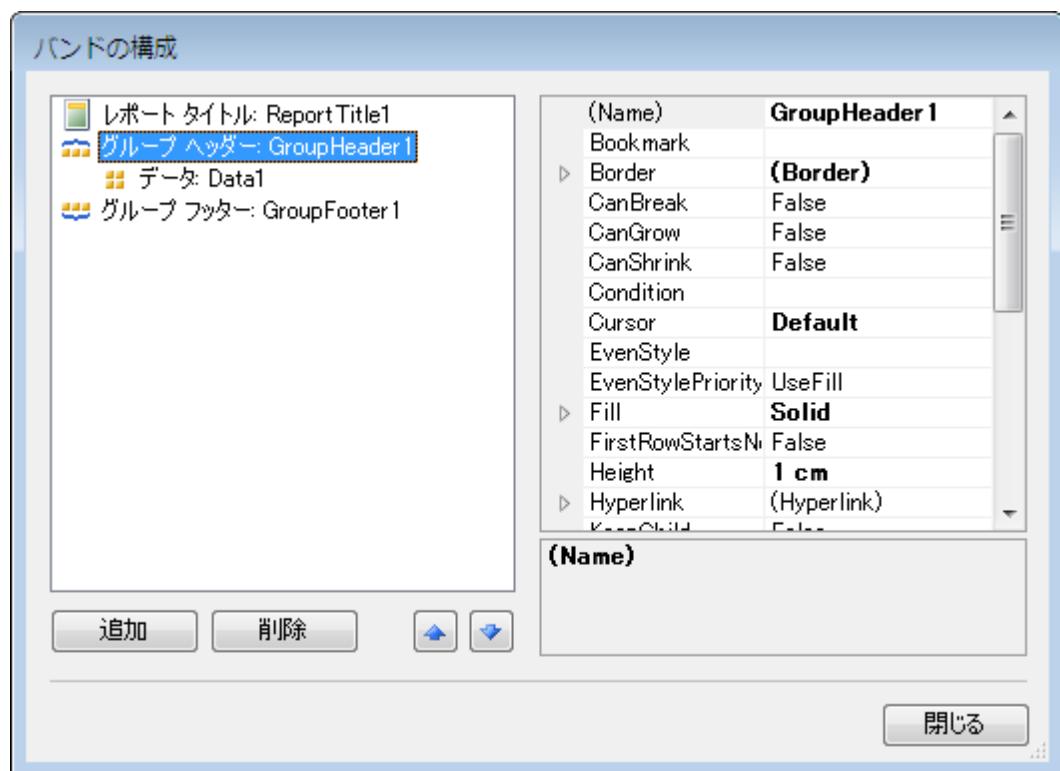
Beverages
Côte de Blaye
Chartreuse verte
Steeleye Stout
Guaraná Fantástica
Sasquatch Ale
Rhönbräu Klosterbier
Lakkalikööri
Outback Lager
Ipoh Coffee
Laughing Lumberjack Lager
Chang
Chai

Condiments
Original Frankfurter grüne Soße
Sirop d'érable
Chef Anton's Gumbo Mix
Northwoods Cranberry Sauce
Grandma's Boysenberry Spread
Chef Anton's Cajun Seasoning
Aniseed Syrup
Louisiana Hot Spiced Okra
Vegie-spread
Louisiana Fiery Hot Pepper Sauce
Gula Malacca
Genen Shouyu

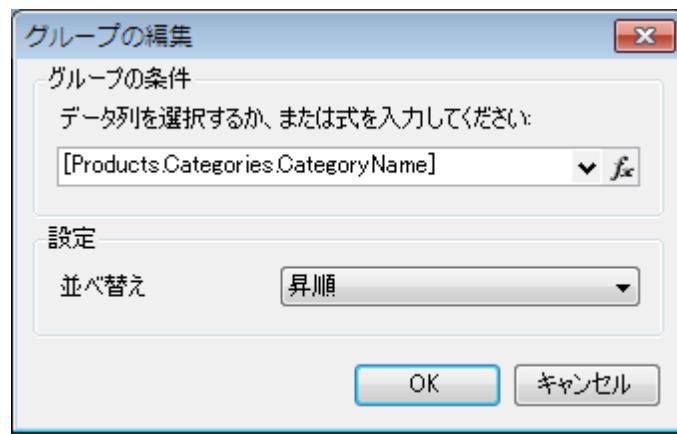
## グループの作成

レポートにグループを追加するには2つの方法があります。

第1の方法：[バンドの構成] ウィンドウでグループヘッダー バンドを追加します。これを行うには、[追加] ボタンをクリックして"グループヘッダー" バンドを選択します。FastReport は対象となるデータバンドにグループを追加するか、または対象となるデータバンドがレポート内ない場合はグループ全体を作成します。

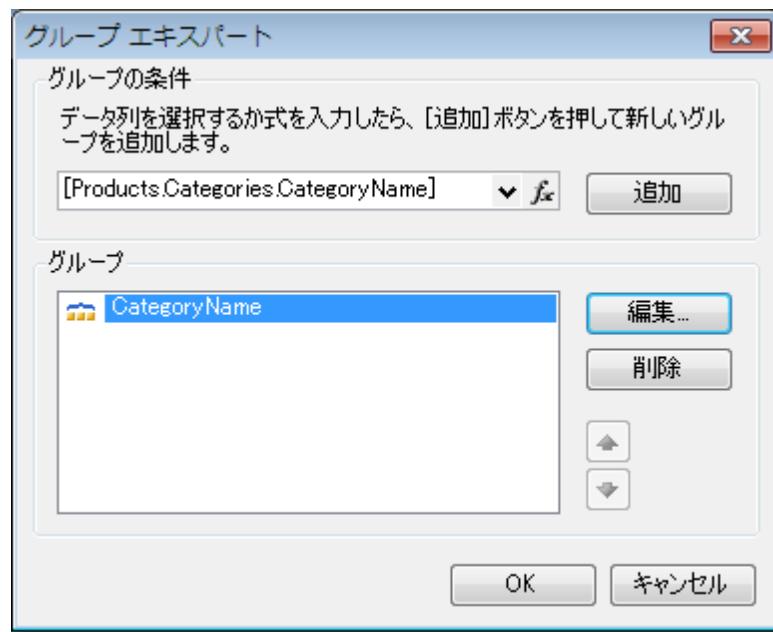


グループの設定を編集するには、グループヘッダー バンドをダブルクリックします。次のようなグループヘッダー エディターのウインドウが表示されます。

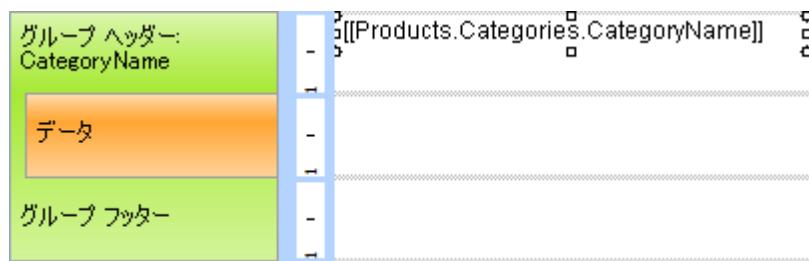


グループ条件を設定する必要があります。これには式またはデータソースの列を指定できます。並べ替えも選択します。既定では、データは昇順で並べ替えられます。

第 2 の方法 : [グループエキスパート] ウィザードを使用します。これは [レポート] メニューの [グループエキスパート] を選択して呼び出すことができます。グループを作成するには、グループ条件を入力し、[追加] ボタンをクリックします。



ウィザードによって、すべてのグループ要素がレポートへ追加されます。また、グループヘッダーにテキストオブジェクトが作成され、その中にグループ条件が表示されます。



## データの並べ替え

グループ化が正しく動作するには、グループ条件に使用されるデータソースが、グループ条件の対象となる列で並べ替えられている必要があります。この条件を満たしていないと、次のように同じグループのデータ行が分散して出力されてしまいます。

Beverages
Sasquatch Ale
Steeleye Stout

Seafood
Inlagd Sill
Gravad lax

Beverages
Côte de Blaye
Chartreuse verte

Seafood
Boston Crab Meat
Jack's New England Clam Chowder

データソースを並べ替える方法は2つあります。

- グレープエディターでデータの並べ替えを設定できます。データソースはグレープ条件に対して自動的に並べ替えられます。
- データバンドエディターで並べ替えを設定できます。

どちらの方法も同じですが、最初の方法を使用した方が便利です。グレープを作成するときに、データのグレープ化と並べ替えの設定が1つのダイアログで行えます。

場合によっては、最初の方法を使用しないようにして下さい。この方法は、製品名の先頭文字を対象にグレープ化を設定すること前提としています。そのような場合、製品は名前の先頭文字でのみ並べ替えられます。これは条件として適切ではありません。2番目の方法を使用して、製品のフレーム名に対して並べ替えを行うように指定してください。

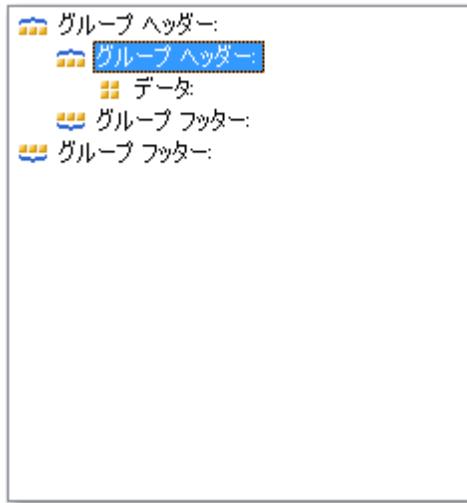
## 入れ子のグレープ

入れ子のグレープには複数のグレープヘッダー/バンドがあります。最後(最下層)のバンドにはデータバンドが含まれます。



各グループヘッダーがそれぞれ独自のグループ条件を持っています。

入れ子のグループは一般的なグループ作成と同様に作成することができます。[バンドの構成] ウィンドウで、シンプルなグループを作成し、入れ子のグループを追加します。このためには、既存のグループヘッダー バンドを選択し、[追加] ボタンをクリックしてもう1つのグループヘッダー バンドを追加します。



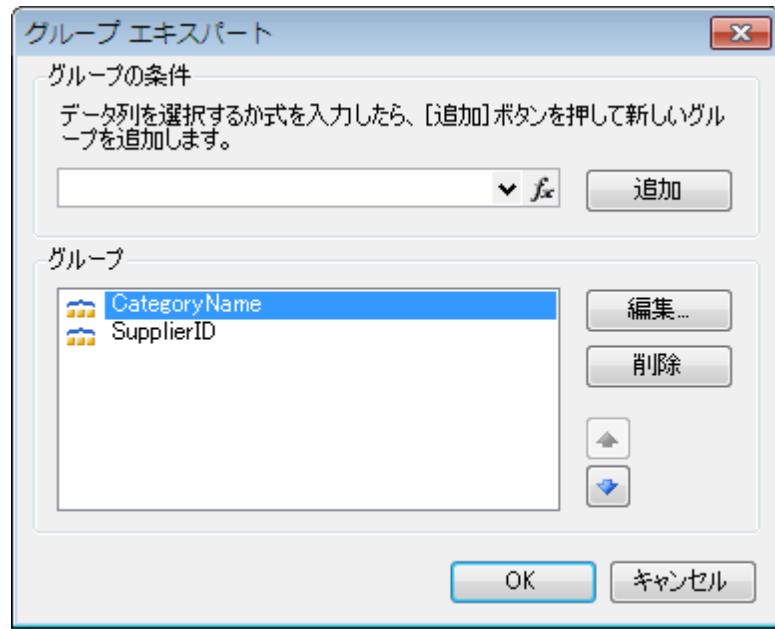
その後、追加したグループのエディターを呼び出してグループ条件を設定します。

別 の方法として、前のセクション(「グループの作成」)で紹介したグループエキスパートを使用することもできます。グループ条件を設定し、[追加] ボタンをクリックします。このウィザードによって、既存のグループは新しくグループが追加されます。

入れ子のグループの出力は一般的なグループの出力とあまり変わりません。データの出力時、FastReport は全グループのすべてのグループ条件をチェックします。グループ条件が変わったら、該当するグループを終了し、新しいグループの出力を開始します。

## グループの管理

グループを管理するには、グループエキスパートを使用できます。これは[レポート]メニューの[グループエキスパート]を選択して呼び出すことができます。



このウィザードを使用すると グループの追加や削除、およびグループ化の順序変更を行うことができます。グループの順序を変更する場合は ボタンと ボタンを使用します。[編集] ボタンを使用すると 選択したグループのグループ条件を変更することができます。

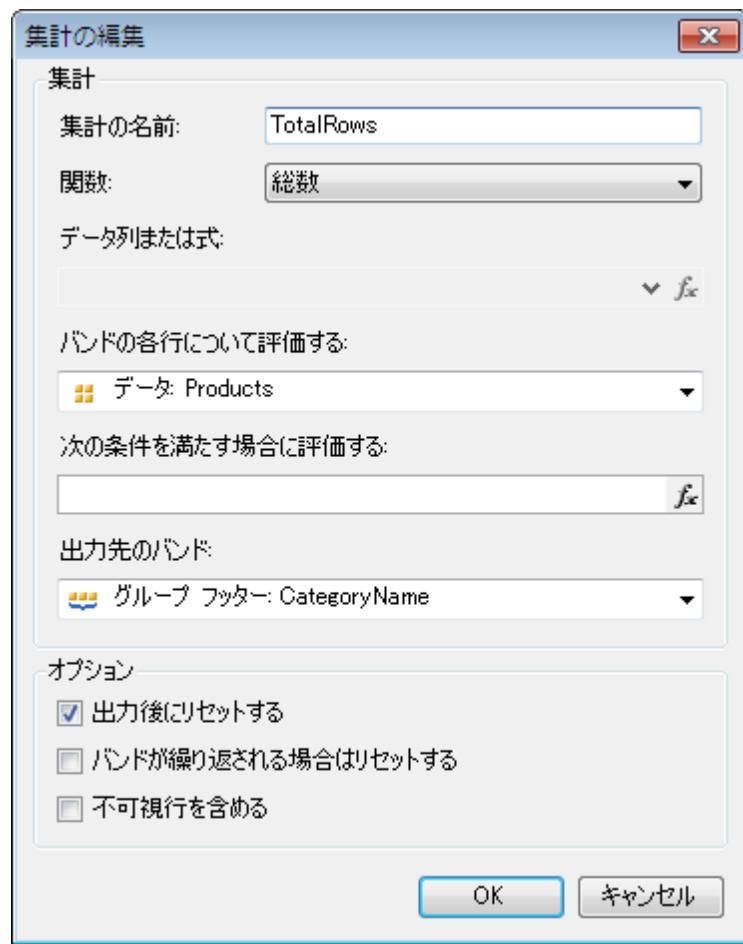
## 集計値の出力

多くの場合、グループ化はグループ単位の集計値を出力するために使用されます。たとえば グループ内の行数や1つのデータ列の合計などを出力できます。このような値を出力する場合に集計が使用されます。集計の使用の詳細については 「データ」の章の集計を参照してください。

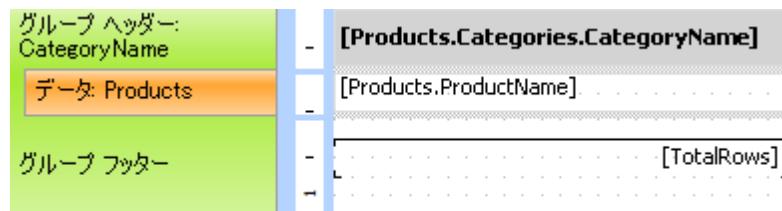
グループ単位で集計値を出力するには 以下のことを行う必要があります。

- 「データ」ウィンドウの「操作」|「新しい集計」を選択して集計を作成します。
- 「バンドの各行について評価する」コンボボックスで、グループのデータバンドを選択します。
- 「出力先のバンド」コンボボックスで、グループフッターを選択します。
- 集計値を出力するテキストオブジェクトをグループフッターに置きます。

たとえば 各グループの行数を出力する場合は、次のように集計を設定します。



集計値を表示する場合は、その集計項目を[データ]ウインドウからグループフッター上へドラッグします。



実行されたレポートは次のようになります。

Meat/Poultry
Alice Mutton
Perth Pasties
Thüringer Rostbratwurst
Pâté chinois
Tourtière
Mishi Kobe Niku

---

Produce
Rössle Sauerkraut
Uncle Bob's Organic Dried Pears
Manjimup Dried Apples
Longlife Tofu
Tofu

---

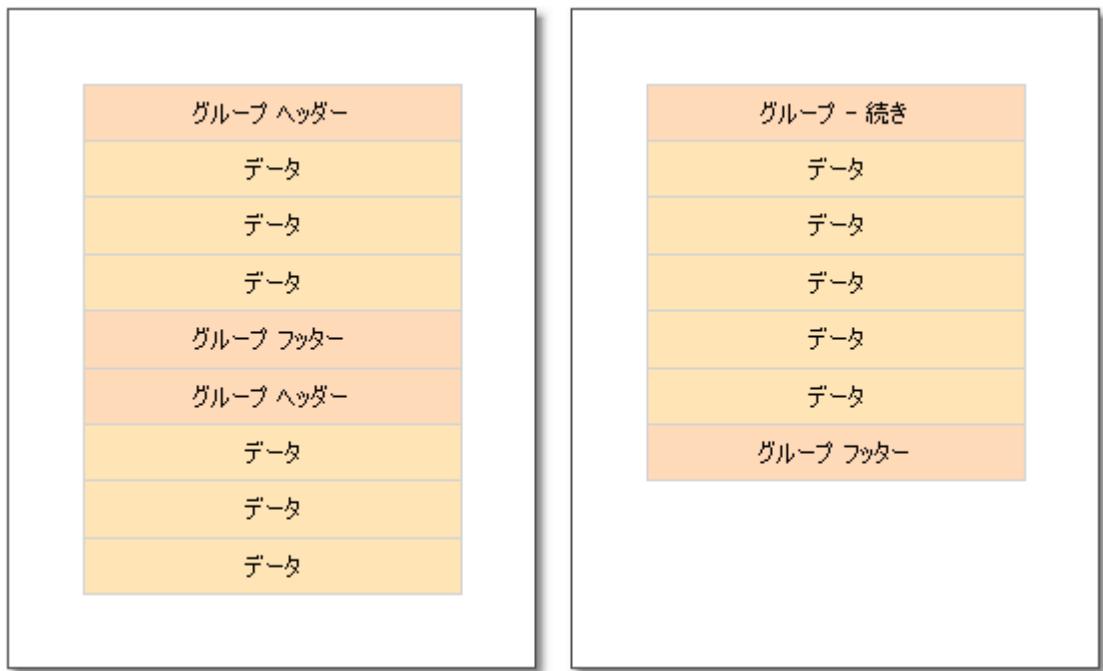
## ヘッダーとフッターの繰り返し

グループヘッダーおよびグループフッターには RepeatOnEveryPage プロパティがあります。これは、グループが実行レポートの 1 ページ内に収まらない場合に役立ちます。このプロパティを使用すれば、グループが 出力されるページごとにヘッダー/フッターを出力することができます。このようなヘッダー/フッターを出力する場合、FastReport はその Repeated フラグを設定します。これは、通常のグループヘッダーと繰り返し用グループヘッダーとで、異なるオブジェクトを出力させる場合に役立ちます (たとえば、新しいページでは "続き..." というテキストを出力する)。このため、テキストオブジェクトの PrintOn プロパティを使用します (詳細については [奇数/偶数ページ上の印刷](#) を参照)。

異なるテキストを出力するには、グループヘッダーに 2 つのオブジェクトを追加し、それらが上下に並ぶように配置します。

- 1 つ目のオブジェクトは通常のヘッダーに出力されます。このオブジェクトの PrintOn プロパティには "FirstPage, LastPage, OddPages, EvenPages" (つまり RepeatedBand 以外の値すべて) を設定します。
- 2 つ目のオブジェクトは繰り返されるヘッダーにのみ出力されます。このオブジェクトの PrintOn プロパティには "RepeatedBand" を設定します。"続き..." というテキストを持つテキストオブジェクトをヘッダーに追加します。

レポートは次のように出力されます。



グループフッターも各ページで繰り返すことができます。

Raclette Courdavault	5,500
Ravioli Angelo	1,950
Rhönbräu Klosterbier	775
Röd Kaviar	1,500
Røgede sild	950

Rössle Sauerkraut	4,560
<b>Total products: 6</b>	

このレポートでは、グループフッターに2つのオブジェクトがあり、上下に並ぶよう配置しています。



## グループのプロパティ

グループヘッダー バンドには便利なプロパティがあります。

StartNewPage プロパティを使用すると、そのグループを出力する前に新しいページを作成することができます。その結果、各グループが新しいページに出力されます。

ただし、最初に出力されるグループの前に新しいページは追加されません。これは、先頭ページが空になることを避けるためです。

ResetPageNumber プロパティを使用すると、新しいグループの出力時にページ番号をリセットします。通常、このプロパティは StartNewPage プロパティと共に使用します。結果として、両方のプロパティを有効 (True) にすると、各グループが新しいページに出力され、そのグループ単位でページ番号をカウントします。

## サブレポート

メインレポート内のある特定の場所に、追加データを表示する必要がある場合もあります。このデータは非常に複雑な構造を持つ別個のレポートを表す可能性もあります。FastReport の豊富なバンドコレクションを使用することで、このタスクの解決に臨むことができます。ただし、場合によってはサブレポートオブジェクトを使用することが望ましいこともあります。

サブレポートオブジェクトは普通のレポートオブジェクトであり、1つのバンド上に置くことができます。この場合、FastReport はレポートに別のページを追加し、そのページとサブレポートを接続します。このページで、任意の構造を持つ別のレポートを作成することができます。

サブレポートオブジェクトがあるレポートの出力時には、以下のことが行われます。

1. メインレポートが出力されますが、サブレポートオブジェクトは見えません。
2. サブレポートのバンドが出力されます。
3. メインレポートの出力が続行されます。

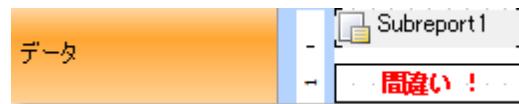
サブレポートはメインレポートのシート上に形成されるので、これにレポートタイトル、レポートの概要、ページヘッダー/フッター、列ヘッダー/フッター、オーバーレイバンドを含めることはできません。

## 出力モード

サブレポートには2つの出力モードがあります。

1つ目の出力モードでは、メインレポートのページにサブレポートのバンドとオブジェクトが表示されます。この場合、以下の制限があります。

- サブレポートオブジェクトはバンドの下端に置く必要があります。
- サブレポートオブジェクトより下にほかのオブジェクトを置かないようにしてください。レポートが実行されると、それらのオブジェクトはサブレポートのオブジェクトと重なってしまいます。



サブレポートの下にほかのオブジェクトを置く場合は、子バンドを使用してください。オブジェクトは以下のよう配置します。



2つ目の出力モードの場合は、サブレポートオブジェクトを含むバンド上にサブレポートのオブジェクトが表示されます。サブレポートオブジェクトのコンテキストメニューからこのモードを有効にすることができます。モードを有効にするには、メニューから「[親に]出力」を選択します。このモードではオブジェクトの配置について制限はありません。それだけでなく、このモードではサブレポートに表示されるデータ量に応じて親バンドを拡大または縮小させることができます。

2つ目の出力モードで唯一の問題点は サブレポートに大量のデータが存在する可能性があるということです。それによって出力時に親バンドの高さが大きくなります。このようなバンドを正しく出力するには その内容をブレークする必要があります (CanBreak プロパティ)。ブレークアルゴリズムの品質は完全なものではなく 場合によってはオブジェクトが移動してしまうこともあります。

## 横並び表示のサブレポート

2つのサブレポートオブジェクトを同じバンド上で横に並べて配置すると 2つの独立したデータリストを出力することができます。このようなレポートを出力する場合、FastReport は次の方法で動作します。

- サブレポートオブジェクトが現れない限り メインレポートを出力します。
- 最初のサブレポートを出力します。
- サブレポートの出力が開始したページに移動し 次のサブレポートを出力します。
- すべてのサブレポートが出力されたら 最も長いサブレポートが停止したところからメインレポートの出力を続行します。

## 入れ子のサブレポート

サブレポートのページ上で、さらに別のサブレポートオブジェクトを置くことができます。そのため、そのサブレポートは入れ子になります。入れ子のレベル数に明確な制限はありませんが、必要以上にレベルを設けるようなことはしないでください。入れ子のレベルが多すぎると、非常にわかりづらくなります。可能であれば、入れ子のデータの出力にデータバンドを使用してください。データバンドは1つまたは複数の入れ子のデータバンドを含めることができます。マスター/詳細またはマスター/詳細/サブ詳細タイプのレポートを出力しなければならない場合は、サブレポートを使用する必要はありません。

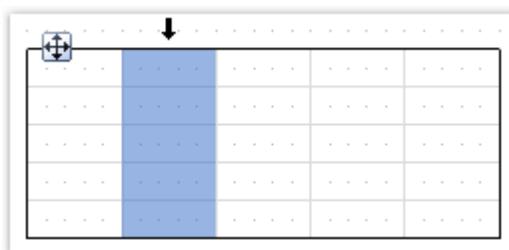
## 表タイプのレポート

表オブジェクトは行、列およびセルで構成されています。このオブジェクトはMicrosoft Excelの表を簡略化したものです。これは次のように表示されます。

## 列の設定

コンテキストメニューを使用して列を追加または削除することができます。これは次の手順で行います。

- 表全体または表内の要素を選択してから、作業対象の列の上部にカーソルを置きます。カーソルの形状が次のように黒い矢印に変わります。



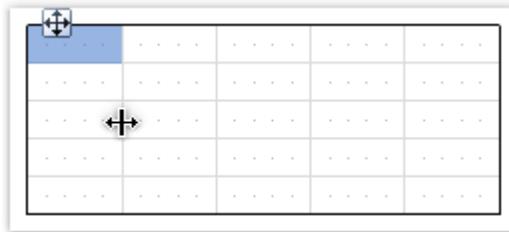
- 左クリックして列を選択します。
- 右クリックして列のコンテキストメニューを表示します。
- 隣接する複数の列を選択する場合は、左ボタンを押したままマウスを右または左へ移動させて列を選択していきます。

列のコンテキストメニューは、[レポートツリー]ウインドウ内からも呼び出すことができます。このウインドウを開き、作業対象の列を選択して右クリックします。

## 列のサイズの管理

列の幅は、以下のいずれかの方法を使用して設定できます。

- 表全体または表内の要素を選択してから、2つの列間の境界線上にカーソルを置きます。カーソルの形状が次のように左右分割に変わり、幅を変更できるようになります。



- 列を選択し、Width プロパティに適切な幅を指定します。このプロパティは [プロパティ] ウィンドウで設定できます。

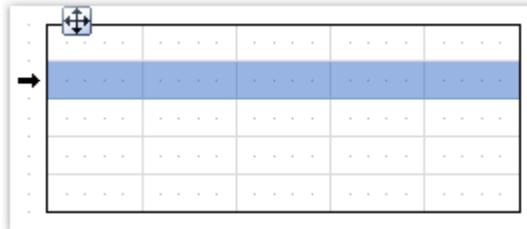
列のAutoSize (自動サイズ調整) プロパティも効果的にすることができます。これにより、レポートの実行時に列の幅が自動的に計算されます。列の幅を制限する場合は、MinWidth (幅の最小値) および MaxWidth (幅の最大値) プロパティで指定することができます。

列の幅はページ幅よりも大きくならないようにしておきましょう。

## 行の設定

行は列と同じ方法で設定されます。行を選択するには、以下の手順に従います。

- 表全体または表内の要素を選択してから、作業対象の行の左側にカーソルを置きます。カーソルの形状が次のように黒い矢印に変わります。



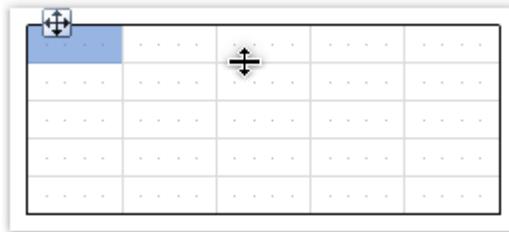
- 左クリックして行を選択します。
- 右クリックして行のコンテキストメニューを表示します。

隣接する複数の行を選択する場合は、左ボタンを押したままマウスを上または下へ移動させて行を選択していきます。

## 行のサイズの管理

行の高さは、以下のいずれかの方法を使用して設定できます。

- 表全体または表内の要素を選択してから、2つの行間の境界線上にカーソルを置きます。カーソルの形状が次のように上下分割に変わります。



マウスの左ボタンを押したまま上下に移動して行の高さを変更します。

- 行を選択し、Height プロパティに適切な高さを指定します。このプロパティは [プロパティ] ウィンドウで設定できます。

行のAutoSize (自動サイズ調整) プロパティも有効にすることができます。これにより、レポートの実行時に行の高さが自動的に計算されます。行の高さを制限する場合は、MinHeight (高さの最小値) および MaxHeight (高さの最大値) プロパティで指定することができます。

行の高さはページの高さよりも大きくならないようにしてください。

## セルの設定

セルはテキストのオブジェクトです。実質的に、セルクラスはテキストオブジェクトから継承されています。テキストオブジェクトについて説明されていることはすべて表のセルにも適用されます。

セルのテキストの編集は、テキストオブジェクトと同様に行うことができます。また、[データ] ウィンドウの要素をセルへドラッグアンドドロップすることができます。

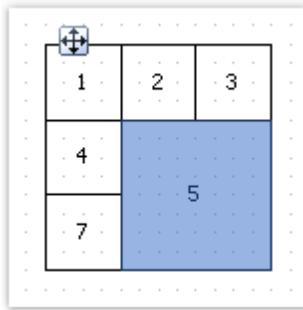
セルの罫線および塗りつぶしは、[罫線と塗りつぶし] ソールバーを使って設定できます。

セルのコンテキストメニューを呼び出すには、セルを右クリックします。

## セルの結合および分割

表内の隣接するセルは結合することができます。セルの結合により1つの大きなセルが形成されます。これは以下の手順で行います。

- マウスで最初のセルを選択します。
- 左ボタンを押したままマウスを移動させ、セルの一群 (結合対象の範囲) を選択します。
- 選択した範囲内で右クリックし、セルのコンテキストメニューを表示します。
- このコンテキストメニューから [セルの結合] を選択します。



セルを分割するには、そのセルのコンテキストメニューを表示し、[セル結合の解除]を選択します。

## セルにオブジェクトを挿入する

セル内には図などのオブジェクトを挿入して使用することができます。ただし、以下のオブジェクトはセルで使用することができません。

- 表
- 行列
- サブレポート

セルにオブジェクトを挿入するには、そのオブジェクトをセル内にドラッグします。オブジェクトはセル間で自由に移動させることができます。また、表の境界線を超えて戻すこともできます。



セルは、オブジェクトを置くためのコンテナとして機能します。これは、セル内でオブジェクトの Dock および Anchor プロパティが使用できるということです。これにより、セルのサイズが変更されたときにオブジェクトのサイズを変更することができます。

## 表の出力

表には2つの出力モードがあります。

1つ目のモードでは、表は自身が属するバンド内に表示され、デザイナーでの表示と同じように表示されます。このモードでは、表の幅がレポートページの幅よりも大きい場合でも、その表はページをまたいで分割されません。これは既定の出力モードです。

2つ目のモードは動的です。このモードでは、表はスクリプトを用いて作成されます。この場合、実行結果の表は初期の表とは異なる可能

性があります。これは FastReport の実行レポートとレポートテンプレートが異なるのと同じです。動的モードでは、表がレポートページ上に収まらない場合、その表はページをまたいで分割されます。

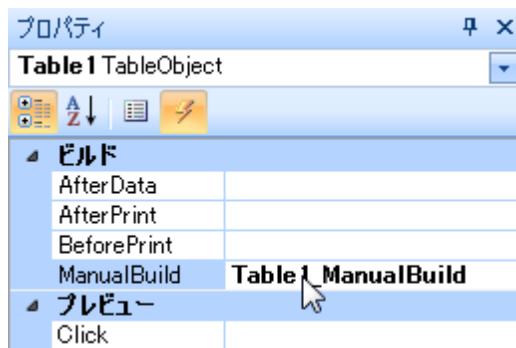
動的モードの場合、表は自身が置かれているバンドには出力されません。その代わりに、実行結果の表を含む一連のバンドを表自身が生成します。このように動作するモードには以下の制限事項があります。

- 表の下または表の近くに別のオブジェクトを置かないこと。その代わりに子バンドを使用してください。
- 1 つのバンドに 2 つのテーブルオブジェクトを置かないこと。

次に動的モードについて詳しく説明します。

このモードはプログラミングが伴うため、レポート開発者にはより高度な知識が必要となります。

表はスクリプトを使用して形成されます。スクリプトを作成するには、表オブジェクトを選択してから [プロパティ] ウィンドウでイベント用ボタンをクリックし、ManualBuild イベントをダブルクリックします。



これで、次のような空のイベントハンドラーがレポートコードに追加されます。

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
}
```

このモードでは、ソーステーブルがテンプレートとして使用されます。イベントコードで、ソーステーブルの行および列を必要なだけ出力することができます。この過程で、実行結果の表が形成されます。この表には行と列を無制限に含めることができます。このように表は、レポートページ上に収まらない場合、ページをまたいで分割されます。

表を出力する場合は、表オブジェクトの以下のメソッドが使用されます。

メソッド	パラメーター	説明
------	--------	----

PrintRow	int インデックス	指定されたインデックスを使用して行を出力します。行番号は0 から開始します。
PrintColumn	int インデックス	指定されたインデックスを使用して列を出力します。列番号は0 から開始します。
PrintRows	int[] インデックス	テーブルの複数の行を出力します。
PrintRows	-	テーブルのすべての行を出力します。
PrintColumns	int[] インデックス	テーブルの複数の列を出力します。
PrintColumns	-	テーブルのすべての列を出力します。
PageBreak	-	次の列または行を出力する前に、改ページを挿入します。

表は以下のいずれかの方法を使用して出力することができます。

第1の方法 - 上から下へ出力し、その後、左から右へ出力します。この方法は、行数が変動する表を出力する場合に適しています。メソッドは以下の順序で呼び出す必要があります。

- PrintRow(行インデックス)
- 指定の列を出力するため、1つまたは複数の PrintColumn(列インデックス) または PrintColumns(複数の列インデックス) を呼び出します。
- または、すべての列の出力に PrintColumns() を1回呼び出します。
- この順序を繰り返して、表の行を必要なだけ出力します。

表の各行には、同じ数の列を含める必要があります。PrintColumn(int インデックス) および PrintColumns(int[] 複数インデックス) メソッドを使用する場合は、このことを覚えておいてください。

第2の方法 - 左から右へ出力し、その後、上から下へ出力します。この方法は、列数が変動する表を出力する場合に適しています。メソッドは以下の順序で呼び出す必要があります。

- PrintColumn(列インデックス)
- 指定の行を出力するため、1つまたは複数の PrintRow(行インデックス) または PrintRows(複数の行インデックス) を呼び出します。
- または、すべての行の出力に PrintRows() を1回呼び出します。
- この順序を繰り返して、表の列を必要なだけ出力します。

表の各列には、同じ数の行を含める必要があります。PrintRow(int インデックス) および PrintRows(int[] 複数インデックス) メソッドを使用する場合は、このことを覚えておいてください。

出力用のメソッドの呼び出し順序を誤ると、レポートの実行時にエラーが発生します。たとえば、次のコードを使用して表を出力しようとすると、エラーになります。

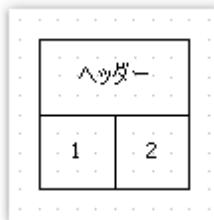
Table1.PrintRows();

```
Table1.PrintColumns();
```

これはメソッドの順序が間違っています。表の出力は、最初にPrintRow または PrintColumn メソッドのいずれかを用いて開始する必要があります。

## 複雑なヘッダーの出力

ここではスパンした(セルをまたいで広がった)セルを含むヘッダーについて説明します。スパンしたセルを持つ表の行または列を出力する場合、そのセルのサイズは自動的に大きくなります。次に例を挙げて説明します。



ManualBuild イベントハンドラーを作成します。このハンドラーでは1番目の列を3回、2番目の列を1回出力します。

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
    // 1行および列 0, 0, 0, 1 を出力する
    Table1.PrintRow(0);
    Table1.PrintColumn(0);
    Table1.PrintColumn(0);
    Table1.PrintColumn(0);
    Table1.PrintColumn(1);

    // 1行および列 0, 0, 0, 1 を出力する
    Table1.PrintRow(1);
    Table1.PrintColumn(0);
    Table1.PrintColumn(0);
    Table1.PrintColumn(0);
    Table1.PrintColumn(1);
}
```

各行には同数の列を出力していることを注目してください。この出力規則を守らないと予測できない結果を招きます。

このコードを実行すると、次のような結果になります。

ヘッダー			
1	1	1	2

ご覧のとおり ヘッダー セルは自動的に広げられます。次に、2 つの列グループを出力できる、もろ少し複雑なコードを作成してみましょう。

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
    // 行 0 および列 0, 0, 0, 1 の 2 つのグループを出力する
    Table1.PrintRow(0);
    // グループ1
    Table1.PrintColumn(0);
    Table1.PrintColumn(0);
    Table1.PrintColumn(0);
    Table1.PrintColumn(1);
    // グループ2
    Table1.PrintColumn(0);
    Table1.PrintColumn(0);
    Table1.PrintColumn(0);
    Table1.PrintColumn(1);

    // 行 1 および列 0, 0, 0, 1 の 2 つのグループを出力する
    Table1.PrintRow(1);
    // グループ1
    Table1.PrintColumn(0);
    Table1.PrintColumn(0);
    Table1.PrintColumn(0);
    Table1.PrintColumn(1);
    // グループ2
    Table1.PrintColumn(0);
    Table1.PrintColumn(0);
    Table1.PrintColumn(0);
    Table1.PrintColumn(1);
}
```

レポートを実行すると、次のような結果になります。

ヘッダー				ヘッダー			
1	1	1	2	1	1	1	2

2番目の列を出力する場合は次のコードが使用されます。

```
Table1.PrintColumn(1);
```

これで、ヘッダーが終了し、次のコードで1番目の列の出力が新しいヘッダーを開始します。

```
// グループ2
```

```
Table1.PrintColumn(0);
```

## 集計の使用

表オブジェクトの動的モードでは、以下の集計関数がサポートされます。

関数	パラメーター	説明
Sum (合計)	TableCell セル	セルに含まれる値の合計を返します。
Min (最小値)	TableCell セル	セルに含まれる値の中から最小値を返します。
Max (最大値)	TableCell セル	セルに含まれる値の中から最大値を返します。
Avg (平均)	TableCell セル	セルに含まれる値の平均値を返します。
Count (総数)	TableCell セル	指定されたセルが含まれる行数を返します。

通常の出力モード(非動的)の場合、これらの関数は動作しません。

集計関数を使用するには、表のセル内にその関数を置きます。たとえば、次の関数は"Cell1"という名前のセルに含まれる値の合計を計算します。

```
[Sum(Cell1)]
```

関数の実行時、(合計を計算している)現在のセルより上および左にあるすべての列が分析されます。

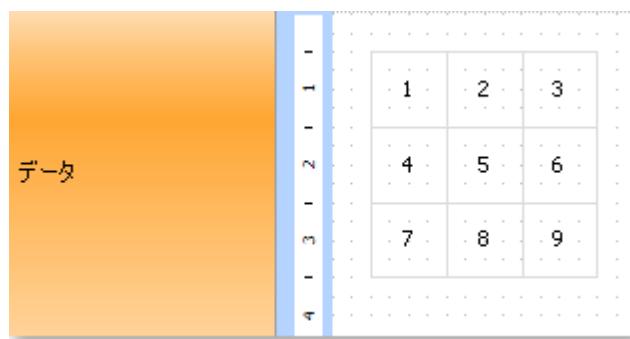
## 表のレイアウト

動的に作成される表は、自動的に複数ページに分割することができます。この動作は表のLayout プロパティで制御します。以下の値のいずれかを選択できます。

値	説明
AcrossThenDown	表は左から右方向へ表示されます
DownThenAcross	表は上から下方向へ表示されます。
Wrapped	幅が広い表は、折り返され同じページ上に表示されます。

## 例

表の出力例を見てみましょう。テンプレートとして、次のレポートを使用します。



始めに、表を選択して ManualBuild イベント用のイベントハンドラーを作成します。

### 例 1 表全体を上から下へ出力する

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
    // 行番号 0 の行およびその行のすべての列を出力する
    Table1.PrintRow(0);
    Table1.PrintColumns();

    // 行番号 1 の行およびその行のすべての列を出力する
    Table1.PrintRow(1);
    Table1.PrintColumns();

    // 行番号 2 の行およびその行のすべての列を出力する
    Table1.PrintRow(2);
    Table1.PrintColumns();
}
```

実行の結果、次のような表が出力されます。テンプレートとの相違はありません。

1	2	3
4	5	6
7	8	9

## 例 2 繰り返し行を含む表を上から下へ出力する

```

private void Table1_ManualBuild(object sender, EventArgs e)
{
    // 行番号 0 の行およびその行のすべての列を出力する
    Table1.PrintRow(0);
    Table1.PrintColumns();

    // 行番号 1 の行を3回およびその行のすべての列を出力する
    for (int i = 0; i < 3; i++)
    {
        Table1.PrintRow(1);
        Table1.PrintColumns();
    }

    // 行番号 2 の行およびその行のすべての列を出力する
    Table1.PrintRow(2);
    Table1.PrintColumns();
}

```

このコード例では、中段の行を3回出力しています。実行の結果は次のようになります。

1	2	3
4	5	6
4	5	6
4	5	6
7	8	9

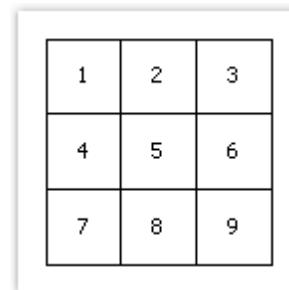
### 例 3 表全体を左から右へ出力する

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
    // 列番号 0 の列およびその列のすべての行を出力する
    Table1.PrintColumn(0);
    Table1.PrintRows();

    // 列番号 1 の列およびその列のすべての行を出力する
    Table1.PrintColumn(1);
    Table1.PrintRows();

    // 列番号 2 の列およびその列のすべての行を出力する
    Table1.PrintColumn(2);
    Table1.PrintRows();
}
```

実行の結果、次のような表が出力されます。テンプレートとの相違はありません。



1	2	3
4	5	6
7	8	9

### 例 4 繰り返し列を含む表を左から右へ出力する

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
    // 列番号 0 の列およびその列のすべての行を出力する
    Table1.PrintColumn(0);
    Table1.PrintRows();

    // 列番号 1 の列を3回およびその列のすべての行を出力する
    for (int i = 0; i < 3; i++)
    {
        Table1.PrintColumn(1);
        Table1.PrintRows();
    }

    // 列番号 2 の列およびその列のすべての行を出力する
    Table1.PrintColumn(2);
    Table1.PrintRows();
}
```

このコード例では、表の中央の列を3回出力しています。実行の結果は次のようにになります。

1	2	2	2	3
4	5	5	5	6
7	8	8	8	9

## 例 5 :繰り返し行/列を含む表を出力する

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
    // ----- 行番号 0 の行を出力する
    Table1.PrintRow(0);
    // 列番号 0 の列を出力する
    Table1.PrintColumn(0);
    // 列番号 1 の列を3回出力する
    for (int i = 0; i < 3; i++)
        Table1.PrintColumn(1);
    // 列番号 2 の列を出力する
    Table1.PrintColumn(2);

    // ----- 行番号 1 の行を3回出力する
    for (int j = 0; j < 3; j++)
    {
        Table1.PrintRow(1);
        // 列番号 0 の列を出力する
        Table1.PrintColumn(0);
        // 列番号 1 の列を3回出力する
        for (int i = 0; i < 3; i++)
            Table1.PrintColumn(1);
        // 列番号 2 の列を出力する
        Table1.PrintColumn(2);
    }

    // ----- 行番号 2 の行を出力する
    Table1.PrintRow(2);
    // 列番号 0 の列を出力する
    Table1.PrintColumn(0);
    // 列番号 1 の列を3回出力する
    for (int i = 0; i < 3; i++)
        Table1.PrintColumn(1);
    // 列番号 2 の列を出力する
    Table1.PrintColumn(2);
}
```

各行に同数の列が output されることを注目してください。この出力規則を守らないと予測できない結果を招きます。

このコード例では、表の中央に位置する行と列がそれぞれ 3 回ずつ出力されています。実行の結果は次のようになります。

1	2	2	2	3
4	5	5	5	6
4	5	5	5	6
4	5	5	5	6
7	8	8	8	9

## 例 6 :データ ソースの使用

今まで見てきた例では、一般的なテキストを含む表を出力しました。このコード例では、データソースを使用して表を形成する方法を示します。このため、次のようなフォームを持つ表を作成します。

Product Name	Unit Price	Units In Stock
[Products.ProductName]	[Products.UnitPrice]	[Products.UnitsInStock]
.....	.....	.....

ここでは ManualBuild イベントハンドラーを作成します。このハンドラーでは以下のことを行います。

- レポートに定義されるデータソースを取得する
- そのデータソースを初期化する（データで埋める）
- データソースに存在する行数と同数の行を出力する

このイベントハンドラーのコードは次のとおりです。

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
    // データソースを名前で取得する
    DataSourceBase rowData = Report.GetDataSource("Products");
    // データソースを初期化する
    rowData.Init();

    // 表のヘッダーを出力する
```

```

Table1.PrintRow(0);
Table1.PrintColumns();

// データソースの行をループする
while (rowData.HasMoreRows)
{
    // 表の行を出力する
    Table1.PrintRow(1);
    Table1.PrintColumns();

    // 次のデータ行を選択する
    rowData.Next();
}

// 表のフッターを出力する
Table1.PrintRow(2);
Table1.PrintColumns();
}

```

レポートを実行すると 次のように出力されます。

Product Name	Unit Price	Units In Stock
Chai	1,800	39
Chang	1,900	17
Aniseed Syrup	1,000	13
Chef Anton's Cajun Seasoning	2,200	53
Chef Anton's Gumbo Mix	2,135	0
Grandma's Boysenberry Spread	2,500	120
Uncle Bob's Organic Dried Pears	3,000	15
Northwoods Cranberry Sauce	4,000	6

## 例 7 :改ページの挿入

表オブジェクトのPageBreak メソッドを使用すると 表の出力時に改ページを挿入することができます。このメソッドは1 つの行/列を出力する前に呼び出します。

例 1 を使用してPageBreak メソッドの動作方法を示します。ここでは 3 行目を新しいページに出力します。

```

private void Table1_ManualBuild(object sender, EventArgs e)
{
    // 行番号 0 の行とその行のすべての列を出力する
    Table1.PrintRow(0);
    Table1.PrintColumns();

    // 行番号 1 の行とその行のすべての列を出力する
    Table1.PrintRow(1);
    Table1.PrintColumns();

    // 行番号 2 の行とその行のすべての列を出力する
    Table1.PrintRow(2);
    Table1.PrintColumns();
}

```

```

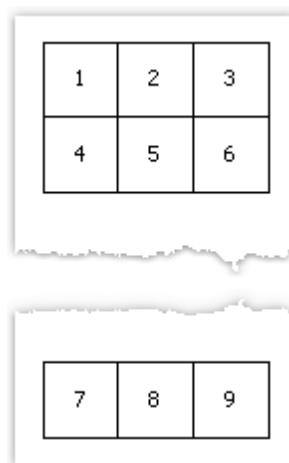
Table1.PrintRow(1);
Table1.PrintColumns();

// 行番号 2 の行の前に改ページを挿入する
Table1.PageBreak();

// 行番号 2 の行とその行のすべての列を出力する
Table1.PrintRow(2);
Table1.PrintColumns();
}

```

実行の結果は次のようにになります。



## 例 8: 集計の出力

例 6 の出力に集計関数を使用してみましょう。例 6 の表のフォームを次のように変更します。

Product Name	Unit Price	Units In Stock
[Products.ProductName]	[Products.UnitPrice]	[Products.UnitsInStock]
Total:		[Sum(Cell8)]

レポートを実行すると、次のように出力されます。

Outback Lager	1,500	15
Fløtemysost	2,150	26
Mozzarella di Giovanni	3,480	14
Röd Kaviar	1,500	101
Longlife Tofu	1,000	4
Rhönbräu Klosterbier	775	125
Lakkalikööri	1,800	57
Original Frankfurter grüne Soße	1,300	32
<b>Total:</b>	<b>3119</b>	

## 行列タイプのレポート

行列オブジェクトはさまざまな表や表オブジェクトのようご行、列およびセルで構成されています。ただし、行列内に入る行数と列数については事前に把握できません。これはこのオブジェクトが接続するデータによって決まります。

このオブジェクトは次のように表示されます。

Employee	[Year]	集計
[Name]	[Revenue]	
集計		

オブジェクトの出力時、行列は値で埋められ、それに応じてその行列は拡張したり縮小したりします。出力結果は次のようにになります。

Employee	1999	2000	2001	2002	集計
Andrew Fuller	¥390,000	¥210,000		¥180,000	¥780,000
Janet Leverling	¥610,000	¥320,000			¥930,000
Nancy Davolio	¥330,000	¥270,000	¥310,000	¥170,000	¥1,080,000
Steven Buchanan		¥399,900	¥810,000		¥1,209,900
集計	¥1,330,000	¥1,199,900	¥1,120,000	¥350,000	¥3,999,900

## 行列の概要

行列の要素について説明します。

	1	2	3	4
a	a1	a2	a3	a4
b	b1	b2	b3	b4

上の図は 2 行 4 列の行列です。"a"、"b" は行ヘッダー、"1"、"2"、"3"、"4" は列ヘッダー、a1 ~ a4 および b1 ~ b4 はセルを表します。このようなレポートを作成する場合、3 つの列に以下のよのうデータを含むデータソースが 1 つ必要です。

a	1	a1
a	2	a2
a	3	a3
a	4	a4
b	1	b1
b	2	b2
b	3	b3
b	4	b4

第 1 列は行列の行、第 2 列は行列の列、そして第 3 列は第 1 列と第 2 列で示される行と列が交差する位置にあるセルの内容を表します。レポートを作成する場合、FastReport はメモリに行列を作成し、それにデータを埋めています。指定された数の行または列にデータが入りきらなくなると、行列が動的に増加します。

ヘッダーは複数のレベルを持つことができます。次の例を見てみましょう。

10		20	
1	2	1	2
a	a10.1	a10.2	a20.1
b	b10.1	b10.2	b20.1

この例では、1 つの列に 2 つの値が含まれています。このレポートは次のようなデータが必要です。

a	10	1	a10.1
a	10	2	a10.2
a	20	1	a20.1
a	20	2	a20.2
b	10	1	b10.1
b	10	2	b10.2
b	20	1	b20.1
b	20	2	b20.2

第 1 列は行列の行を表し、第 2 列と第 3 列は行列の列を表します。最後のデータ列にはセルの値が含まれます。

次の図は、小計および総計を表す行列要素の例を示しています。

	10			20			集計
	1	2	集計	1	2	集計	
a	a10.1	a10.2	a10.1+a10.2	a20.1	a20.2	a20.1+a20.2	合計(a)
b	b10.1	b10.2	b10.1+b10.2	b20.1	b20.2	b20.1+b20.2	合計(b)
集計	a10.1+b10.1	a10.2+b10.2	a10.1+a10.2+b10.1+b10.2	a20.1+b20.1	a20.2+b20.2	a20.1+a20.2+b20.1+b20.2	合計(a)+合計(b)

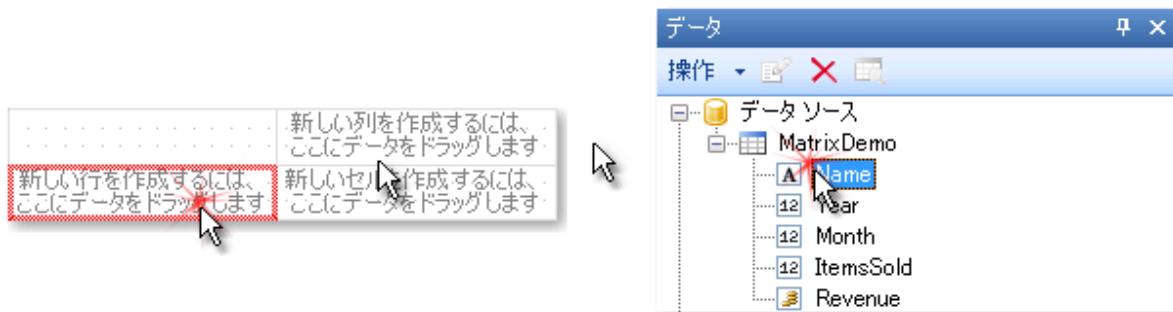
このレポートは前の例で使用した同じデータを基に作成されます。この図で、灰色で示される行は自動的に計算されています。

## 行列の設定

シート上に新しい行列オブジェクトを置くと、次のように表示されます。

新しい列を作成するには、 ここにデータをドラッグします	新しいセルを作成するには、 ここにデータをドラッグします
新しい行を作成するには、 ここにデータをドラッグします	新しいセルを作成するには、 ここにデータをドラッグします

行列はマウスを使用して設定することができます。これを行うには、[データ] ウィンドウにリストされるデータソースの列を、行列オブジェクト上へドラッグアンドドロップします。これで行、列およびセルが作成されます。行列では、新しいデータが置かれる場所が赤い枠で強調表示されます。



The screenshot shows the 'Data' window on the right, which lists a data source named 'MatrixDemo' containing columns for 'Name', 'Year', 'Month', 'ItemsSold', and 'Revenue'. On the left, a matrix is displayed with three columns labeled 'Year', 'Name', and '集計'. The second row of the matrix is highlighted with a red box, indicating where a new row is being created. The 'Name' column is currently selected in the data source tree.

行列に既になんらかの要素が含まれていた場合、新しい要素を置くときにはインジケーターが表示されます。次のようないくつかの状況では、新しいデータは "[Year]" と "Name" 要素の間に置かれます。

Year	Name	新しい列を作成するには、 ここにデータをドラッグします
[Year]	[Name]	新しいセルを作成するには、 ここにデータをドラッグします
集計	新しい行を作成するには、 ここにデータをドラッグします	

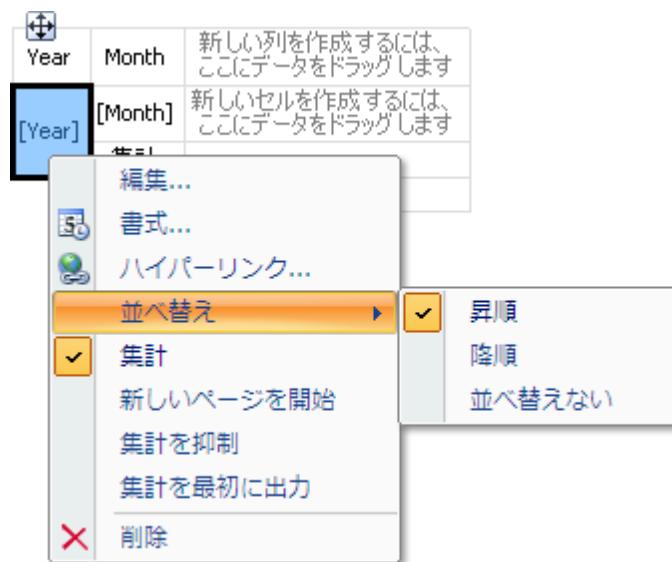
既存の要素の順序を変更することもできます。これを行うには、要素の境界線上（黒枠で示されている部分）でマウスの左ボタンを押し、そのまま移動先の場所へドラッグします。

Year	Month	Name	新しい列を作成するには、 ここにデータをドラッグします
[Year]	[Month]	[Name]	新しいセルを作成するには、 ここにデータをドラッグします
		集計	
		集計	

要素を削除するには、その要素を選択し、Delete キーを押します。

## ヘッダーの設定

ヘッダー要素を設定するには、その要素(セル)を選択して右クリックし、コンテキストメニューを表示します。



既定では、行列ヘッダーのデータは昇順で並べ替えられます。並べ替えの順序は、[並べ替え]サブメニューから選択することで変更することができます。

通常、行列のヘッダーの項目にはそれぞれ集計(「集計」というテキストのセル)があります。集計を削除するには、その集計(セル)を選択してDeleteキーを押します。集計を再び有効にするには、その集計の対象となる要素を選択して右クリックし、コンテキストメニューから[集計]を選択します。

[新しいページを開始]を有効にすると、行列に対し、各ヘッダー値を出力した後に改ページを挿入することを通知します。たとえば、上図の[Year]項目に対して改ページを有効にした場合、年の値が変わると改ページされるようになります。

[集計を抑制]を有効にすると、グループ(集計値の計算対象)に値が1つしかない場合は集計を行わないようにすることができます。

## セルの設定

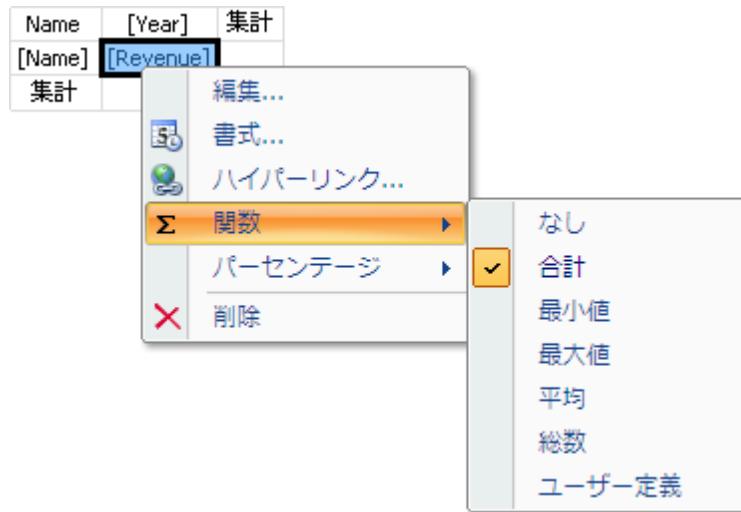
行列のセルでは、集計を計算するときに使用される関数を選択することができます。使用できる関数の一覧は次のとおりです。

関数	説明
なし	セルの値は処理されません。
Sum(合計)	行列のセルに含まれる値の合計を返します。
Min(最小値)	最小値を返します。
Max(最大値)	最大値を返します。

Average (平均) 平均値を返します。

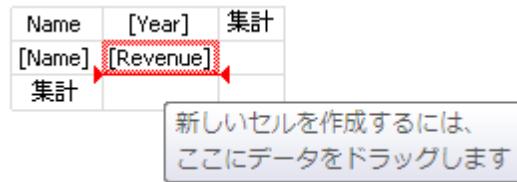
Count (総数) 空でない値の数を返します。

既定では Sum (合計) 関数が使用されます。この関数を変更するには、セルを右クリックしてコンテキストメニューの [関数] サブメニューから選択します。



指定のセルに集計を出力させない場合は、[なし] 関数を選択します。

行列には、1つまたは複数のデータセルを持つことができます。行列に複数のセルがある場合は、それらを横に並べる、または縦に交互に配置されるようにすることができます。行列の CellsSideBySide プロパティはセルの配置方法を制御します。この配置方法は、行列オブジェクトのコンテキストメニュー ([セルを横に並べる]) から変更することができます。また、行列に2つ目のセルを追加する場合の配置順序を選択することもできます。これを行う場合、2つ目の値が置かれる場所を示す赤いインジケーターを見てください。



2つ目の値が追加されると、その他の値は選択した配置順序に基づいて追加されます。

## 行列のスタイル

行列のセルの外観を変更するには、そのセルをクリックして選択し、ツールバーを使用してフォント、罫線および塗りつぶしを設定することができます。一度に複数のセルの外観を変更するには、それらのセルをまとめて選択します。これを行うには、対象とするセル範囲の左上のセルを選択し、左ボタンを押したままマウスを希望の選択範囲まで移動させます。

Name	[Year]		集計
	[Month]	集計	
[Name]	[Revenue]		
集計			

行列全体の外観を変更する場合はスタイルを使用できます。これを行うには、行列オブジェクトのコンテキストメニューを開き、[スタイル]サブメニューからスタイルを選択します。

## 行と列のサイズ管理

行列オブジェクトは一種の表オブジェクトであるため、表オブジェクトと同様に行と列のサイズを設定できます。

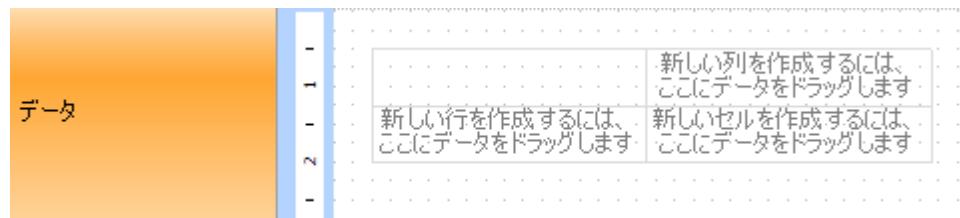
行列はあらかじめ"自動サイズ調整"モードが有効になっています。このモードでは、行列オブジェクトが行および列のサイズを自動的に計算します。オブジェクトのサイズは手動で管理することもできます。これを行うには、行列オブジェクトのAutoSize(自動サイズ調整)プロパティを無効(False)にします。行と列には同じプロパティがあり、それらは行列オブジェクトの自動サイズ調整が無効になっている場合に使用できます。

列の幅を最小値と最大値を指定して制限するには、その列のMinWidthおよびMaxWidthプロパティを設定します。

行の高さを最小値と最大値を指定して制限するには、その行のMinHeightおよびMaxHeightプロパティを設定します。

## 例

Matrix オブジェクトの使用例について説明します。まず、新しいレポートを作成し、レポートタイトルバンドに行列オブジェクトを置きます。データバンドに行列オブジェクトを置くこともできます。その場合は、バンドをデータソースに接続する必要はありません。行列を置くバンドは、2つのバンドのどちらでも構いません。これは、どちらのバンドもレポートを実行したときに1回出力されるものだからです。レポートデザインは次のようになります。



ページが変わったときに出力されるページヘッダー、ページフッターなどのバンド上には行列オブジェクトを置かないでください。そのようなバンドに行列が置かれると、バンドが変更されるたびに行列が作成され、スタックオーバーフローを引き起こします。

行列の使用例の多くは MatrixDemo テーブルを使用しています。これは FastReport パッケージに含まれています。このテーブルには以下のデータが含まれています。

名前	Year	Month	Items Sold	Revenue
Nancy Davolio	1999	2	1	100000
Nancy Davolio	1999	11	1	110000
Nancy Davolio	1999	12	1	120000
Nancy Davolio	2000	1	1	130000
Nancy Davolio	2000	2	2	140000
Nancy Davolio	2001	2	2	150000
Nancy Davolio	2001	3	2	160000
Nancy Davolio	2002	1	2	170000
Andrew Fuller	2002	1	2	180000
Andrew Fuller	1999	10	2	190000
Andrew Fuller	1999	11	2	200000
Andrew Fuller	2000	2	2	210000
Janet Leverling	1999	10	3	300000

Janet Leverling	1999	11	3	310000
Janet Leverling	2000	3	3	320000
Steven Buchanan	2001	1	3	400000
Steven Buchanan	2001	2	4	410000
Steven Buchanan	2000	1	4	399900

## 例 1 : シンプルな行列

行列には、1つのデータセルと同様に1つの値が行および列に含まれます。行列を作成するには、MatrixDemo テーブルのデータ列を次のように追加する必要があります。

- "Year" データ列を行ヘッダーに追加する
- "Name" データ列を列ヘッダーに追加する
- "Revenue" データ列を行列セルに追加する

この作業後、行列は次のように表示されます。

Year	[Name]	集計
[Year]	[Revenue]	
集計		

次に、行列の外観を改良します。

- 行列のスタイルに"オレンジ"を選択する
- 行列のすべてのセルに対し、8 ポイントサイズの "Tahoma" フォントを選択する
- "集計" に太字属性を選択する
- 最上行のセルに対する塗りつぶし設定で、"ガラス" タイプを選択する(色は"Gold" に設定)
- 行列の自動サイズを無効 (AutoSize = False) にし、行と列のサイズを広げます。

この作業後、行列は次のような外観になります。

Year	[Name]	集計
[Year]	[Revenue]	
集計		

レポートを実行すると、次のように出力されます。

Year	Andrew Fuller	Janet Leverling	Nancy Davolio	Steven Buchanan	集計
1999	390000	610000	330000		1330000
2000	210000	320000	270000	399900	1199900
2001			310000	810000	1120000
2002	180000		170000		350000
集計	780000	930000	1080000	1209900	3999900

この行列には以下のことが不足しています。

- "Name" データ列用のタイトルがありません。
- 合計が通貨書式で出力されません。

以下のいくつかの方法を用いて"Name" データ列に対するタイトルを追加することができます。

- "Year/Employee" テキストを行列の左上隅に入力することができます。
- 対角線および1つのテキストオブジェクトを追加して次のように配置することができます。

Year	Employee	[Name]	集計
[Year]		[Revenue]	
集計			

- 行列のタイトルを有効にします。これを行うには 行列オブジェクトのコンテキストメニューで [タイトルを表示] を選択します。タイトルにはテキストを含めることができます。

	Employee	
Year	[Name]	集計
[Year]	[Revenue]	
集計		

データの書式を設定するには 下図のようセル領域全体を選択し コンテキストメニューから [書式] を選択して書式を設定します。

	Employee	
Year	[Name]	集計
[Year]	[Revenue]	
集計		

この作業後に実行されたレポートは次のようになります。

		Employee			
Year	Andrew Fuller	Janet Leverling	Nancy Davolio	Steven Buchanan	集計
1999	¥390,000	¥610,000	¥330,000		¥1,330,000
2000	¥210,000	¥320,000	¥270,000	¥399,900	¥1,199,900
2001			¥310,000	¥810,000	¥1,120,000
2002	¥180,000		¥170,000		¥350,000
<b>集計</b>	<b>¥780,000</b>	<b>¥930,000</b>	<b>¥1,080,000</b>	<b>¥1,209,900</b>	<b>¥3,999,900</b>

## 例 2 :複数 レベルのヘッダー

行列には 1 行に 1 つの値、1 列および 1 つのデータセルに 2 つの値が含まれます。ここでは、前の例で作成した行列に新しい項目を追加します。

- "Year" 項目の右側の行ヘッダーに"Month" データ列を追加します。

新しい項目の追加後、行列の外観を変更します。セルの書式も設定する必要があります。この作業後、行列は次のように表示されます。

		Employee	
Year	Month	[Name]	集計
[Year]	[Month]	[Revenue]	
	集計		
<b>集計</b>			

レポートを実行すると、次のように出力されます。

		Employee				
Year	Month	Andrew Fuller	Janet Leverling	Nancy Davolio	Steven Buchanan	集計
1999	2			¥100,000		¥100,000
	10	¥190,000	¥300,000			¥490,000
	11	¥200,000	¥310,000	¥110,000		¥620,000
	12			¥120,000		¥120,000
	<b>集計</b>	<b>¥390,000</b>	<b>¥610,000</b>	<b>¥330,000</b>		<b>¥1,330,000</b>
2000	1			¥130,000	¥399,900	¥529,900
	2	¥210,000		¥140,000		¥350,000
	3		¥320,000			¥320,000
	<b>集計</b>	<b>¥210,000</b>	<b>¥320,000</b>	<b>¥270,000</b>	<b>¥399,900</b>	<b>¥1,199,900</b>
2001	1				¥400,000	¥400,000
	2			¥150,000	¥410,000	¥560,000
	3			¥160,000		¥160,000
	<b>集計</b>			<b>¥310,000</b>	<b>¥810,000</b>	<b>¥1,120,000</b>
2002	1	¥180,000		¥170,000		¥350,000
	<b>集計</b>	<b>¥180,000</b>		<b>¥170,000</b>		<b>¥350,000</b>
<b>集計</b>		<b>¥780,000</b>	<b>¥930,000</b>	<b>¥1,080,000</b>	<b>¥1,209,900</b>	<b>¥3,999,900</b>

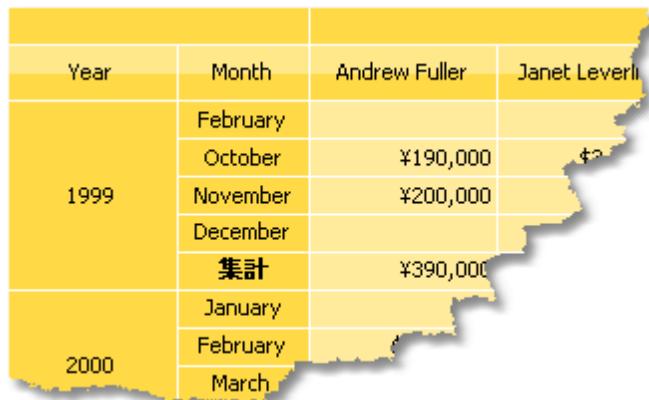
## 例 3 :月名の出力

前の例の行列では、Month 項目のセルに数字の月が出力されました。これは "Month" データ列に月の名前ではなく数字が含まれているからです。月の名前が出力されるようするには、以下のことを行います。

- 月の数字が出力されるセルを選択します。ここでは、そのセル名を "Cell8" と仮定します。
- [プロパティ] ウィンドウで、 ボタンをクリックし、BeforePrint イベントをダブルクリックします。
- FastReport はレポートスクリプトに空のイベントハンドラーを追加します。次のコードを記述します。

```
private void Cell8_BeforePrint(object sender, EventArgs e)
{
    string[] monthNames = new string[] {
        "January", "February", "March", "April",
        "May", "June", "July", "August",
        "September", "October", "November", "December" };
    // Cell8 は月の数字を出力するセル
    // Cell8.Value はセルに出力される値 (つまり月の数字)
    // この値は System.Object 型なので、これを int にキャストする必要がある
    Cell8.Text = monthNames[(int)Cell8.Value - 1];
}
```

レポートを実行すると、次のように出力されます。



Year	Month	Andrew Fuller	Janet Leverling
1999	February		
	October	¥190,000	¥200,000
	November	¥200,000	
	December		
	集計	¥390,000	
2000	January		
	February		
	March		

## 例 4 :条件付き強調表示

行列には、テキストオブジェクトのようく条件付き強調表示を設定することができます。この詳細については [条件付き強調表示](#) を参照してください。

ここでは、例 2 で作成した行列を使用し、金額が 300000 を超える場合に赤色で強調表示する方法を説明します。これを設定するに

は "Revenue" とテキストのセルを選択し、テキストツールバー上の  ボタンをクリックします。強調表示の条件エディターで次の条件を追加します。

Value > 300000

条件と一致するテキストの色に赤 ("Red") を設定します。実行されたレポートは次のようになります。

		Employee				
Year	Month	Andrew Fuller	Janet Leverling	Nancy Davolio	Steven Buchanan	集計
1999	2			¥100,000		¥100,000
	10	¥190,000	¥300,000			¥490,000
	11	¥200,000	¥310,000	¥110,000		¥620,000
	12			¥120,000		¥120,000
	集計	¥390,000	¥610,000	¥330,000		¥1,330,000
2000	1			¥130,000	¥399,900	¥529,900
	2	¥210,000		¥140,000		¥350,000
	3		¥320,000			¥320,000
	集計	¥210,000	¥320,000	¥270,000	¥399,900	¥1,199,900
2001	1				¥400,000	¥400,000
	2			¥150,000	¥410,000	¥560,000
	3			¥160,000		¥160,000
	集計			¥310,000	¥810,000	¥1,120,000
2002	1	¥180,000		¥170,000		¥350,000
	集計	¥180,000		¥170,000		¥350,000
集計		¥780,000	¥930,000	¥1,080,000	¥1,209,900	¥3,999,900

ここでは、集計値は強調表示されていません。強調表示の条件は1つのセルにのみ設定しているため、集計値は強調表示されません。他の値も強調表示させるには、行列のすべてのセルに対して強調表示を設定する必要があります。

この例では、セル自身の値に対する条件付き強調表示を使用しています。このほか、行列のヘッダーの値に応じてセルを強調表示することができます。次の例では、2000年に属するセルを強調表示する方法を示します。このため、次の図で示すように行列のセルを選択します。

		Employee	
Year	Month	[Name]	集計
[Year]	[Month]	[Revenue]	
	集計		
集計			

選択したセルに対し、次の強調表示の条件を設定します（塗りつぶしの色も適宜設定します）。

(int)Matrix1.RowValues[0] == 2000

この条件の "Matrix1" は行列の名前です。行列の RowValues プロパティは "Object []" 型を持ち 現在出力されている行に対応する行ヘッダーの値の配列が含まれます。配列の値の数は ヘッダーのレベル数と同等です。この例では2 つの値があります。最初の値は "Year"、もう1 つの値は "Month" です。

2 つ目の行は強調表示しないでください。これは RowValues プロパティの値が確定しないため、レポート作成時にエラーが発生します。

レポートを実行すると 次のよう出力されます。

		Employee				
Year	Month	Andrew Fuller	Janet Leverling	Nancy Davolio	Steven Buchanan	集計
1999	2			¥100,000		¥100,000
	10	¥190,000	¥300,000			¥490,000
	11	¥200,000	¥310,000	¥110,000		¥620,000
	12			¥120,000		¥120,000
	集計	¥390,000	¥610,000	¥330,000		¥1,330,000
2000	1			¥130,000	¥399,900	¥529,900
	2	¥210,000		¥140,000		¥350,000
	3		¥320,000			¥320,000
	集計	¥210,000	¥320,000	¥270,000	¥399,900	¥1,199,900
2001	1				¥400,000	¥400,000
	2			¥150,000	¥410,000	¥560,000
	3			¥160,000		¥160,000
	集計			¥310,000	¥810,000	¥1,120,000
2002	1	¥180,000		¥170,000		¥350,000
	集計	¥180,000		¥170,000		¥350,000
集計		¥780,000	¥930,000	¥1,080,000	¥1,209,900	¥3,999,900

列の値を参照する場合は 行列の ColumnValues プロパティを使用することもできます。

## 例 5 :偶数行の強調表示

行列の外観を改良するため、偶数行または偶数列を別の色で強調表示することができます。ここでは例 2 で作成した行列を使用して、この設定方法を説明します。

下図で示すように 行列のデータ領域全体を選択します。

		Employee	
Year	Month	[Name]	集計
[Year]	[Month]	[Revenue]	
	集計		
集計			

強調表示の条件エディターを呼び出します。次の条件を追加します。

```
Matrix1.RowIndex % 2 != 0
```

また、背景色(塗りつぶし)には、元の色より少し暗めの色を選択します。この条件例の"Matrix1"とは行列の名前です。この行列のRowIndex プロパティは現在出力されている行の番号を返します。

列を強調表示する場合は、行列のColumnIndex プロパティを同様に使用します。

レポートを実行すると、次のように出力されます。

		Employee				
Year	Month	Andrew Fuller	Janet Leverling	Nancy Davolio	Steven Buchanan	集計
1999	2			¥100,000		¥100,000
	10	¥190,000	¥300,000			¥490,000
	11	¥200,000	¥310,000	¥110,000		¥620,000
	12			¥120,000		¥120,000
	集計	¥390,000	¥610,000	¥330,000		¥1,330,000
2000	1			¥130,000	¥399,900	¥529,900
	2	¥210,000		¥140,000		¥350,000
	3		¥320,000			¥320,000
	集計	¥210,000	¥320,000	¥270,000	¥399,900	¥1,199,900
2001	1				¥400,000	¥400,000
	2			¥150,000	¥410,000	¥560,000
	3			¥160,000		¥160,000
	集計			¥310,000	¥810,000	¥1,120,000
2002	1	¥180,000		¥170,000		¥350,000
	集計	¥180,000		¥170,000		¥350,000
集計		¥780,000	¥930,000	¥1,080,000	¥1,209,900	¥3,999,900

## 例 6 :式の使用

前の例では、[データ]ウインドウからデータ列をドラッグすることで行列を作成しました。行列の作成には式を使用することもできます。行列に式を挿入するには、以下のことを行います。

- [データ]ウインドウの要素を行列に追加します。どのような要素でも可能です(システム変数 "Date" など)。その要素を使用して行列要

素を作成します。

- その要素をダブルクリックし、式エディター ウィンドウで必要な式を選択します。

行列にデータフィールドではなく式を指定する場合は、行列のDataSource プロパティが正しく設定されていることを確認してください。データ列を使って行列を作成する場合は、データ列を行列にドラッグしたときにこのプロパティの値は自動的に設定されます。

式の使用方法の例を見てみましょう。このため、データソースとして Order Details テーブルを使用します。このテーブルには販売された製品の一覧が含まれています。また、このテーブルには複数の関係が設定されており、従業員の名前、製品名およびそのカテゴリにアクセスできます。

この行列では、各従業員の売上高を製品別に示します。行列を作成するには、以下の手順に従います。

- "Order Details.Products.Categories.CategoryName" データ列を列ヘッダーに追加します。
- 行ヘッダーに項目を追加して行列の要素を作成します。ここではヘッダー要素として次のような式を設定します。

[Order Details.Orders.Employees.FirstName] + " " + [Order Details.Orders.Employees.LastName]

- データセルに項目を追加して行列の要素を作成します。ここではデータセル要素として次のような式を設定します。

[Order Details.UnitPrice] \* [Order Details.Quantity] \* (decimal)(1 - [Order Details.Discount])

従業員の名前は "Employees.FirstName" データ列を使用すれば取得できますが、ここでは長いデータ列を指定しています。これは、行列が Order Details データソースに接続されているからです。このデータソースとほかのテーブル間の関係を使用すると、それぞれの列への参照が簡単になります（関係の詳細については、データの章を参照してください）。

"Employees.FirstName" データ列へ直接参照する場合は、テーブルで先頭に位置する従業員の名前が取得されます。

行列の外観を設定します。この作業後、行列は次のように表示されます。

Employee	[CategoryName]	集計
[Order Details.Orders.Employees.FirstName] + " " + [Order Details.Orders.Employees.LastName]	[[Order Details.UnitPrice] * [Order Details.Quantity] * (decimal)(1 - [Order Details.Discount])]	
集計		

レポートを実行すると、複数シートにわたる大きな行列が出力されます。

Employee	Beverages	Condiments	Confections	Dairy Products	Grains/Cereals
Andrew Fuller	4,024,825	1,485,067	2,145,569	2,381,255	1,234,500
Anne Dodsworth	1,964,256	1,012,555	805,316	2,110,113	1,024,689
Janet Leverling	4,475,741	1,338,164	3,362,240	3,232,084	1,124,500
Laura Callahan	1,789,785	1,463,766	2,169,991	2,126,947	1,024,689
Margaret Peacock	5,030,821	2,331,487	2,776,873	3,351,235	1,124,500
Michael Suyama	945,020	464,847	685,963	724,890	462,500
Nancy Davolio	4,659,936	1,356,156	2,856,843	2,937,500	1,124,500
Robert King	2,796,383	885,138	1,024,500	1,024,500	1,024,500
Steven Buchanan	1,100,053	1,024,500	1,024,500	1,024,500	1,024,500

## 例 7 :セル内の図

行列のセルはテキストオブジェクトから継承されテキストデータを表示することができます。テキストだけでは不十分な場合、ほかのオブジェクトをセル内に置くことができます。ここでは、行列に図を表示させる方法について説明します。

そのため、例 6 で作成した行列を使用します。従業員の写真 ("Employees.Photo" データ列) とカテゴリのイメージ写真 ("Categories.Picture" データ列) を追加します。以下のことを行います。

- 従業員の名前が含まれるセルを選択し、そのサイズを大きくします。
- そのセルに図オブジェクトを追加します。
- 従業員の写真を表示するには、図オブジェクトを次に示すデータ列にバインドします（これはオブジェクトのエディターで設定できます）。

Order Details.Orders.Employees.Photo

- カテゴリ名を含むセルを選択し、そのサイズを大きくします。
- そのセルに図オブジェクトを追加します。
- カテゴリのイメージ写真を表示するには、図オブジェクトを次に示すデータ列にバインドします（これはオブジェクトのエディターで設定できます）。

Order Details.Products.Categories.Picture

この作業後、行列は次のように表示されます。

Employee	[CategoryName] 	集計
[Order] 	$[[Order Details.UnitPrice] * [Order Details.Quantity] * (decimal)(1 - [Order Details.Discount])]$	
集計		

レポートを実行すると、次のように出力されます。

Employee	Beverages 	Condiments 	Confections 
Andrew Fuller 	4,024,825		1,485,067
Anne Dodsworth 	1,964,256		1,012,5
Janet Leverling 	4,4		

### 例 8 :セル内のオブジェクト

行列のセルに挿入されたオブジェクトを使用すると、さまざまな視覚効果が得られます。次の例では、従業員の売上高のレベルを表すシンプルな目盛りの描画方法を示します。

この行列にはMatrixDemo データソースを使用します。行列を作成するには 次のようデータ列を追加します。

- "Year" データ列を行ヘッダーに追加する
- "Name" データ列を列ヘッダーに追加する
- "Revenue" データ列を行列セルに追加する

行列を外観を次のように設定します。

Employee	[Year]	集計
[Name]	[Revenue]	
集計		

次に 3 つの図形オブジェクトを"Revenue" 値のセルに追加します。これらのオブジェクトは以下のよなインジケーターの役割を果たします。

- セルの値が10000 未満の場合は、赤色の図形オブジェクトを1 つ表示する
- セルの値が300000 未満の場合は、黄色の図形オブジェクトを2 つ表示する
- セルの値が300000 以上の場合は、緑色の図形オブジェクトを3 つ表示する

行列は次のように表示されます。

Employee	[Year]	集計
[Name]	■ ■ ■ [Revenue]	
集計		

これらのオブジェクトを制御するには、行列セルのイベントハンドラーを使用します。"Revenue" データのセルを選択し、[プロパティ] ウィンドウから BeforePrint イベントハンドラーを作成します。このハンドラーで以下のコードを記述します。

```
private void Cell4_BeforePrint(object sender, EventArgs e)
{
    // この例では Cell4 という名前のセルを使用

    // Cell4.Value プロパティからセルの値を取得する
    // 行列内のいくつかのセルが空であることを考慮する(ヌルチェック)
    // [MatrixDemo.Revenue] が System.Decimal 型であるため
    // 値を Decimal 型にキャストする必要がある
    decimal value = Cell4.Value == null ? 0 : (decimal)Cell4.Value;

    // 図形オブジェクトの表示を、次のように値に応じて切り替える
    // 値 < 10000 - 1 つのオブジェクトを表示
    // 値 < 300000 - 2 つのオブジェクトを表示
    // 値 >= 300000 - すべてのオブジェクトを表示
    Shape1.Visible = true;
    Shape2.Visible = value >= 10000;
```

```
Shape3.Visible = value >= 300000;
```

```
// オブジェクトの色を選択する
```

```
// 値 < 10000 - 赤色
```

```
// 値 < 300000 - 黄色
```

```
// 値 >= 300000 - 緑色
```

```
Color color = Color.Red;
```

```
if (value >= 10000)
```

```
color = Color.Yellow;
```

```
if (value >= 300000)
```

```
color = Color.GreenYellow;
```

```
// オブジェクトの色を設定する
```

```
Shape1.Fill = new SolidFill(color);
```

```
Shape2.Fill = new SolidFill(color);
```

```
Shape3.Fill = new SolidFill(color);
```

```
}
```

レポートを実行すると、次のように出力されます。

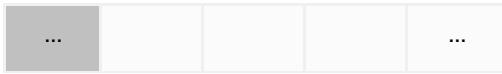
Employee	1999	2000	2001	2002	集計
Andrew Fuller	■■■ 390,000	■■ 210,000	■■ 180,000		780,000
Janet Leverling	■■■ 610,000	■■■ 320,000	■■ 170,000		930,000
Nancy Davolio	■■■ 330,000	■■ 270,000	■■■ 310,000	■■ 170,000	1,080,000
Steven Buchanan	■■ 399,900	■■■ 810,000	■■ 350,000		1,209,900
集計	1,330,000	1,199,900	1,120,000		3,999,900

## 例 9 :行列データの手動埋め込み

今までの例では、行列にはデータが自動的に入っていました。これは、行列がデータソースに接続されていました。行列用のデータソースはDataSource プロパティで指定されます。このプロパティの値は手動で設定しても、データ列を行列に追加したときに自動的に設定されます。

スクリプトを使用すると、行列にデータを手動で入力することができます。これには、行列のManualBuild イベントハンドラーを作成する必要があります。このイベントハンドラーのコードで AddValue メソッドを呼び出し、値を追加します。ここでは、次のような 10 行 10 列のテーブルを出力する行列を作成する方法を説明します。

	1	2	3	...
1	1			
2		2		
3			3	



以下のことを行います。

- 空の行列をレポートに追加します。
- 【データ】ウインドウから行列の行、列およびセルへ要素をドラッグして配置します。その行列要素をダブルクリックして式エディターを呼び出し、その式をクリアします。
- 行列のDataSource プロパティの値をクリアします。

上記の手順は1つの行、列およびセルを持つダミーの行列を作成するため必要です。この結果、行列は次のように表示されます。

	[0]	集計
[0]	[0]	
集計		

次にManualBuild イベントハンドラーを作成します。このため、行列を選択し、【プロパティ】ウインドウで ボタンをクリックします。

ManualBuild イベントをダブルクリックすると FastReport は空のイベントハンドラーを作成します。このハンドラーで以下のコードを記述します。

```
private void Matrix1_ManualBuild(object sender, EventArgs e)
{
    // 行列には、行、列およびセルに対し1つのレベルがある
    // 各要素にオブジェクト型 (object[]) の3つの配列を作成する
    // (レベル数に従う)
    object[] columnValues = new object[1];
    object[] rowValues = new object[1];
    object[] cellValues = new object[1];

    for (int i = 1; i <= 10; i++)
    {
        // 配列を埋める
        columnValues[0] = i;
        rowValues[0] = i;
        cellValues[0] = i;

        // データを行列へ追加する
        Matrix1.AddValue(columnValues, rowValues, cellValues);
    }
}
```

ハンドラーで、Matrix オブジェクトの AddValue メソッドを使用してデータを行列に入力します。このメソッドには3つのパラメーターがあり、各

パラメーターはSystem.Object型の配列となっています。最初のパラメーターは列の値、2番目のパラメーターは行の値、3番目のパラメーターがセルの値です。各配列の値の数は、オブジェクトの設定に従う必要があるので注意してください。この場合、オブジェクトは行、列およびセルにそれぞれ1つのレベルがあります。それに応じて各行、列およびセルに1つの値を提供します。

レポートを実行すると、次のように出力されます。

	1	2	3	4	5	6	7	8	9	10	集計
1	1										1
2		2									2
3			3								3
4				4							4
5					5						5
6						6					6
7							7				7
8								8			8
9									9		9
10										10	10
集計	1	2	3	4	5	6	7	8	9	10	55

行列で、行3および列7が交差する場所に"21"という値を追加する方法を示します。このため、コードを次のように変更します。

```
private void Matrix1_ManualBuild(object sender, EventArgs e)
{
    object[] columnValues = new object[1];
    object[] rowValues = new object[1];
    object[] cellValues = new object[1];

    for (int i = 1; i <= 10; i++)
    {
        columnValues[0] = i;
        rowValues[0] = i;
        cellValues[0] = i;

        Matrix1.AddValue(columnValues, rowValues, cellValues);
    }

    columnValues[0] = 7;
}
```

```

rowValues[0] = 3;
cellValues[0] = 21;
Matrix1.addValue(columnValues, rowValues, cellValues);
}

```

実行の結果は次のようになります。

	1	2	3	4	5	6	7	8	9	10	集計
1	1										1
2		2									2
3			3				21				24
4				4							4
5					5						5
6						6					6
7							7				7
8								8			8
9									9		9
10										10	10
集計	1	2	3	4	5	6	28	8	9	10	76

ご覧のとおり 行列は自動的に集計を計算します。

データに接続される行列用にManualBuild イベントハンドラーを使用することもできます。この場合、最初にイベントハンドラーが呼び出され、その後、データソースのデータが行列に格納されます。

## 対話型レポート

FastReport の実行レポートは対話形式で動作させることができます。これは、プレビュー ウィンドウでユーザーの操作に反応することを意味します。以下の相互作用を使用できます。

- レポートオブジェクト上をクリックしたときに、なんらかの操作を実行します。たとえば 詳細レポートを実行し、それを別のウィンドウに表示できます。
- プレビュー ウィンドウにレポートのアウトラインを表示できます。これはレポート上の検索に使用できます。

## ハイパー リンク

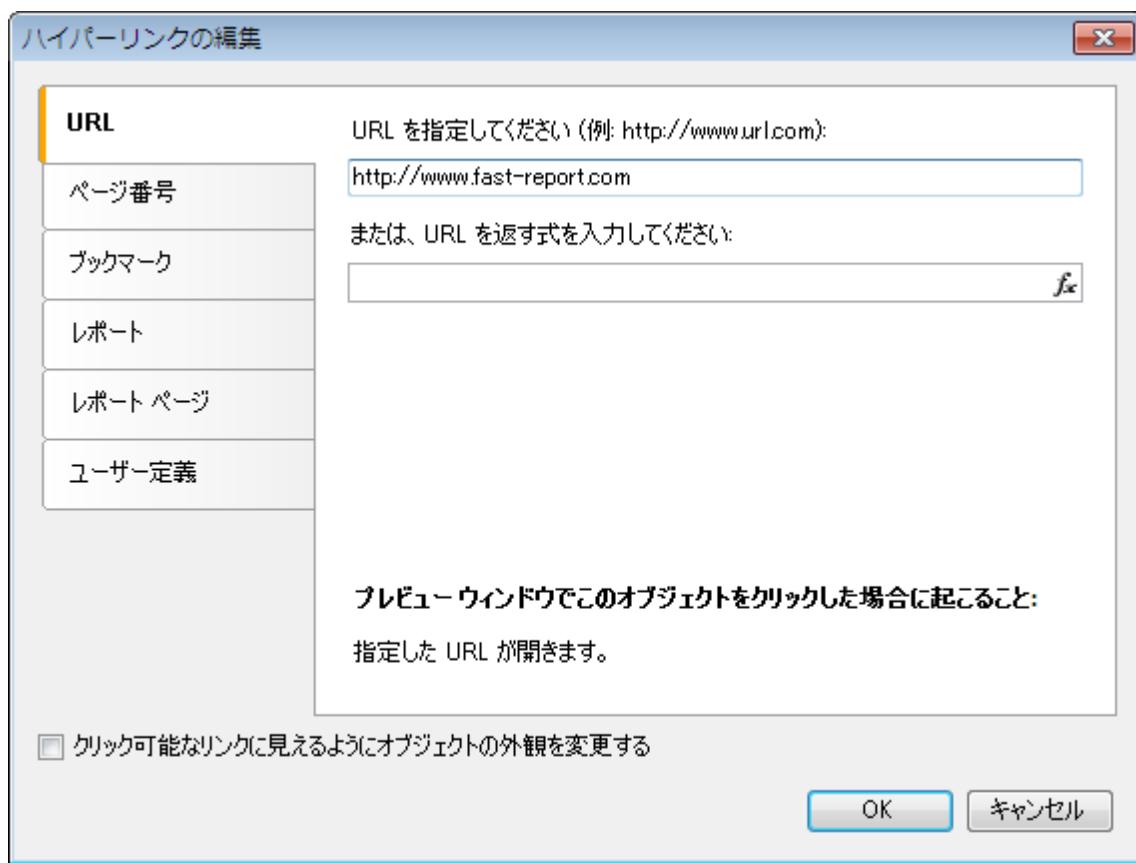
ほぼすべてのレポートオブジェクトにHyperlink プロパティがあります。このプロパティを使用すると、プレビュー ウィンドウ内でのマウスクリックに対するオブジェクトの反応を定義できます。

このようなオブジェクトをクリックしたときには、以下のいずれかの動作を発生させることができます。

- URL アドレスへ移動する
- 電子メールを送信する
- あらゆるシステム コマンドを実行する
- 指定した番号のレポートページへ移動する
- 別のレポートオブジェクトで定義されたブックマークへ移動する
- 別のプレビュー ウィンドウで詳細レポートを実行する
- スクリプトで定義したカスタム アクション

## ハイパー リンクの設定

ハイパー リンクを設定するには、対話型にするオブジェクトを選択して右クリックします。このコンテキストメニューから [ハイパー リンク] を選択します。ハイパー リンクエディター ウィンドウが開きます。



このウインドウの左側にあるタブからハイパーリンクの種類を選択します。タブを選択したら、ウインドウ下部にある [クリック可能なリンクに見えるようにオブジェクトの外観を変更する] チェックボックスをクリックしてオンにします。オブジェクトの外観を次のように変更します。

- テキストの色を青色にし、下線を設定します。
- カーソルのフォームを手の形 (Hand) に設定します。

ハイパーリンクはプレビュー ウィンドウでのみ表示され、印刷する必要がない場合もあります。その場合はオブジェクトの **Printable** プロパティを無効 (False) にするだけです。これは [プロパティ] ウィンドウで設定できます。

## URL へのリンク

このタイプのハイパーリンクを使用すると、以下のことが行えます。

- 指定したインターネットアドレスへ移動する
- ある種のシステムコマンドを実行する (例: 電子メール送信用の "mailto:" など)

このタイプのハイパーリンクをクリックすると、その値をパラメーターとする `System.Diagnostics.Process.Start` メソッドが実行されます。

ハイパーリンクの値は、以下の 2 つの方法を使用して指定できます。

- 値を直接指定します (例: `http://www.fast-report.com`)。

- ハイパーアリンクの値を返す式を指定します。この式はレポートの実行時に計算されます。

## ページ番号へのリンク

このタイプのリンクを使用すると、実行されたレポート内の特定のページへ移動させることができます。これは、先頭ページへの移動に用いることが多いでしょう。このためには、リンクの値としてページ番号（この場合であれば1）を指定します。

ページ番号は、以下の2つの方法を使用して指定できます。

- 番号を直接指定します（例：1）。
- ページ番号を返す式を指定します。この式はレポートの実行時に計算されます。

## ブックマークへのリンク

このタイプのリンクを使用すると、別のレポートオブジェクトで定義されたブックマークへ移動することができます。

HTML言語をご存じであれば、ブックマークはアンカーのように動作すると言えばおわかりでしょう。ブックマークには、名前および実行されたレポート内の明確な位置（ページ番号およびそのページ上の位置）が含まれます。名前によってそのブックマークへ移動すると、それは、指定した位置に移動します。

このリンクタイプを使用するには、まずブックマークを定義する必要があります。これを行うには、リンクをクリックしたときに移動させるオブジェクトを選択します。ほとんどのレポートオブジェクトにBookmarkプロパティがあります。このプロパティの変更は、[プロパティ]ウインドウで行うことができます。

Bookmarkプロパティには式が含まれます。この式は次のように使用できます。

- ブックマーク名を次のように文字列として指定する

"MyBookmark"

- ブックマークの名前を返す式を指定します。たとえば、データ列を式として使用できます。式の値は、レポートを実行したときに計算されます。

ブックマークを定義した後、ハイパーアリンク設定ウインドウでその名前を指定することができます。これには以下の2つの方法を使用します。

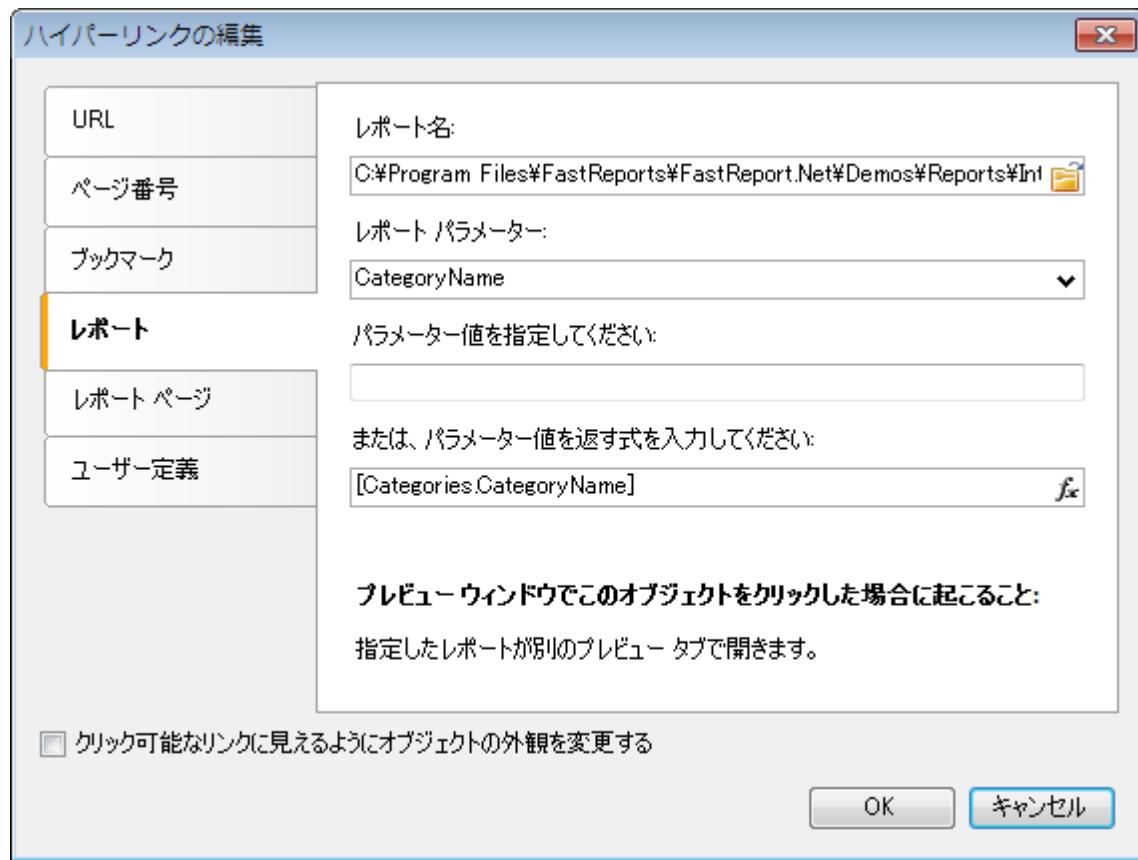
- ブックマークの名前を直接指定します。
- ブックマークの名前を返す式を指定します。たとえば、データ列がこれに該当します。この式はレポートの実行時に計算されます。

## 詳細レポートへのリンク

このリンクタイプを使用すると、別のレポートを実行し、そのレポートを別のプレビュー ウィンドウに表示することができます。

このタイプのハイパーアリンクには、以下のパラメーターを設定する必要があります。

- 詳細レポートの名前
- ハイパーアクションの値に使用されるレポートのパラメーターの名前
- ハイパーアクションの値



リンクをクリックすると、以下のことが実行されます。

- 指定されたレポートが読み込まれます。
- レポートのパラメーターがハイパーアクションの値に設定されます。
- レポートが作成され、別のプレビュー ウィンドウに表示されます。

レポートパラメーターの値は、以下の2つの方法を使用して指定できます。

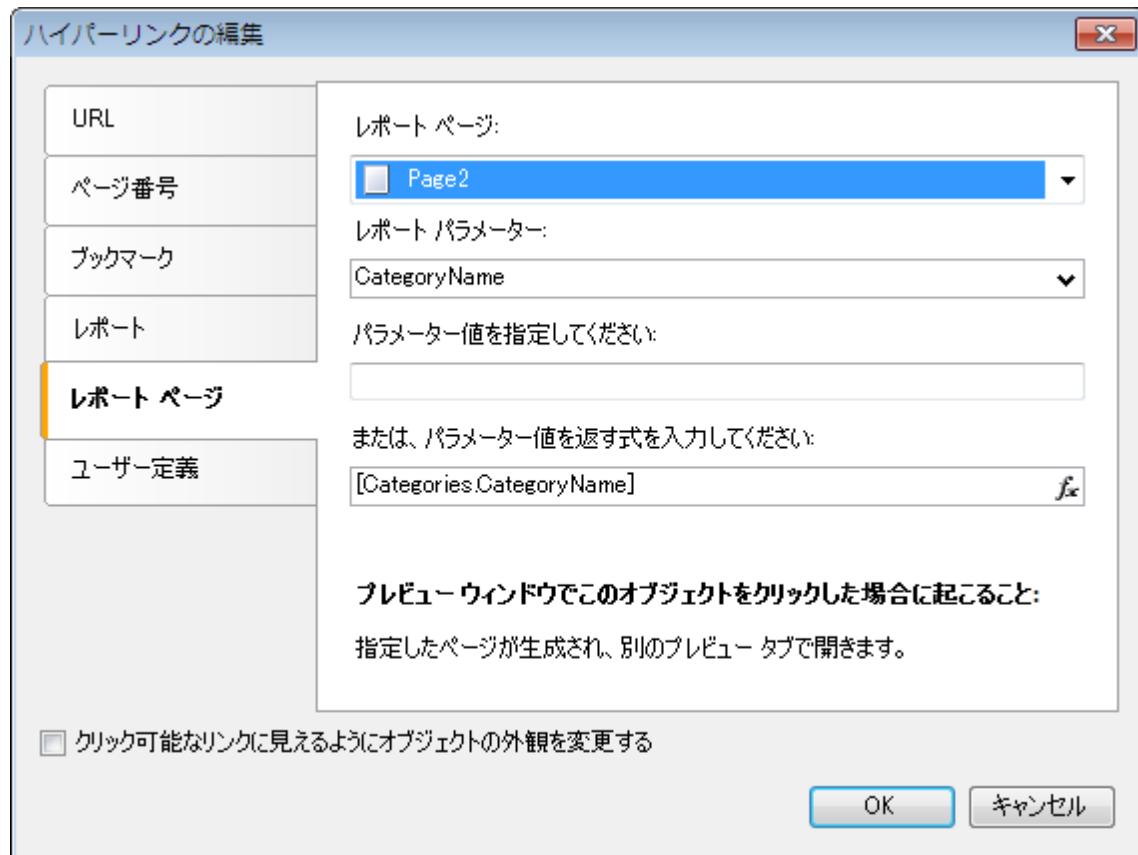
- 値を直接指定します。
- 値を返す式を指定します。この式はレポートの実行時に計算されます。

### 詳細ページへのリンク

このリンクタイプは、現在のレポートの別のページを詳細レポートとして使用することを除けば、「詳細レポートへのリンク」と同様に動作します。このため、レポートは少なくとも2つのページが含まれている必要があります（1つはメインレポート、もう1つは詳細レポート）。

このタイプのハイパーアリンクには、以下のパラメーターを設定する必要があります。

- そのレポートのページ名
- ハイパーアリンクの値に使用されるレポートのパラメーターの名前
- ハイパーアリンクの値



リンクをクリックすると、以下のことが実行されます。

- レポートのパラメーターがハイパーアリンクの値に設定されます。
- 指定されたレポートページが作成され、別のプレビュー ウィンドウに表示されます。

レポートパラメーターの値は、以下の2つの方法を使用して指定できます。

- 値を直接指定します。
- 値を返す式を指定します。この式はレポートの実行時に計算されます。

レポートページを選択する場合、そのVisible プロパティは **False** にリセットします。これは、メインレポートが作成される場合、このページが省かれることを意味します。

## カスタム リンク

このリンクタイプを使用すると、マウスクリックに対する独自の反応を定義できます。このため、オブジェクトのClick イベントハンドラーを使用します。これは次の手順で行います。

- オブジェクトを選択し、(表示されてない場合は) [プロパティ] ウィンドウを開きます。
- ⚡ ボタンをクリックしてオブジェクトのイベントを表示します。
- Click イベント項目をダブルクリックします。[コード] ウィンドウが開き、空のイベントハンドラーが作成されます。

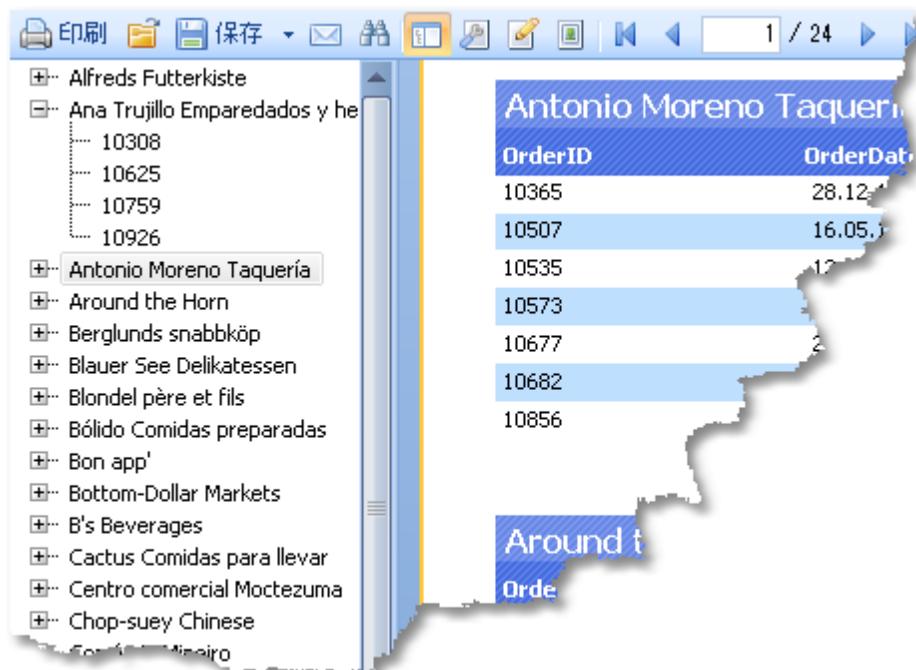
このイベントハンドラーのコード内で、必要なアクションを行います。最も多いと思われるアクションは、ハンドラーの呼び出しやハイパーインクの値のためのオブジェクトへのリンクです。ハンドラーのパラメーターである sender を使用します。

```
private void Text2_Click(object sender, EventArgs e)
{
    // sender - これはクリックされたオブジェクトです。
    // ハイパーインクの値を受け取るために
    // この sender を ReportComponentBase 型にキャストする必要があります。
    object hyperlinkValue = (sender as ReportComponentBase).Hyperlink.Value;

    MessageBox.Show("Hyperlink value = " + hyperlinkValue.ToString());
}
```

## レポートのアウトライン

レポートのアウトライン(別名 "ドキュメントマップ")はプレビュー ウィンドウに表示されるTreeView コントロールです。



このコントロールはツリー構造を示し、レポート作成時に形成されます。ツリー要素をクリックすると、対応するレポート要素へ移動します。

レポートにアウトラインが設定されると、自動的にそれが表示されるようになります。ツールバーの  ボタンをクリックすることでアウトラインの表示/非表示を切り替えることができます。レポートは自動的にアウトラインを作成しません。これについては手動で行う必要があります。

レポートページおよびそのすべてのバンドには OutlineExpression プロパティがあります。アウトラインを形成するには、このプロパティに要素のテキストを返す式を指定します。この式はバンドの出力時に計算され、その値がアウトラインに追加されます。マスター/詳細またはグレープタイプのレポートの場合、アウトラインの構造はレポートと同じような構造になります。

OutlineExpression プロパティは [プロパティ] ウィンドウで設定できます。

レポートタイプ別のアウトラインについては、以下の設定方法を推奨します。

- アウトラインに実行レポートのシートを表示させる場合は、レポートページの OutlineExpression プロパティを設定します。この式はページ番号を返します。

[PageN]

- 1 つのデータバンドを使用するシンプルリストレポートタイプの場合、そのバンドの OutlineExpression プロパティを設定します。式として、バンドに出力されるデータ列を使用します。
- 2 つのデータバンドを使用するマスター/詳細レポートタイプの場合、対応するバンドの OutlineExpression プロパティを設定します。たとえば "カテゴリ/製品" レポートタイプでは、最初のバンドの OutlineExpression プロパティにはカテゴリの名前、2 番目の OutlineExpression プロパティには製品の名前を含めます。
- グレープレポートタイプの場合、グレープヘッダーおよびデータバンドの OutlineExpression プロパティを設定します。グレープヘッダー用の式には、グレープ化の条件を使用します。データバンド用の式には、バンドに出力されるデータ列を使用します。

## 例

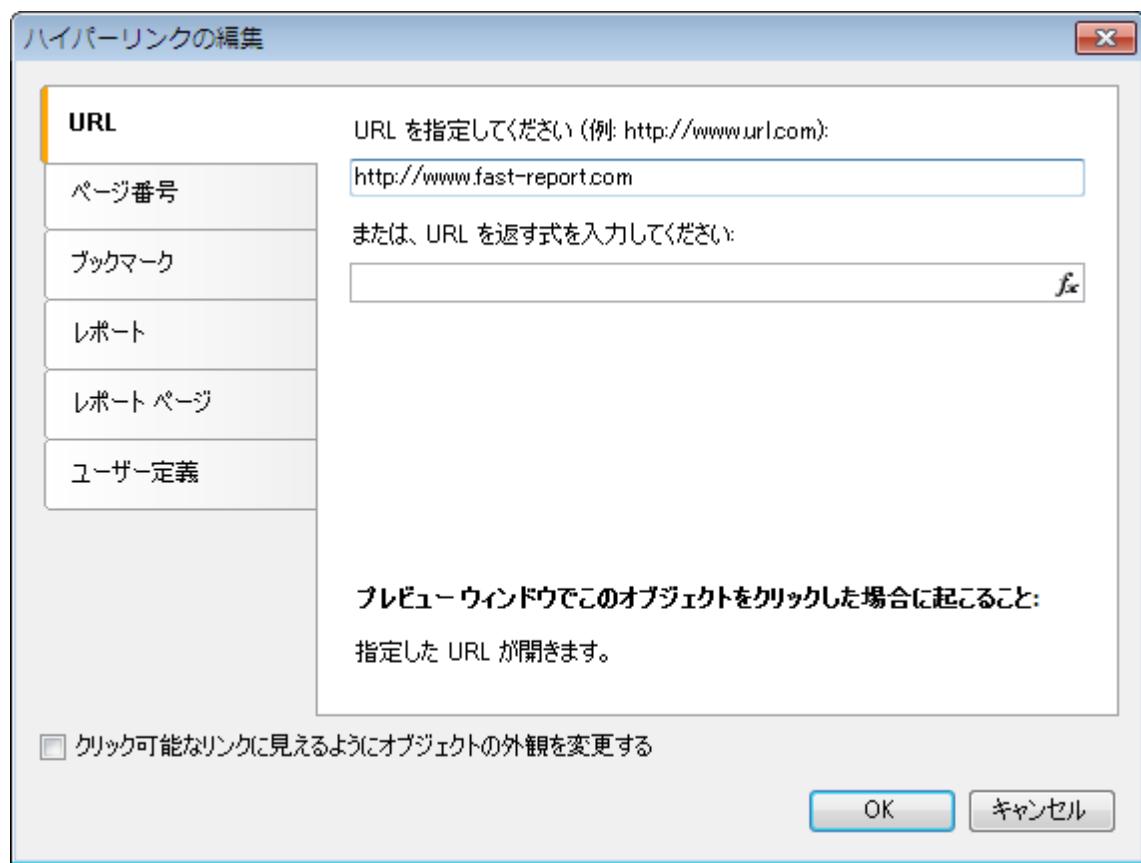
### 例 1 :Web ページへのリンク

この例では、1 つのテキストオブジェクトを持つシンプルなレポートを作成します。プレビュー ウィンドウでそのテキストオブジェクトをクリックすると FastReport の Web ページへ移動します。

新しいレポートを作成し、テキストオブジェクトを追加します。このオブジェクトに次のようなテキストを追加します。

FastReport ホーム ページへ

次に、オブジェクトを右クリックして、コンテキストメニューから [ハイパーリンク] を選択します。次のようにリンクを設定します。



リンクの設定後、テキストオブジェクトに対してリンク属性 (青色のテキスト、下線付き、手形のカーソル) を適用するため、[クリック可能なリンクに見えるようにオブジェクトの外観を変更する] オプションを有効 (オン) にします。

レポートを実行し、オブジェクトをクリックします。Web ブラウザ ウィンドウが開き、FastReport のホーム ページへ移動します。

## 例 2 詳細レポートの作成

この例では、カテゴリの一覧を表示するレポートを作成します。カテゴリ名をクリックすると、そのカテゴリの製品の一覧を含む詳細レポートが表示されます。

以下のことを行う必要があります。

- 最初に、詳細レポートを作成します。
- カテゴリを識別するレポートパラメーターを定義します。
- このパラメーターにデータフィルターを設定します。
- メインレポートを作成します。
- 選択したカテゴリに設定されたパラメーターに基づいて詳細レポートが実行されるようするため、メインレポートでハイパーリンクを設定します。

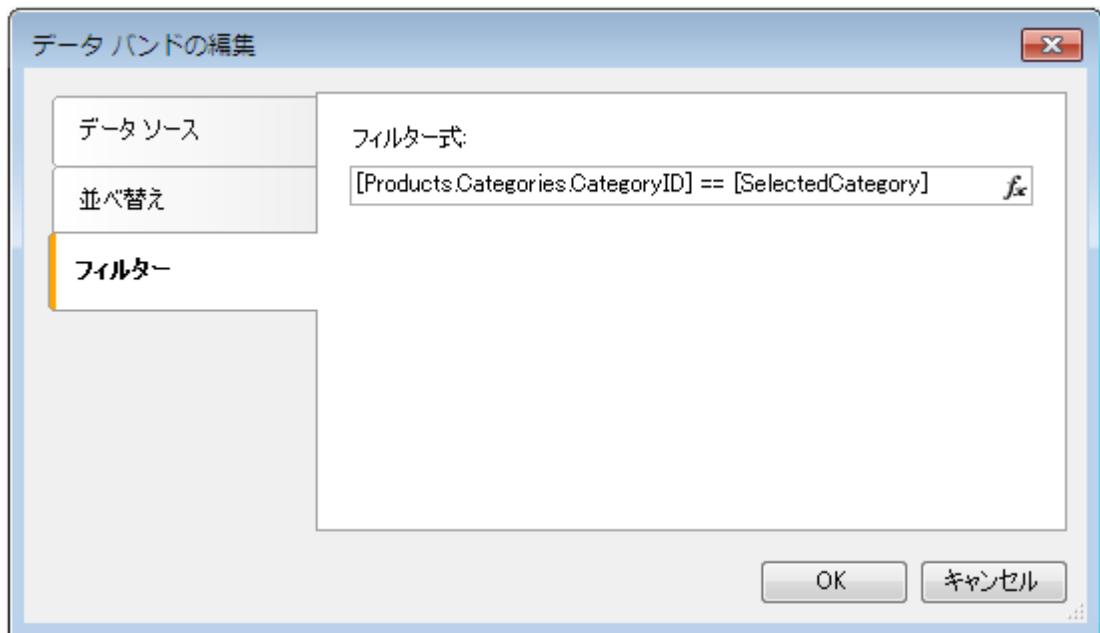
まず、製品の一覧を出力する詳細レポートを作成します。このため、新しいレポートを作成し、データソースとして"Products"を選択します。次のようにオブジェクトを配置します。

レポート タイトル	-
ページ ヘッダー	<b>Product Name</b> [Products.ProductName] [Products.QuantityPerUnit] [Products.UnitPrice]
データ: Products	[Products.ProductName] [Products.QuantityPerUnit] [Products.UnitPrice]

選択したカテゴリをメインレポートから詳細レポートへ渡すために使用するパラメーターを作成します。カテゴリの識別のため、Categories およびProducts テーブルの両方に含まれる"CategoryID"を使用します。次のようにパラメーターを設定します。



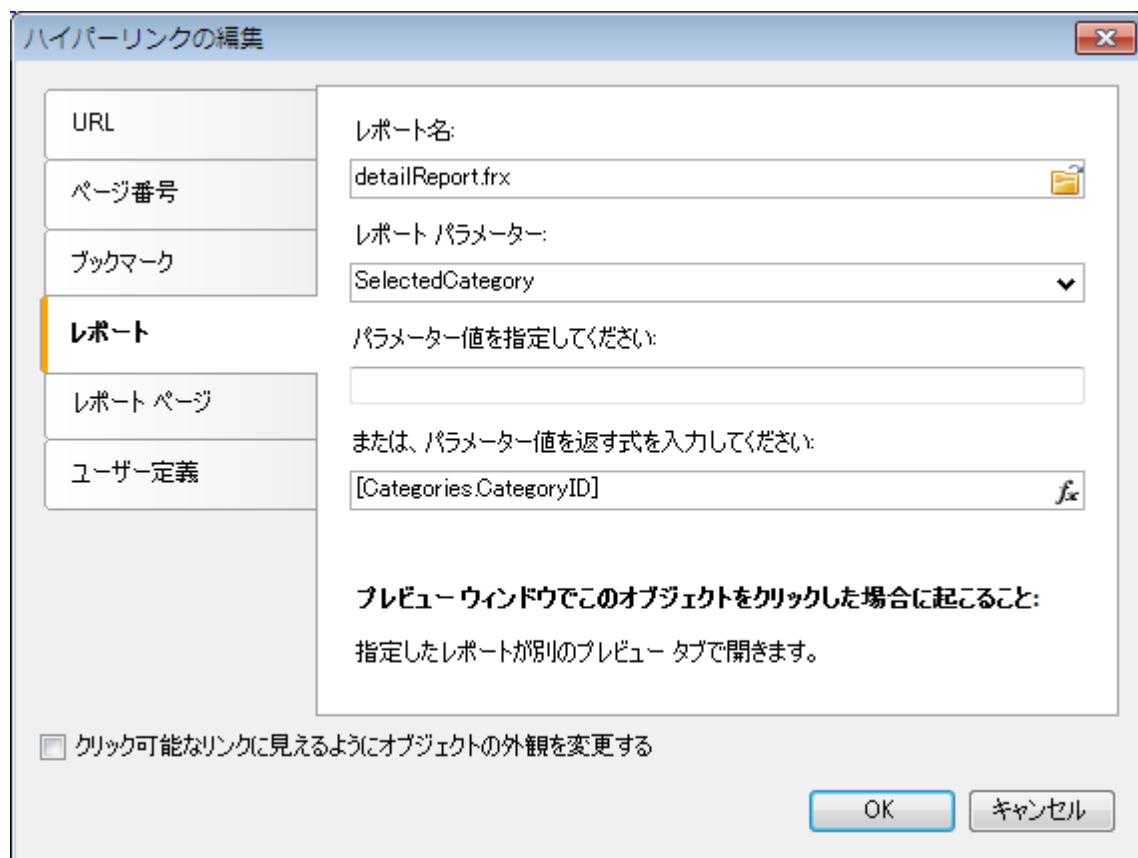
データフィルターを設定して、指定されたカテゴリに属する全製品をフィルター処理する必要があります。これを行うには、データバンドをダブルクリックします。[フィルター]タブを開き、次のような条件を指定します。



次に、メインレポートを作成します。新しいレポートを作成し、データソースとして"Categories" テーブルを選択します。次のようにオブジェクトを配置します。



このテキストオブジェクトを右クリックし、コンテキストメニューから[ハイパーリンク]を選択します。次のようにリンクを設定します。



レポート名として、詳細レポートファイルの名前を選択します。レポートパラメーターは、フィールドの右端にあるボタンをクリックして開くドロップダウンリストから選択できます。パラメーター値として、"[Categories.CategoryID]" という式を指定します。

レポートを実行すると、次のようにカテゴリの一覧が表示されます。



カテゴリの1つをクリックすると、詳細レポートが作成されます。このレポートはプレビュー ウィンドウの別のタブに表示されます。

詳細レポートを開じます

詳細レポートタブ

Product Name	Quantity Per Unit	UnitPrice
Chocolade	10 pkgs.	1,275
Gumbär Gummibärchen	100 - 250 g bags	3,123
Maxilaku	24 - 50 g pkgs.	2,000
NuNuCa Nuß-Nougat-Creme	20 - 450 g glasses	1,400
Pavlova	32 - 500 g boxes	1,745
Schoggi Schokolade	100 - 100 g pieces	4,390
Scottish Longbreads	10 boxes x 8 pieces	1,250
Sir Rodney's Marmalade	30 gift boxes	8,100
Sir Rodney's Scones	24 pkgs. x 4 pieces	1,000
Tarte au sucre	48 pies	4,930
Teatime Chocolate Biscuits	10 boxes x 12 pieces	920
Valkoinen suklaa	12 - 100 g bars	1,625
Zaanse koeken	10 - 4 oz boxes	950

上の図では、タブのタイトルにハイパーリンクの値が設定されています。この場合は、CategoryID データ列に含まれる数値です。これでは意味がわからにくく、体裁もよくありません。数値ではなく、カテゴリ名を使用するようにレポートを変更してみましょう。このため、以下のことを行い

ます。

詳細レポートで、以下のことを行います。

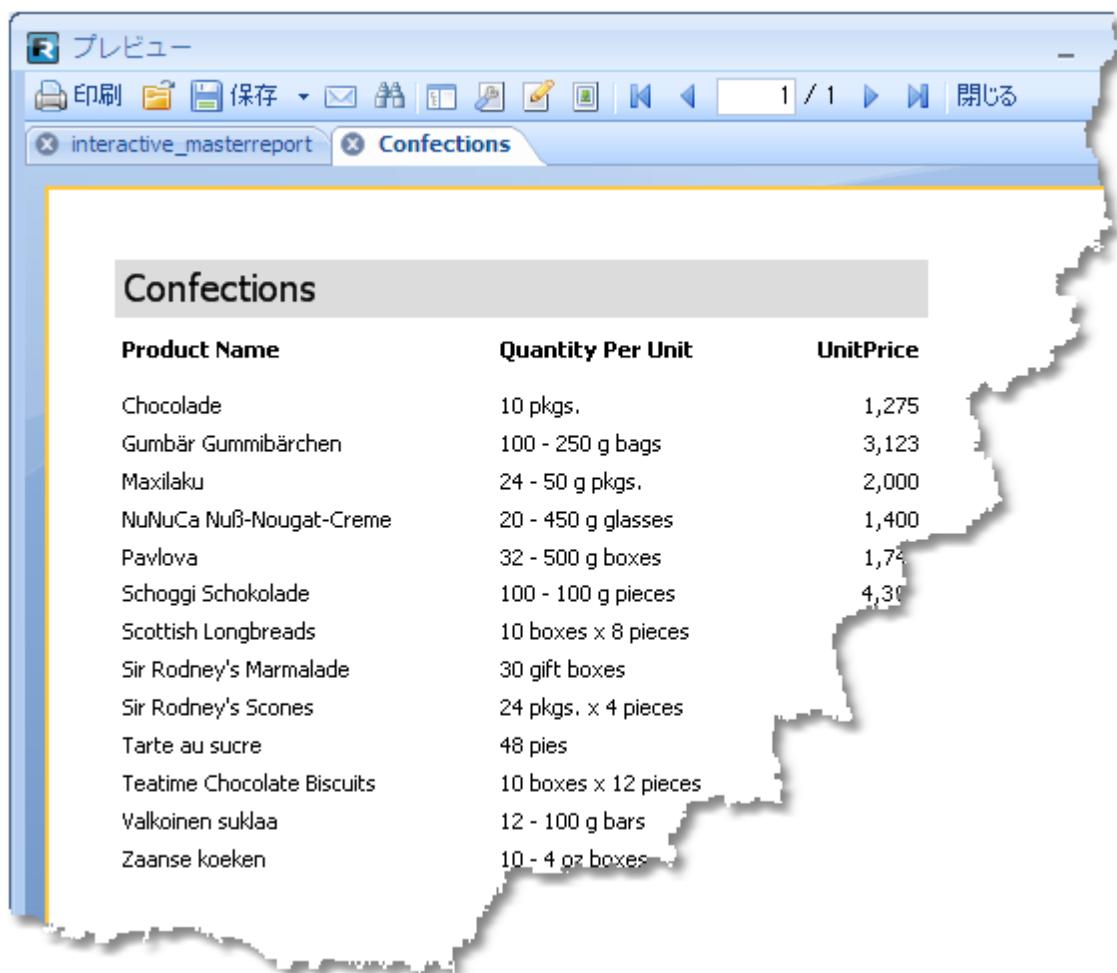
- パラメーターの DataType プロパティを "String" に変更します。
- Categories データソースをレポートに追加します。これは、データのフィルター処理時、CategoryName データ列を参照するために使用します。
- データバンドのフィルター式を次のように変更します。

```
[Products.Categories.CategoryName] == [SelectedCategory]
```

メインレポートで、以下のことを行います。

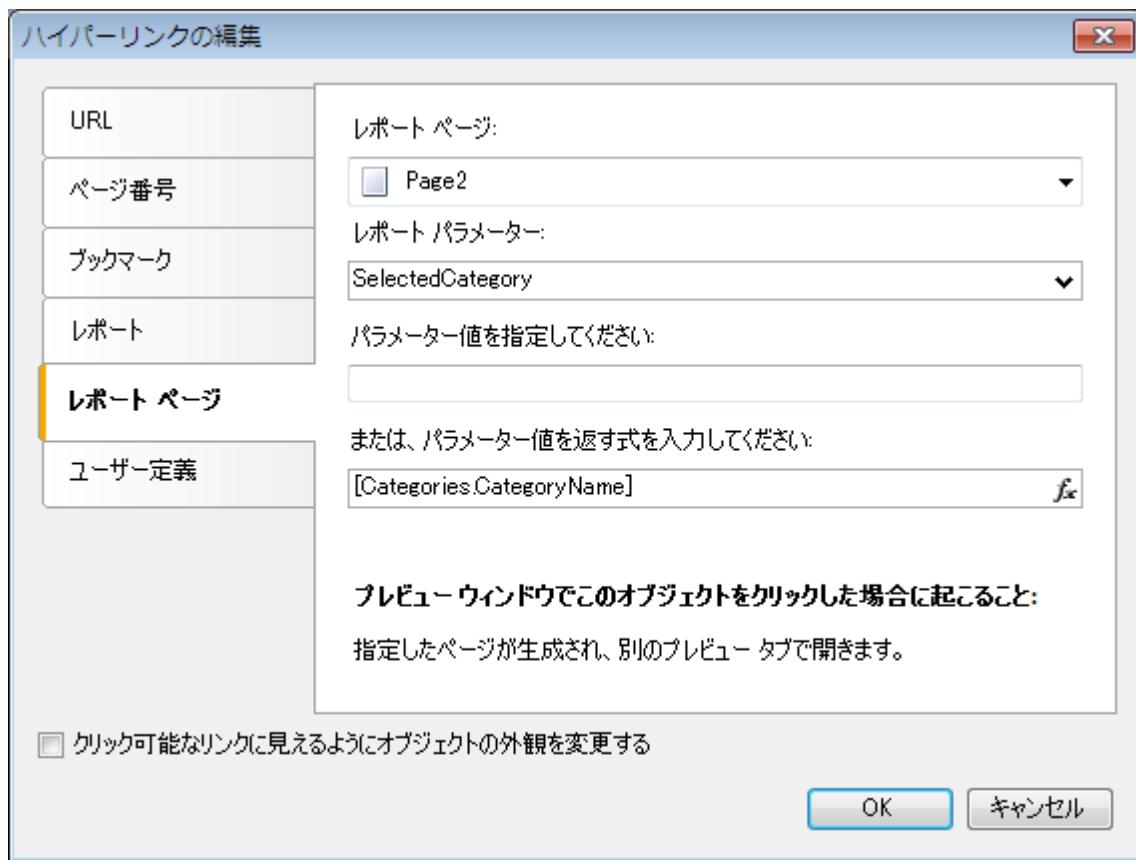
- ハイパーリンクの設定を変更します。"[Categories.CategoryName]" 値をレポートパラメーターへ渡します。

レポートを実行すると、タブのタイトルにカテゴリ名が設定されるようになります。詳細レポートの外観を少し改良します。選択したカテゴリの名前を出力するテキストオブジェクトをレポートタイトルレポートに追加します。



Product Name	Quantity Per Unit	UnitPrice
Chocolade	10 pkgs.	1,275
Gumbär Gummibärchen	100 - 250 g bags	3,123
Maxilaku	24 - 50 g pkgs.	2,000
NuNuCa Nuß-Nougat-Creme	20 - 450 g glasses	1,400
Pavlova	32 - 500 g boxes	1,74
Schoggi Schokolade	100 - 100 g pieces	4,30
Scottish Longbreads	10 boxes x 8 pieces	
Sir Rodney's Marmalade	30 gift boxes	
Sir Rodney's Scones	24 pkgs. x 4 pieces	
Tarte au sucre	48 pies	
Teatime Chocolate Biscuits	10 boxes x 12 pieces	
Valkoinen suklaa	12 - 100 g bars	
Zaanse koeken	10 - 4 oz boxes	

この例では、2つのレポートを作成し、作業に応じてそれぞれのレポートをその都度開いています。これは効率がよくありません。このタスクをもっと簡単にするには、2つのレポートを1つのレポートにまとめ、最初のページにメインレポート、2番目のページには詳細レポートを置くようにします。この場合、ハイパーリンクは次のように設定する必要があります。



ここでは、詳細レポートページとして"Page2"を選択する必要があります。

### 例 3 対話型の行列オブジェクト

この例では、行列オブジェクトのセルをクリックした場合に詳細レポートを作成する方法を示します。例として、従業員別の年間売上を表示する行列を使用します。

行列のデータソースとしてMatrixDemo テーブルを使用します。従業員の売上を年単位および月単位で表します。

Name	Year	Month	Items Sold	Revenue
Nancy Davolio	1999	2	1	100000
Nancy Davolio	1999	11	1	110000
Nancy Davolio	1999	12	1	120000

Nancy Davolio	2000	1	1	130000
Nancy Davolio	2000	2	2	140000
Nancy Davolio	2001	2	2	150000
Nancy Davolio	2001	3	2	160000
Nancy Davolio	2002	1	2	170000
Andrew Fuller	2002	1	2	180000
Andrew Fuller	1999	10	2	190000
Andrew Fuller	1999	11	2	200000
Andrew Fuller	2000	2	2	210000
Janet Leverling	1999	10	3	300000
Janet Leverling	1999	11	3	310000
Janet Leverling	200000	3	3	320000
Steven Buchanan	2001	1	3	400000
Steven Buchanan	2001	2	4	410000
Steven Buchanan	2000	1	4	399900

次のように行列を設定します。

- MatrixDemo.Year データ列を列ヘッダーに置きます。
- MatrixDemo.Name データ列を行ヘッダーに置きます。
- MatrixDemo.Revenue データ列をセルに置きます。

実行される行列は次のようになります。

Employee	Year				
	1999	2000	2001	2002	集計
Andrew Fuller	390,000	210,000		180,000	780,000
Janet Leverling	610,000	320,000			930,000
Nancy Davolio	330,000	270,000	310,000	170,000	1,080,000
Steven Buchanan		399,900	810,000		1,209,900
集計	1,330,000	1,199,900	1,120,000	350,000	3,999,900

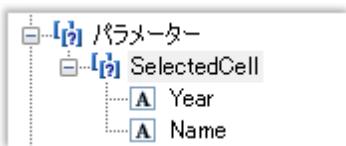
セルの値は各従業員の年間売上を示しています。次に、このセルをクリックしたときに表示される詳細レポートを作成してみましょう。詳細レポートには、選択した従業員の、指定した年の月別売上を含めることにします。

出力されるデータとセルはどのように接続されるのでしょうか。行列のセルはそれぞれ自身のアドレスを持っています。これは、列ヘッダーおよび行ヘッダーの値を組み合わせたものです。この例の場合、セルのアドレスは年および従業員名の組み合わせです。このデータを正確に詳細レポートへ渡すことができます。これを行う方法は、極めて簡単です。レポート名とパラメーターの名前のみを表示するハイパーリンクを設定します。パラメーターの値を指定する必要はありません。行列のセル用に FastReport 自身が値を作成し、それをパラメーターに渡します。

行列内の左上で 390,000 という数字が含まれるセルをクリックすると仮定します。これは "Andrew Fuller" という名前の従業員の 1999 年における年間売上です。この値をパラメーターに渡すためには特定の形式を使用します。FastReport は、列および行の値を次のように区切り文字を使用して結合します。

1999;Andrew Fuller

これは、この文字列から年と従業員名の値を取り出し、年を int 型に変換し、それらの値をデータのフィルター処理に使用するということではありません。作業はもっと簡単で、**入れ子のパラメーターを持つパラメーターを作成するだけ**です。この詳細については、「データ」の章の [「パラメーターの作成」](#) を参照してください。この例では、親パラメーターは次のようになります。



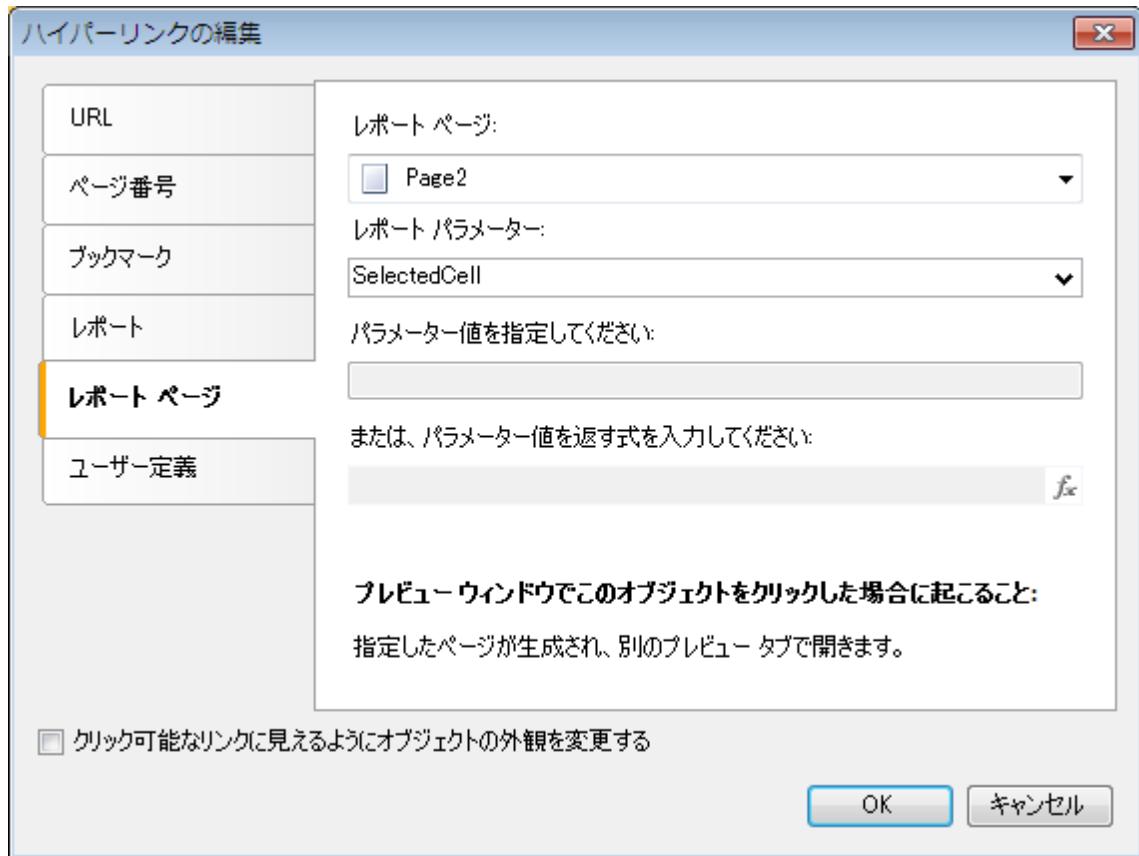
パラメーターを作成する場合は、以下の点に留意してください。

- 親パラメーターを設定する必要はありません。そのパラメーターの名前を指定するだけです。
- 親パラメーターは行列から渡される値と同数の入れ子のパラメーターを持つ必要があります。この例の場合は、2 つの値があります。
- 入れ子のパラメーターの順序は、行列から渡される値の順序と一致させる必要があります。この例の場合は、Year (年) が 1 番目のパラメーターに渡され、Name (従業員名) が 2 番目のパラメーターに渡されます。
- 入れ子のパラメーターには自由に名前を付けることができますが、行列の要素の名前と一致する名前にすることをお勧めします。
- 入れ子のパラメーターに正しいデータ型を設定しておくことが極めて重要です。このデータ型はパラメーターに渡される値の型と一致している必要があります。この例の場合、1 番目のパラメーター (Year) は整数型で、2 番目のパラメーター (Name) は文字列型です。

これらの必要事項が確認できたら、レポートを作成します。次のように、行列のセルを選択し、コンテキストメニューからハイパーリンクエディターを呼び出します。

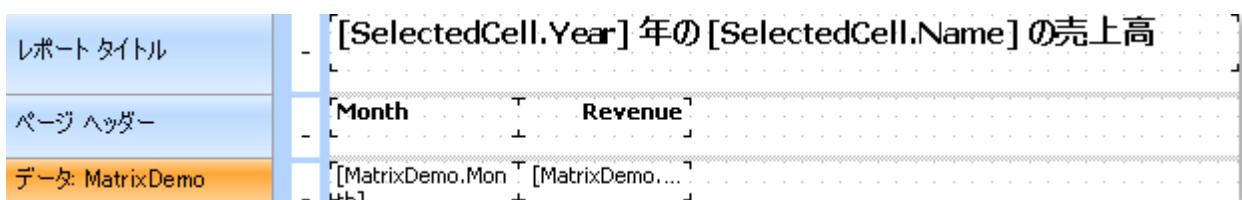


ハイパーリンクの設定で、レポートパラメーターとして親パラメーター指定します(ここでは"SelectedCell" )。



FastReport は値を入れ子のパラメーターである SelectedCell.Year と SelectedCell.Name に渡します。これらの値は、パラメーターの設定で指定されたデータ型に変換されます。このため、パラメーターのデータ型を正しく設定しておくことが重要です。

詳細レポートはメインレポートの別のページに置かれ、同じデータソースを使用します。



選択した従業員の、選択した年の年間売上を表示させるためには、フィルターを設定します。このため、データバンドエディターを開き、[フィルター]タブで次のフィルター条件を指定します。

```
[MatrixDemo.Year] == [SelectedCell.Year] && [MatrixDemo.Name] == [SelectedCell.Name]
```

レポートのデザインが完了しました。レポートを実行して左上のセル(390,000)をクリックします。詳細レポートが開かれ、次のようなデータが表示されます。

## 1999 年の Andrew Fuller の売上高

Month	Revenue
10	190,000
11	200,000

Revenue の各値 (190,000 と200,000 )の合計は 元の行列でクリックしたセルの値 (390,000 )と一致しています。

## 例 4 :目次、ナビゲーションおよびアウトライン付きのレポート

この例では 以下の機能を持つレポートを作成します。

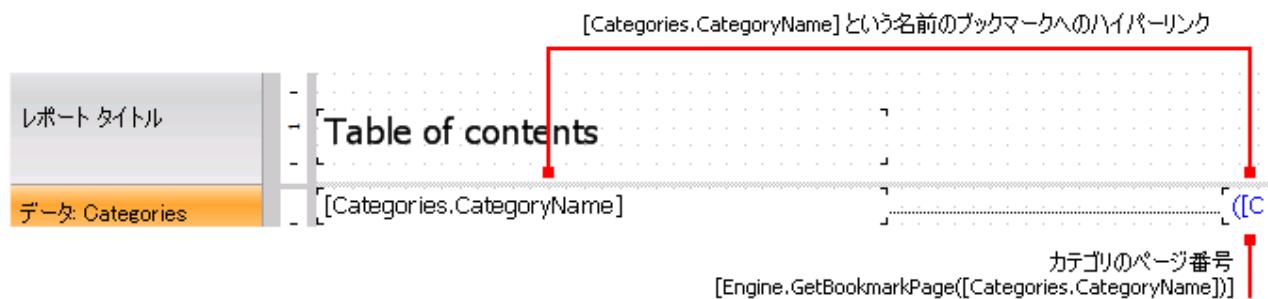
- 最初のページには対話型の目次を出力します。目次の要素をクリックすると 対応するページに移動することができます。
- プレビュー ウィンドウで、アウトラインを表示します。これも同様に対話型です。

The screenshot shows the Microsoft Report Designer interface. On the left, a 'Table of Contents' pane lists categories: Beverages, Condiments, Confections, Dairy Products, Grains/Cereals, Meat/Poultry, Produce, and Seafood. The 'Beverages' section is expanded, showing sub-categories like Soft drinks, coffees, teas, beers, and wine. The main report area displays two tables. The first table, titled 'Beverages', shows a list of products with columns for Unit price, Units in stock, and Discontinued. The second table, titled 'Condiments', also shows a list of products with similar columns. Both tables include a 'View details' link. The bottom of the screen shows the report preview, a status bar with '準備時間: 1437 ミリ秒', and a page number 'ページ 1 / 5'.

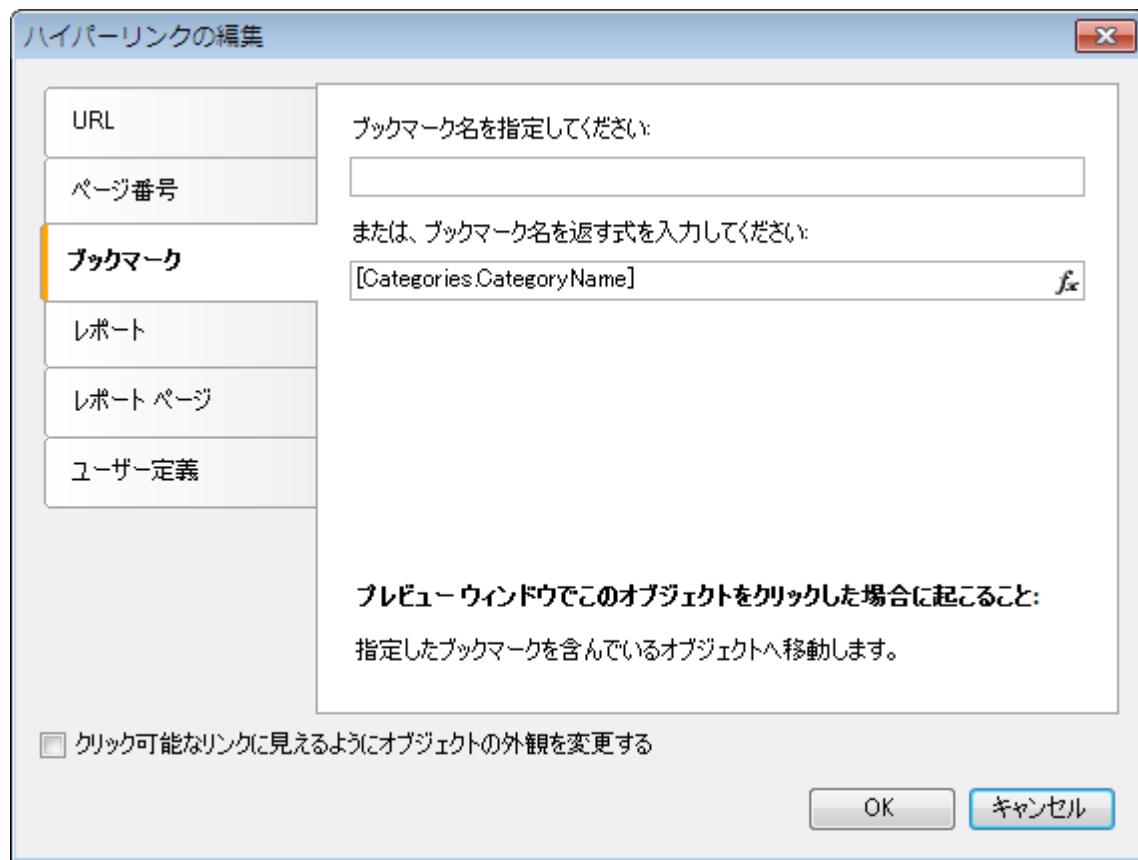
このレポートではCategories およびProducts テーブルを使用します。目次にはカテゴリの一覧を出力します。それ以外のレポート部分にはカテゴリ別の製品リストを出力します。レポートテンプレートは2 つのページで構成されます。最初のページは目次の出力に使用し、2 番目のページはレポートメイン部分として使用します。

まずは目次について説明します。新しいレポートを作成し、データソースとしてCategories およびProducts を追加します。データバンド

をCategories テーブルに接続し、次のようにオブジェクトを配置します。



目次オブジェクトを対話型にするには、そのHyperlink プロパティを設定します。



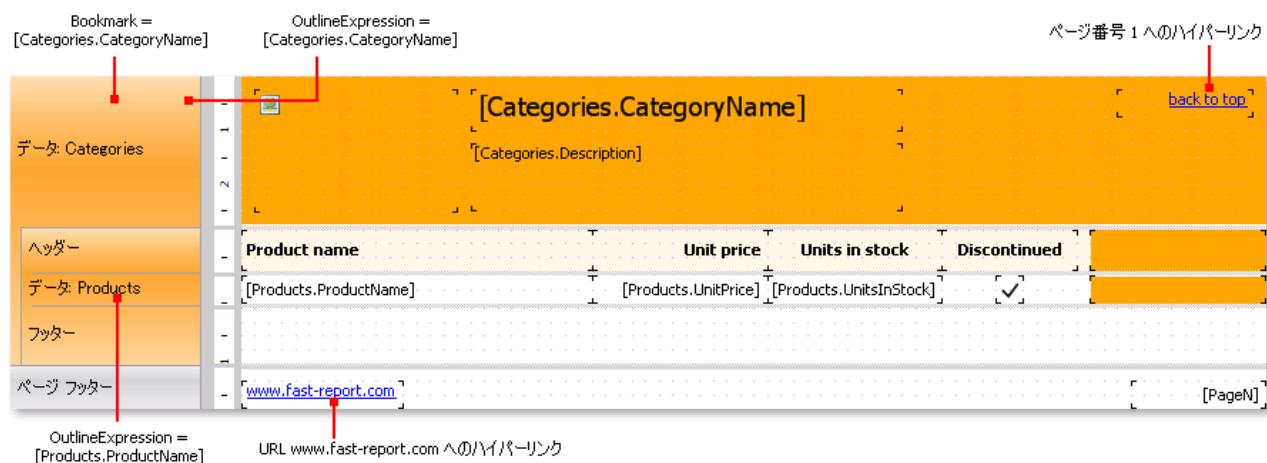
カテゴリ名をブックマークとして指定します。オブジェクトのブックマークは後で定義します。

目次にページ番号を出力するには、以下のことを行う必要があります。

- レポートのダブルパス設定を有効にします。これは[レポート]メニューの[オプション]を選択して行います。目次ページはほかのページよりも先に出力されるため、これが必要です。この時点で、FastReport はカテゴリがどこに出力されるかわかつていません。
- Engine.GetBookmarkPage 関数を使用します。これは指定されたブックマークのページ番号を返します。ここでは、ブックマーク名として"[Categories.CategoryName]" という式を使用します。関数は次のようになります。

[Engine.GetBookmarkPage([Categories.CategoryName])]

レポートの2番目のページで、下図のようなマスター/詳細レポートを作成します。



目次内の要素をクリックしたときに移動するブックマークを設定します。このため、最初のデータバンドを選択して、Bookmark プロパティに次の式を指定します。

[Categories.CategoryName]

レポートアウトラインを設定するには、以下の手順に従います。

- 最初のレポートページを選択します。そのページに切り替われば(開けば)ページが選択されます。
- 「プロパティ」ウインドウで、OutlineExpression プロパティに次の値を設定します。

"Contents"

- 2番目のページに切り替えます。
- 最初のデータバンドを選択し、そのOutlineExpression プロパティに次の値を設定します。

[Categories.CategoryName]

- 2番目のデータバンドを選択し、そのOutlineExpression プロパティに次の値を設定します。

[Products.ProductName]

## レポートの継承

同じデータ(会社のロゴ、社名、電子メールアドレスなど同じ内容のヘッダー/フッター)を含む多くのレポートが存在する場合があります。ここで、電子メールアドレスなど会社の情報を変更する必要がある状況を想像してみましょう。その場合、レポートごとにこの変更を行う必要があります。これを回避するため、レポートの継承を使用することができます。

たとえば、各レポートに、共通する要素(ロゴ、会社名、電子メールなど)があるとします。一般に、これらの要素はレポートタイトルまたはページヘッダー上に置かれます。このため、共通する要素のみを含む基本レポートを作成することができます。それ以外のレポートではその基本レポートを使用するため、その共通要素に加えて、レポート内で定義した独自の要素が含まれます。

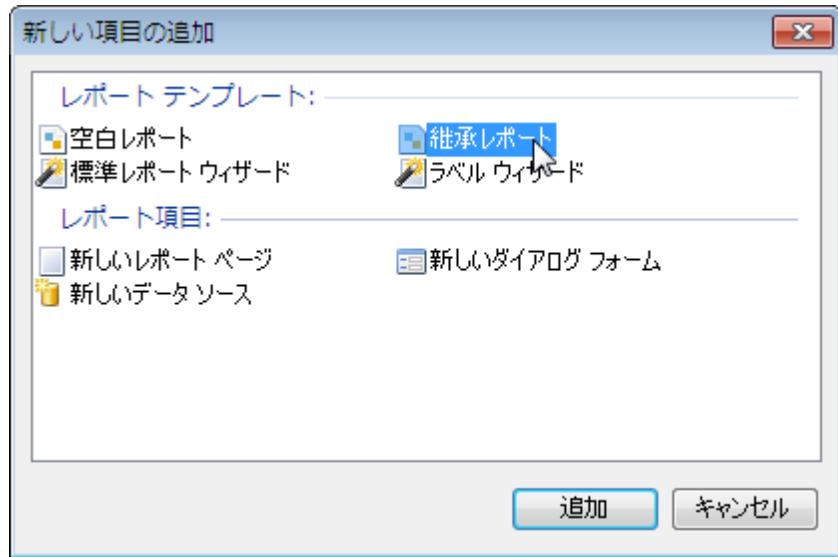
共通する要素(ロゴや電子メール)を変更する必要がある場合は、基本レポートを開いて必要な変更を行なうオペレーターです。基本レポートから継承されるその他すべてのレポートは自動的に変更されます。実際、継承されるレポートを開く場合は、先に基本レポートが開かれ、その後に継承レポートが開かれます。

## レポートの作成

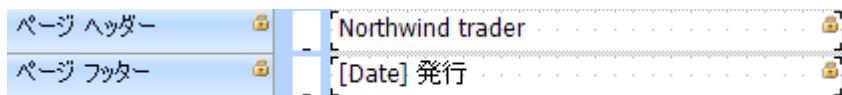
継承を使用するには、以下のことを行う必要があります。

- 基本レポートを作成し、それをファイルに保存します。
- 基本レポートを継承する新しいレポートを作成します。

継承レポートを作成するには、[ファイル]メニューの[新規作成]を選択し、[新しい項目の追加]ウィンドウから[継承レポート]を選択します。



基本レポートファイルの選択を求められます。そのファイルはこの時点で既に作成されている必要があります。ファイルを選択すると、その基本レポートがデザイナーに読み込まれます。目的に応じてそのレポートを変更することができます。基本レポートから継承されるオブジェクトには、次のようにロックアイコンが付けられています。



これは、そのオブジェクトの削除、名前の変更または別のバンドへの移動が行えないことを意味します。

新しいオブジェクトの追加、オブジェクトの外観、サイズおよび場所の変更は可能です。作業が終了したら、そのレポートを保存します。

## 基本レポートの変更

基本レポートを変更する場合に起こる現象について説明します。以下の変更を行うことができます。

- 基本レポートからオブジェクトを削除します。このオブジェクトは、継承レポートからも削除されます。
- オブジェクトを基本レポートに追加します。このオブジェクトは、自動的に継承レポートにも追加されます。
- オブジェクトのサイズ、位置、テキスト、外観を変更します。このオブジェクトが継承レポート側で変更されなければ、すべての変更が継承レポートに反映されます。

最後の項目には説明が必要です。継承を使用する2つの例を見てみましょう。最初の例では、以下のことを行います。

- Text1 オブジェクトを含む基本レポートを作成します。
- 継承レポートを作成し、変更することなく保存します。
- 基本レポートを開き、Text1 オブジェクトの位置を変更します。
- 継承レポートを開くと、Text1 オブジェクトの位置が同様に変更されています。

もう1つの例では、以下のことを行います。

- Text1 オブジェクトを含む基本レポートを作成します。
- 継承レポートを作成します。
- 継承レポートで、Text1 オブジェクトの位置を変更し、レポートを保存します。
- 基本レポートを開き、Text1 オブジェクトの位置を（継承レポートとは異なる位置に）変更します。
- 継承レポートを開くと、Text1 オブジェクトの位置は変更されていません。

位置の変更が反映されないのは、継承レポートでオブジェクトが変更されていたためです。この変更は、継承レポートファイルに保存されています。ここで基本レポートで元のオブジェクトを変更すると、その変更は継承レポートで無視されます。この場合は、新しいオブジェクトの位置が無視されます。それ以外の変更（テキストの色など）はすべて継承レポートに反映されます。

この動作は、継承レポートファイルの内容を見るとよくわかります。たとえば、元のオブジェクトが変更されなかつた場合、継承レポートにはそのオブジェクトが次のように保存されます。

```
<inherited Name="Text1"/>
```

継承レポートでオブジェクトの位置を変更すると、そのオブジェクトは次のように保存されます。

```
<inherited Name="Text1" Left="255.15" Top="28.35"/>
```

継承レポートを開いたとき、FastReport は基本レポートで定義されたオブジェクトのプロパティをすべて読み込み、さらに継承レポートに保存されているオブジェクトのプロパティをすべて読み込みます。

## 制限事項

レポートの継承は、ヘッダー/フッターなどレポートに共通する要素を別々のファイルに保存すること、またそれらの要素を継承レポートで再利用することを目的に設計されます。より複雑なタスクを行う場合は継承を使用しないでください。特に、以下のことをご注意ください。

- 継承レポートからレポートを継承しない（継承を2回にわたって行わない）
- 基本レポートで、表や行列などの複雑なオブジェクトを使用しない
- 基本レポートでスクリプトを使用しない
- 基本レポートでパラメーターを使用しない

## グラフ付きのレポート

FastReport は Microsoft Chart ライブリを使用してグラフを表示します。このライブリは .Net Framework 4.0 に含まれています。これは次のサイトから単独でダウンロードすることもできます。

<http://www.microsoft.com/downloads/details.aspx?FamilyID=130f7986-bf49-4fe5-9ca8-910ae6ea442c&displayLang=ja>

このライブリは .Net Framework 3.5 SP1 が必要です。これはフリーウェアです。次のサイト(英語)ではこの機能について学習することができます。

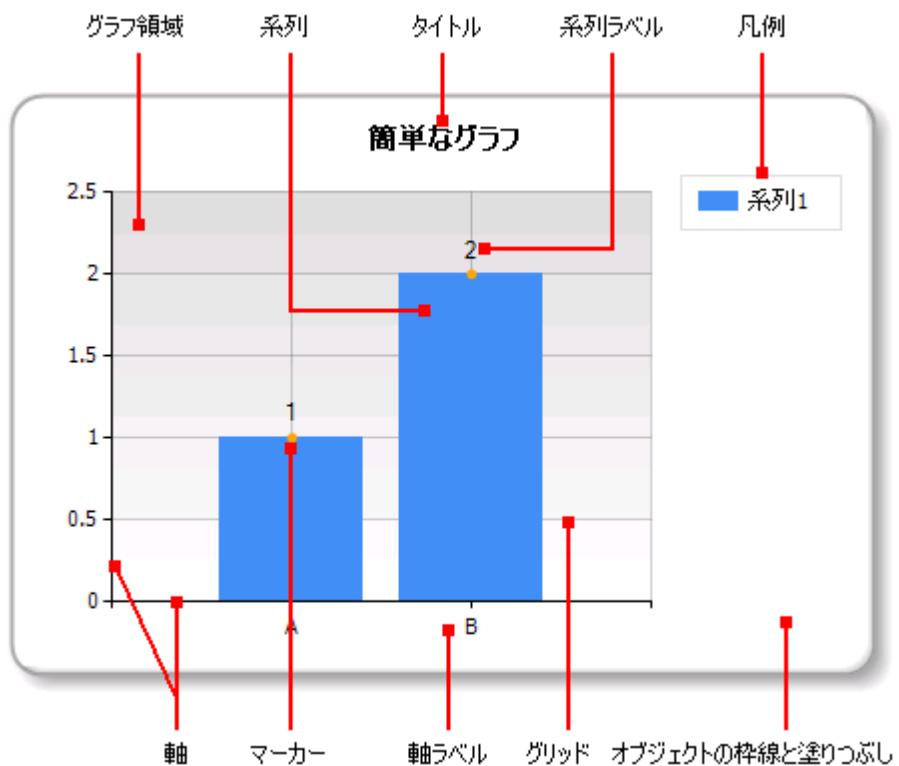
<http://code.msdn.microsoft.com/mschart>

Microsoft Chart のいくつかの機能に注目してみましょう

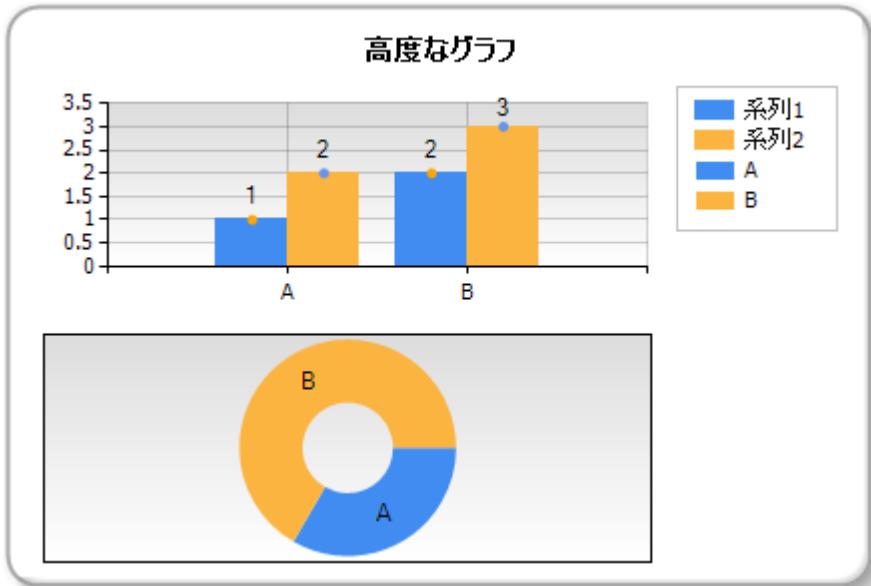
- 30 種類以上の系列タイプ(棒、列、領域、折れ線、バブル、円、円形、財務、ピラミッド、範囲)
- 3D 対応
- 1 つのグラフ内で異なる系列タイプの使用をサポート
- グラフの各要素に対する外観と動作の完全な制御

## グラフの要素

Microsoft Chart には以下の要素で構成されます。



1 つのグラフに1 つまたは複数のグラフ領域を持つことができます。1 つのグラフ領域に1 つまたは複数の系列を持つことができます。次の図では、1 つのグラフに2 つのグラフ領域が含まれています(最初の領域には2 つの系列があり、2 番目の領域には1 つの系列があります)。

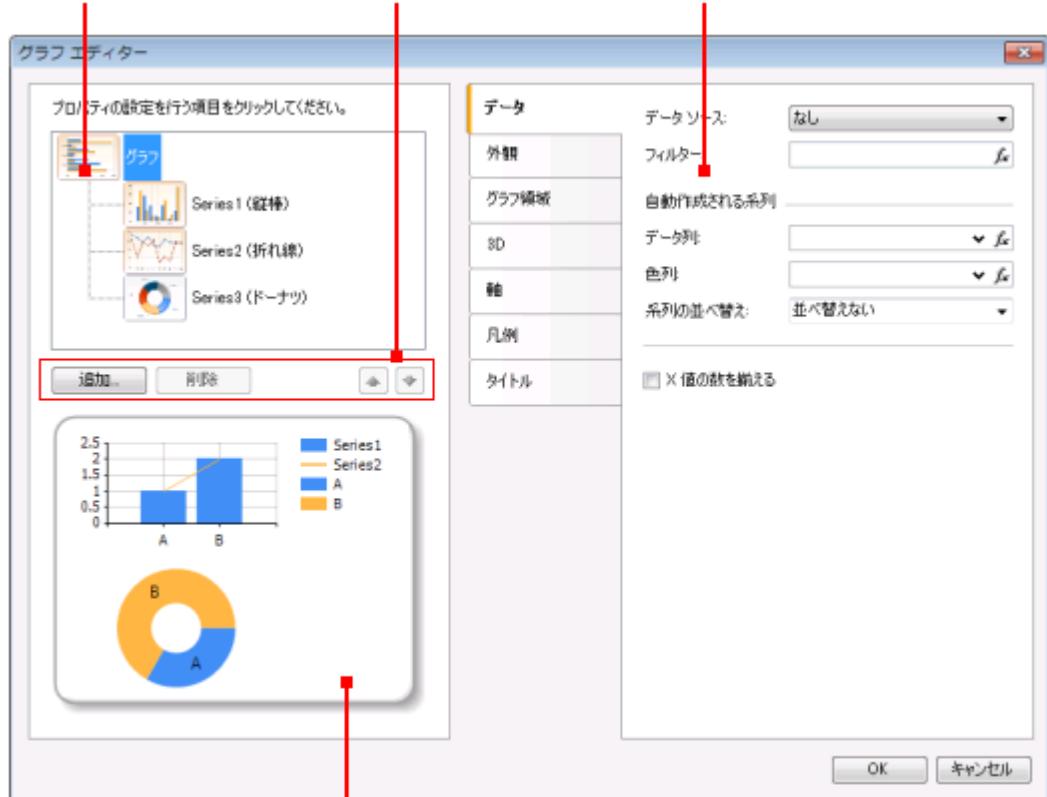


系列(円系列など)のタイプによっては、排他的なグラフ領域が必要となることがあります。

## グラフ エディター

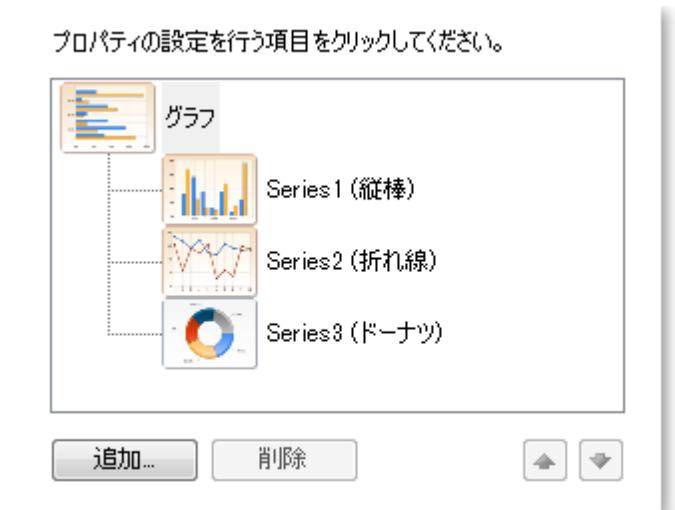
グラフオブジェクトはグラフエディターで処理できる多くの設定があります。グラフエディターを起動するには、グラフオブジェクトをダブルクリックします。

グラフとその系列 系列を処理します 選択した要素(グラフまたは系列)を編集します

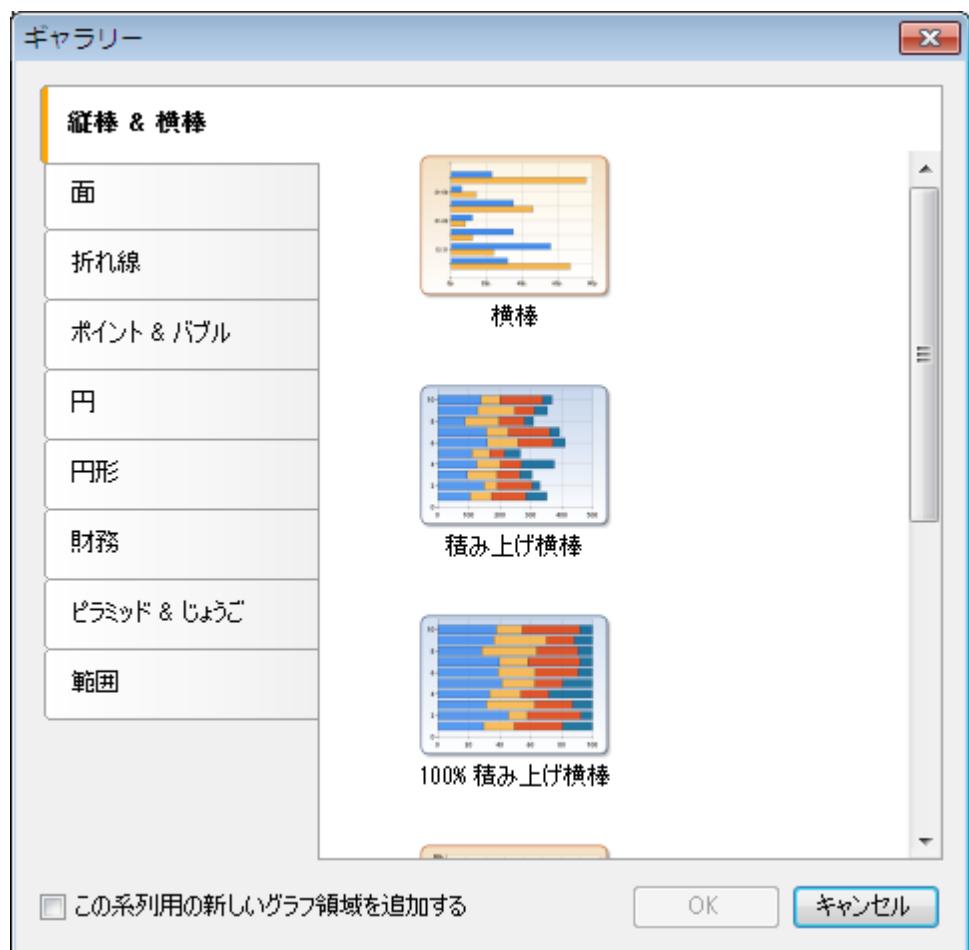


## 系列の処理

MS Chart オブジェクトには1つまたは複数の系列を含めることができます。系列の一覧はグラフエディターで表示されます。



新しい系列を追加するには、[追加] ボタンを押します。[ギャラリー] ダイアログが表示されます。



グラフのカテゴリを選択し、系列タイプを選択します。その系列を独自のグラフ領域に置きた場合は、[この系列用の新しいグラフ領域を追加する]チェックボックスをオンにします。系列タイプによっては(円、円形、財務、ピラミッドなど)、このチェックボックスの状態に関わらず、新しいグラフ領域が自動的に追加されます。

系列を削除するには、[削除]ボタンを押します。系列の順序を変更するには、上矢印および下矢印ボタンを使用します。

## 外観の設定

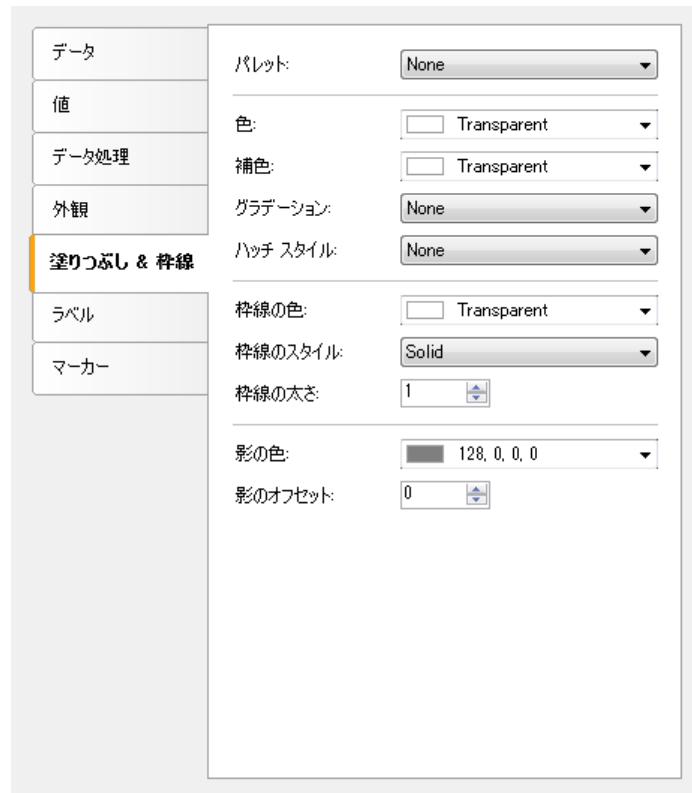
グラフエディターを使用して、グラフの各要素の外観を設定することができます。すべてのプロパティ(100個以上)がカテゴリ別に分けられています。プロパティにはグラフオブジェクトにのみ適用されるものと、系列にのみ適用されるものがあります。

系列リストから"グラフ オブジェクト"を選択すると、以下のプロパティページが表示されます。



- [外観]- グラフの枠線および塗りつぶし
- [グラフ領域]- 枠線、塗りつぶし、影
- [3D]- 3D 設定
- [軸]- 軸、そのタイトル、ラベル、グリッド、マーカー、カスタム ラベルおよびストリップの外観
- [凡例]- 凡例のスタイル、ドッキング、枠線、塗りつぶし、影およびフォント
- [タイトル]- タイトルのスタイル、ドッキング、枠線、塗りつぶし、影およびフォント

系列リストから"系列" オブジェクトを選択すると、以下のプロパティページが表示されます。



- [外観]- 選択した系列に関する設定
- [塗りつぶし&枠線]- 系列の値の塗りつぶしおよび枠線
- [ラベル]- 系列ラベル。ラベルのタイプ、フォント、色および塗りつぶしを選択できます。
- [マーカー]- 系列マーカー。マーカーのタイプ、色および枠線を選択できます。

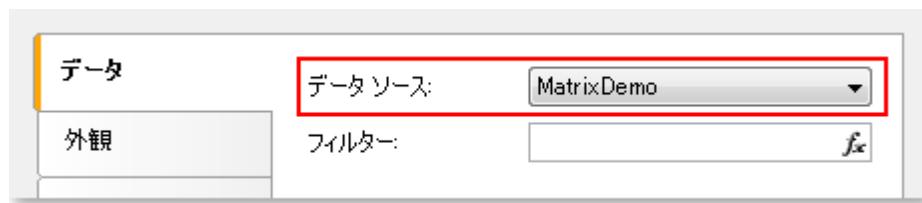
## グラフをデータへ接続する

グラフにデータを取り込む方法はいくつあります。

1. データソースを使用します。これを行うには、グラフオブジェクトに対するデータソースを指定し、各系列をデータ列に接続します。
2. 各系列に固定の値を使用します。
3. スクリプトを使用して、データをオブジェクトに取り込みます。

グラフをデータソースへ接続するには、以下の手順に従います。

- 系列項目の一覧で「グラフ」を選択します。
- [データ]タブを表示します。
- データソースを選択します。

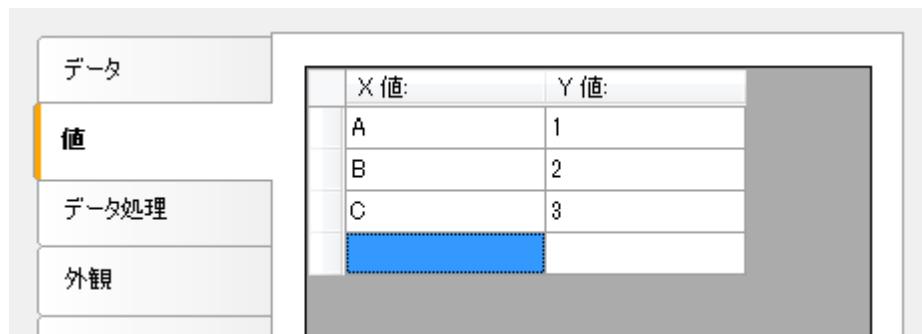


- 必要に応じて、データのフィルター式を設定します。このフィルターはすべてのグラフ系列に適用されます。
- 系列項目の一覧で系列を選択します。
- [データ]タブを表示します。
- 各系列値に対応するデータ列を選択します。系列のタイプによって、2つ以上の値を持つこともあります。多くの系列タイプはX値およびY値という2つの値を持ちます。



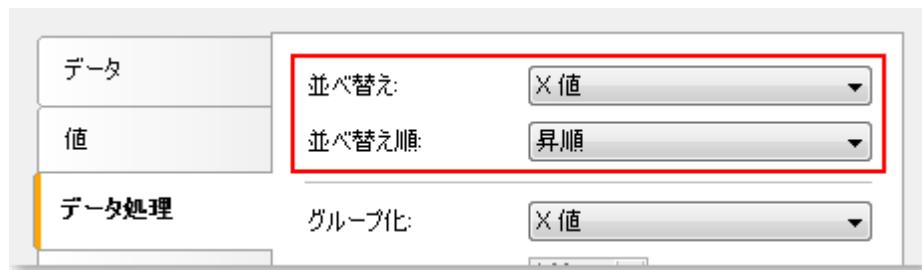
- 必要に応じて、データのフィルター式を設定します。このフィルターは、現在選択している系列にのみ適用されます。
- [色]コントロールでは、色の値を返すデータ列を指定できます。

系列に対する値のリストを提供することもできます。この場合、データソースへの接続は必要ありません。これを行うには、系列項目の一覧で対象の系列を選択し、[値]タブを表示します。ここで値を表に入力します。



## データの並べ替え

既定では、グラフオブジェクトはデータを自然な順序で表示します。この並び順は変更できます。これを行うには、系列の一覧から変更対象の系列を選択し、[データ処理]タブを表示します。



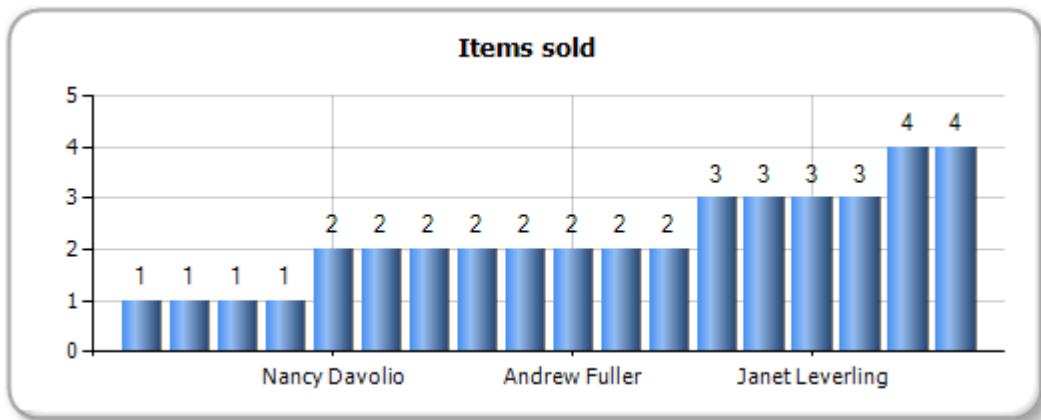
並べ替えモード("並べ替えなし"、"X 値" または"Y 値" )を選択できます。

## データのグループ化

系列に含まれるデータで、同一のX値が複数存在する場合は問題が発生します。たとえば、グラフのサンプルで使用されている MatrixDemo テーブルには以下のデータが含まれています。

Name	Year	Month	Items Sold	Revenue
Andrew Fuller	2002	1	2	180000
Andrew Fuller	1999	10	2	190000
Andrew Fuller	1999	11	2	200000
Andrew Fuller	2000	2	2	210000
Janet Leverling	1999	10	3	300000
Janet Leverling	1999	11	3	310000
Janet Leverling	2000	3	3	320000
...				

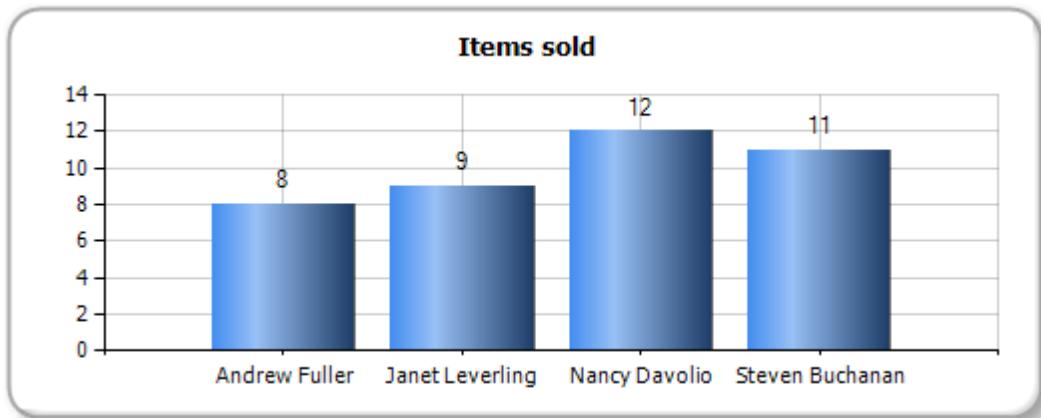
このデータに基づいてグラフを作成しようとした場合(たとえば、従業員別売上高 - X 値にName 列、Y 値にItems Sold 列を設定)、次のような誤った結果が出力されます。



この状況では、同じ従業員をグループ化して1つの値にまとめる必要があります。これを行うには、系列項目の一覧で対象の系列を選択し、「データ処理」タブを表示します。「グループ化」に「X値」を選択し、「グループ関数」に「合計」を選択します。



同一の従業員が1つの値としてグループ化され、その売上が集約されます。出力結果は次のようになります。



## データの収集

このデータ処理のインストレメンタ化により、複数の系列の値を1つの値に収集することができます。以下のアルゴリズムのいずれかを選択できます。

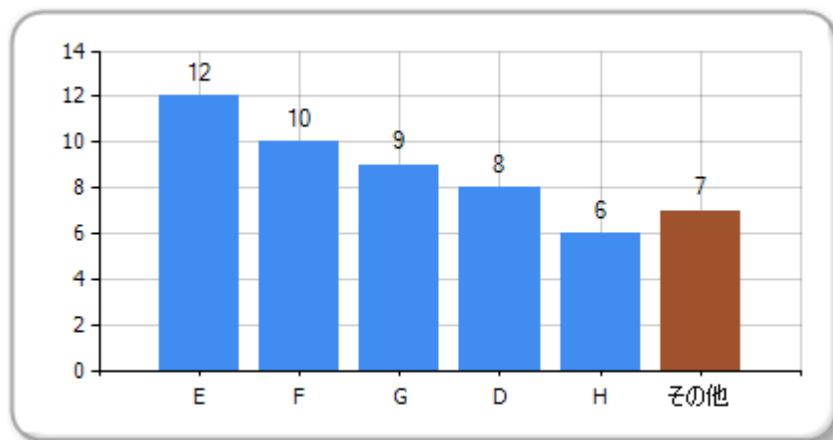
## アルゴリズム 説明

上位 N	上位 N の値のみが表示されます。それ以外の値はすべて収集され "その他" の値として表示されます (この値のラベルは指定できます)。
下位 N	下位 N の値が表示されます。収集された値に対するテキスト(収集データのテキスト)が設定されてない場合、この値は表示されません。
指定値より小さい	指定された値よりも系列の値が小さい場合、その値は収集され "その他" の値として表示されます。
指定パーセントより小さい	指定されたパーセントよりも系列の値が小さい場合、その値は収集され "その他" の値として表示されます。
指定値より大きい	指定された値よりも系列の値が大きい場合、その値は収集され "その他" の値として表示されます。
指定パーセントより大きい	指定されたパーセントよりも系列の値が大きい場合、その値は収集され "その他" の値として表示されます。

たとえば 上位 5 位までの値を表示するには 系列を次のように設定します。



出力結果は次のようにになります。



## 値の分割

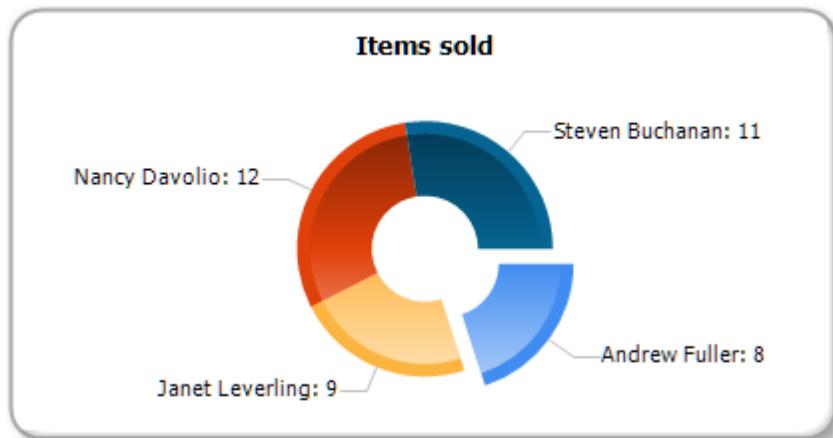
円タイプの系列の場合は、値を分割することができます。これを行うには、系列項目の一覧で対象の系列を選択し、[データ処理]タブを表示します。



分割タイプとして"最大値"、"最小値"、"指定値"の中からいずれかを選択できます。"指定値"モードを選択した場合は、分割する値を指定する必要があります。式も指定できます(詳細については[式の章を参照](#))。たとえば、Andrew Fullerの値を分割するには、次の式を使用します。

"Andrew Fuller"

出力結果は次のようにになります。



## 系列の自動作成

データソースのデータに応じて、新しい系列を自動的に作成するよなグラフを設定することができます。系列の自動作成を設定するには、以下の手順に従います。

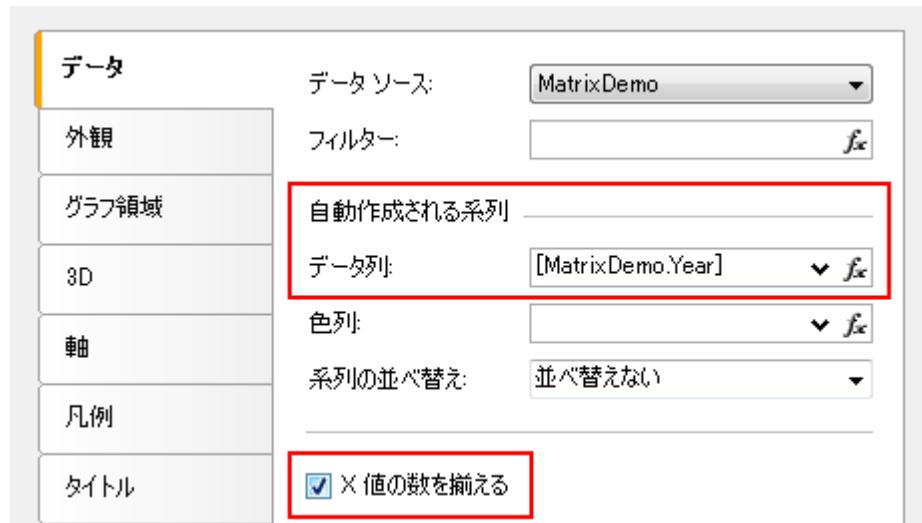
- 1つの系列を作成し、そのプロパティを設定します。この系列は、新しい系列すべてのテンプレートとして使用されます。
- グラフオブジェクトを選択し、自動作成する系列のデータ列を設定します。この列の値は新しい系列の名前になります。その名前の系列がまだない場合は、新しい系列が追加されます。

系列を自動作成する方法を以下に示します。MatrixDemo データテーブルを使用して、従業員別年間売上高のグラフを作成します。1つの系列が1年を表します。これは次の手順で行います。

- グラフをMatrixDemo データソースへ接続します。
- 1 つの系列を作成し、そのデータを次のように設定します。

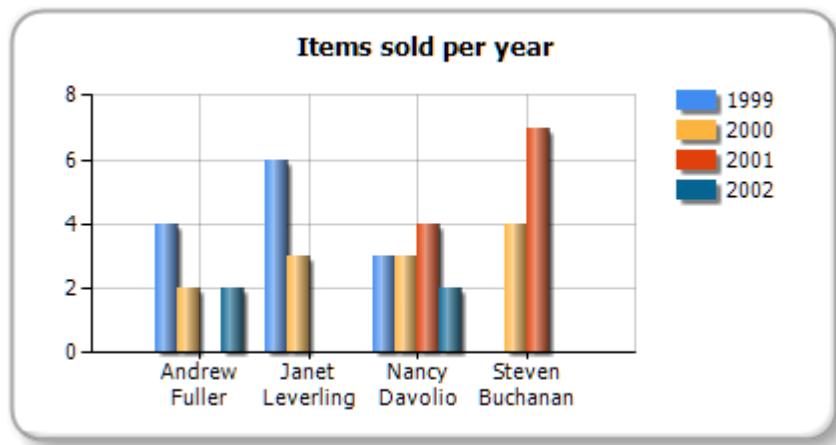


- [データ処理]タブで、[グレーブ化]に"X 値"を選択します。データソースには同じ従業員(同じ名前)のデータが複数存在するため、このオプションは必要です。
- 系列項目の一覧で"グラフ"を選択して[データ]タブを表示し、自動作成する列を設定します。



- (何人かの従業員は売上実績がない年もあるため)系列がかつて値の数が異なる場合があります。系列の値を揃えるには、[X 値の数を揃える]チェックボックスをオンにします。

出力結果は次のようにになります。



## 対話型グラフ

ほかのレポートオブジェクトと同様に、グラフオブジェクトも対話型にすることができます。グラフも、そのグラフ内の値をクリックしたときに別の(詳細)レポートが実行および表示されるよう設定できます。これを行うには、Hyperlink プロパティを設定する必要があります(対話型レポートで説明しています)。グラフ内の要素をクリックしたとき、そのグラフ自身が値をハイパーリンクへ渡します。

FastReport のサンプルレポート "Interactive Chart.frx" を参考にしてみましょう。

2 つのページがあるレポートを作成します。最初のページにはグラフ、2 番目のページにはグラフの値をクリックしたときに表示される詳細レポートが含まれます。

最初のページにはグラフオブジェクトを置き、グラフエディターで次のようにプロパティを設定します。

- 系列項目の一覧から "グラフ" を選択し、[データ]タブの [データソース] で "MatrixDemo" を選択します。
- 系列項目の一覧から対象の系列を選択し、[データ]タブで [X 値] には "MatrixDemo.Name"、[Y 値] には "MatrixDemo.ItemsSold" を指定します。



- [データ処理]タブを表示して、[クリープ化]に "X 値" を指定します。



2番目のレポートページでリストタイプのレポートを作成します。

- 「データ」ウインドウで "SelectedEmployee" という新しいレポートパラメーターを作成します。
- 次のようなレポートレイアウトを作成します。



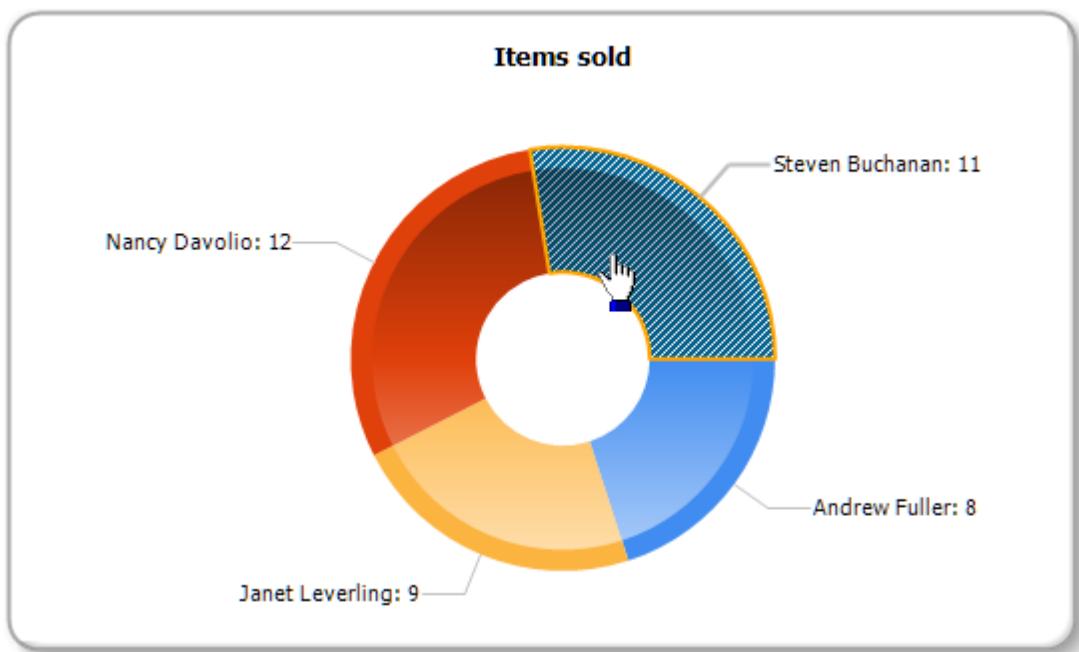
- データバンドエディターを開き、[フィルター]タブで次のフィルター条件を指定します。

```
[MatrixDemo.Name] == [SelectedEmployee]
```

これでグラフオブジェクトのハイパーリンクが設定できました。

- グラフオブジェクトのコンテキストメニューから「ハイパーリンク」を選択します。
- ハイパー リンクタイプとして「レポートページ」タブを選択します。
- 「レポートページ」に2番目のページ("Page2")を選択し、[レポートパラメーター]に"SelectedEmployee"を指定します。

レポートのデザインが完了しました。レポートを実行し、マウスカーソルをグラフの任意の値上に移動させます。値が選択された状態になり、マウスカーソルの形状が変わります。



その値をクリックすると、次のような詳細レポートが表示されます。

### Steven Buchanan orders

Name	Year	Month	Items Sold
Steven Buchanan	2001	1	3
Steven Buchanan	2001	2	4
Steven Buchanan	2000	1	4
<b>Total:</b>			<b>11</b>



# 第3章

データ

# データ

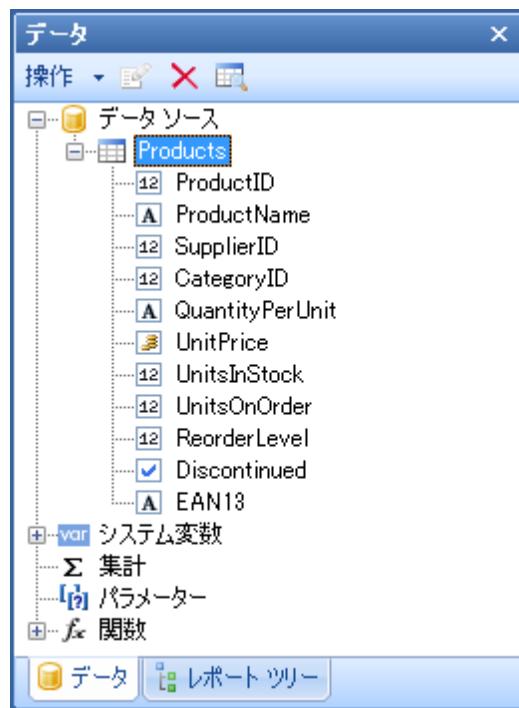
どのレポートもデータを出力します。FastReport では以下のデータについて操作することができます。

- データソース
- システム変数
- 集計値
- レポートパラメーター
- 式 (上記のデータを含む)

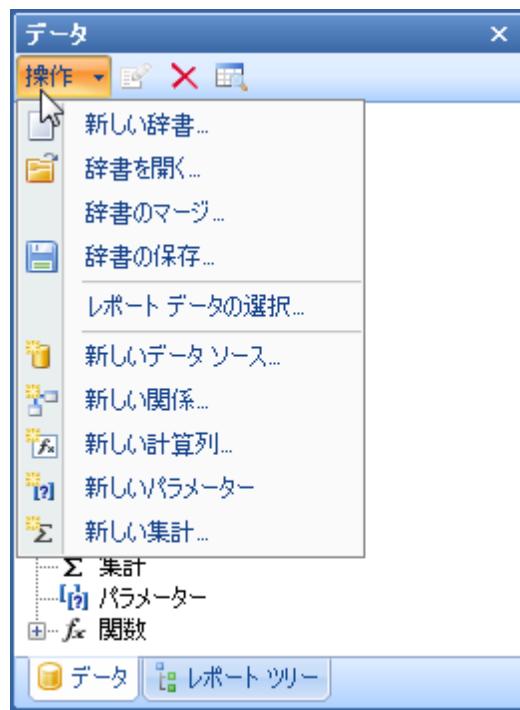
この章では、これらのデータを使った作業方法について説明します。

## [データ] ウィンドウ

[データ] ウィンドウからすべてのデータへアクセスすることができます。このウィンドウは [データ] メニューの [データ ウィンドウの表示] を選択して表示することができます。



[データ] ウィンドウでは、すべてのデータ要素を操作できます。また、その要素をレポートページへドラッグすることもできます。すべての操作は [データ] ウィンドウのツールバーにある [操作] メニューを使って行うことができます。



これらの操作の一部は「データ」ウィンドウのコンテキストメニューと重複しています。たとえば、データソースを選択した場合、そのコンテキストメニューから計算列の作成、データソースの削除、またはデータの表示を行うことができます。

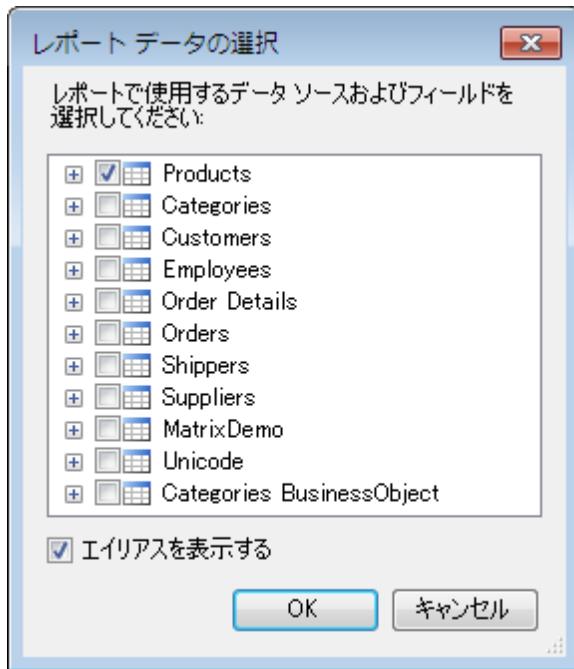
## データ ソース

通常、データソースとは、データベースステータクスまたはSQL クエリを指します。レポートには複数のデータソースを含めることができます。レポートの多くは1つのデータソースを用います。マスター/詳細のようなレポートの場合は、関係を使用して互いに接続されている2つのデータソースが必要です（「関係」については、この章の別のセクションで説明します）。

データソースには1つまたは複数のデータ列があります。各列にはデータ型が定義されています。列のデータ型を調べるには、「データ」ウィンドウでその列を選択して「プロパティ」ウィンドウを見ます。列のデータ型は DataType プロパティで示されます。「データ」ウィンドウで列名の左側に表示されるアイコンもその列のデータ型を特定するのに役立ちます。

レポート用にデータソースを定義する方法は2つあります。

第1の方法 - データソースをアプリケーションで定義し、レポートに登録します。これは、そのアプリケーションを作成したプログラマーが担当します（詳細については、『FastReport .NET Programmer's Manual』を参照してください）。ユーザーは、レポートで使用するデータソースを選択するだけです。これは「データ」メニューの「レポートデータの選択」を選択して開く「レポートデータの選択」ダイアログで行うことができます。



このダイアログには、レポートに登録されているすべてのデータが表示されます。レポートに必要なデータのチェックをオンにします。レポートの作業中には、いつでもこの選択が行えます。

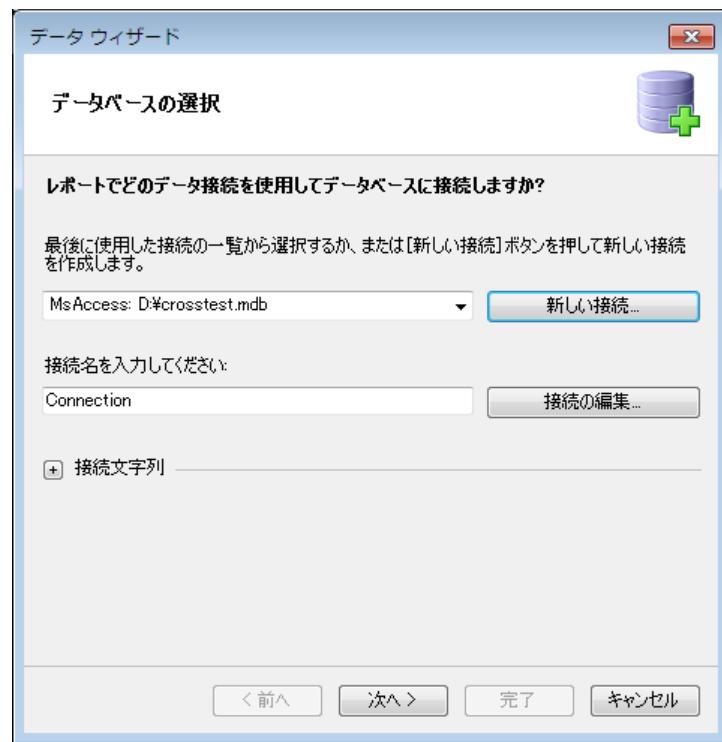
第 2 の方法 - 自分で新しいデータソースを作成します。データベーステーブルまたは SQL クエリが作成可能です。このような場合、データソースの定義はレポートファイルに保存されます。

FastReport では、MS SQL、Oracle、Interbase、Access など多くの一般的な DBMS (データベース管理システム) へ接続できます。さらに、xml/xsd 形式で保存されるデータファイルも使用できます。

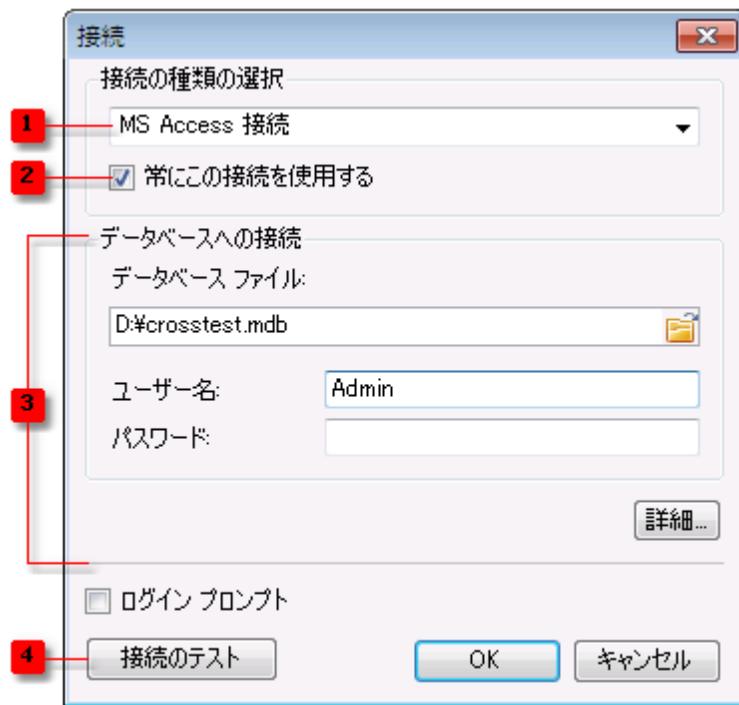
データベーステーブルの接続はレポートファイルには保存されません。その代わり、接続文字列とデータソーススキーマが格納されます。接続文字列にはログインやパスワードのようなデータを含めることができます。それがレポートファイルは暗号化されます。必要であれば、データ暗号化のために独自のキーを用いて安全性を高めることができます。この場合、レポートファイルはご自分のプログラムでのみ正しく開くことができます。

## データソースの作成

新しいデータソースを作成するには、[データ]メニューの [データソースの追加] を選択するか、[データ] ウィンドウのツールバーにある [操作] ボタンをクリックして [新しいデータソース] を選択します。[データ ウィザード] ウィンドウが表示されます。



最初に接続を作成します。そのためには [新しい接続] ボタンをクリックします。接続を設定するウィンドウが表示されます。



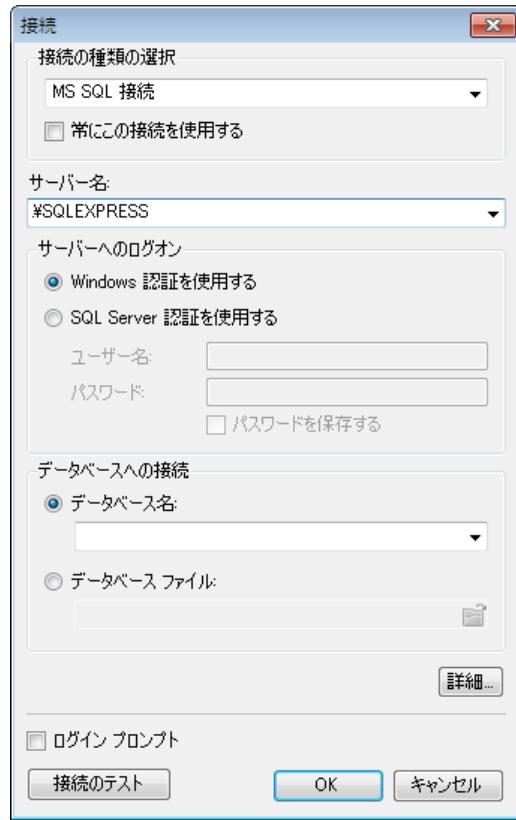
以下の要素をこの図で示します。

1. 接続の種類
2. 有効 (オン)にした場合、選択した接続の種類を既定値として使用します。

### 3. 接続の設定

### 4. 接続のテストボタン

MS Access データベースとの接続は上図のような設定になっています。別の接続を選択した場合は、接続の設定 (3) の領域が変わります。たとえば、MS SQL データベースへの接続には次のようないずれかの設定があります。

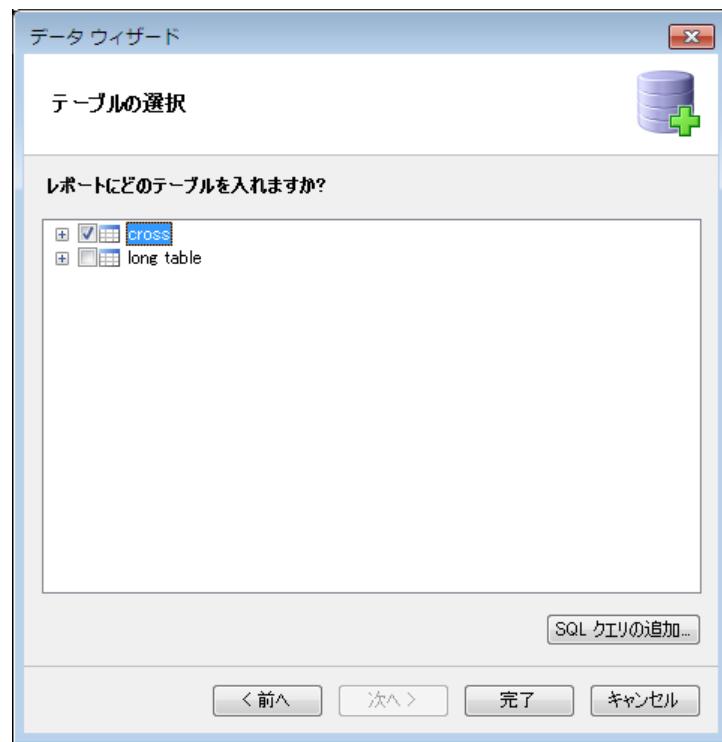


必要な接続の種類を選択し、そのパラメーターを設定します。[OK] ボタンをクリックすると、[接続] ウィンドウを閉じて [データウィザード] ウィンドウに戻ります。

次に、接続名を設定する必要があります。この名前は [データ] ウィンドウに表示されます。



次へ] ボタンをクリックして次の作業へ進みます。ここでは、データベースの中でアクセスするテーブルを選択します。



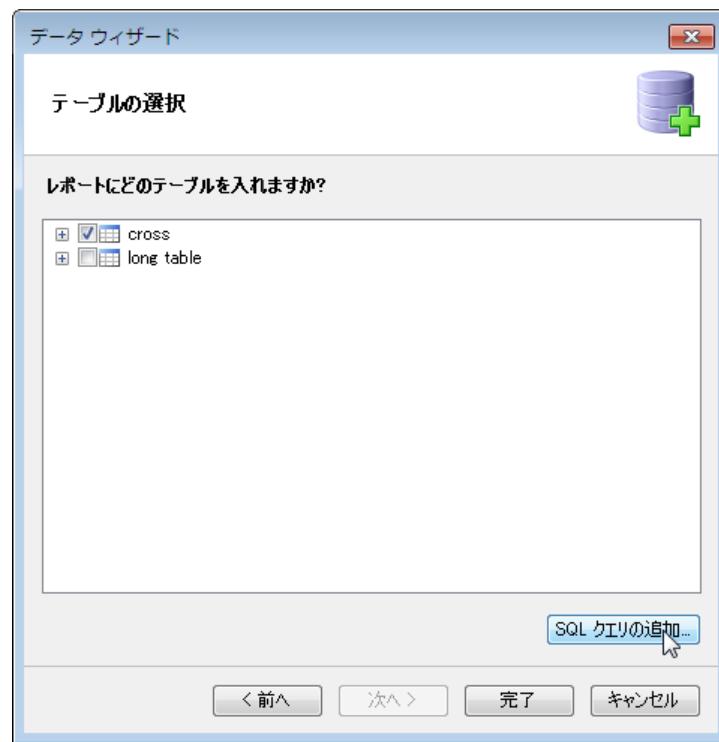
必要なテーブルのチェックをオンにし、[完了] ボタンを押してウィザードを閉じます。これで、作成した接続が [データ] ウィンドウに表示されます。これには選択したデータソースが含まれます。



注意: Fast Reports for .NET はあらゆるプラットフォーム/CPU に対応する "Any CPU" モードでコンパイルされます。このため、デザイン時にデータソースを作成する場合、作成するレポートがその配置環境で実行されるときに使用されるビット数で、そのデータソースが利用できることを確認してください。

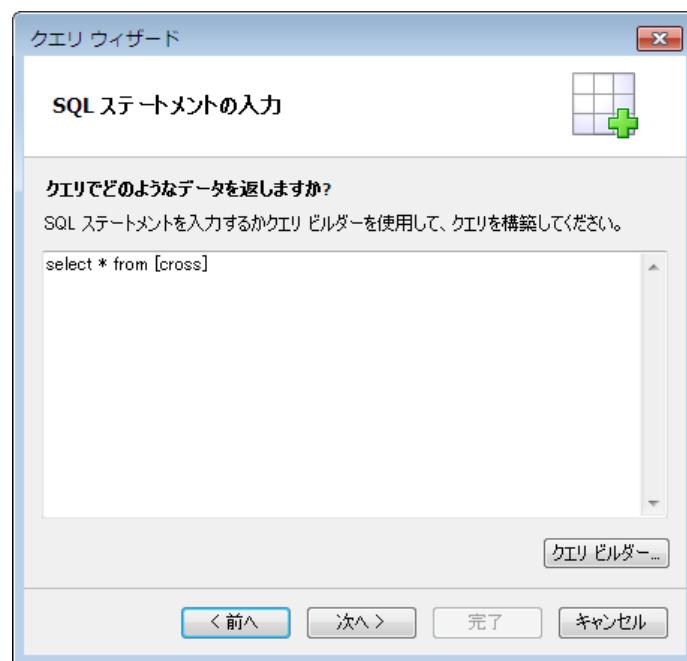
## SQL クエリの作成

データウィザードでは、データベースに含まれるテーブルをすばやく選択することができます。SQL を作成するには、追加作業が必要です。このため、ウィザードの 2 番目のステップで [SQL クエリの追加] ボタンをクリックします。



[クエリ ウィザード] ウィンドウが表示されます。クエリ ウィザードには 4 つのページ (ステップ) があります。[次へ] および [前へ] ボタンを使用してページを切り替えます。

最初のステップで、クエリの名前を設定します。この名前は [データ] ウィンドウに表示されます。一意の名前を入力して、[次へ] ボタンをクリックします。

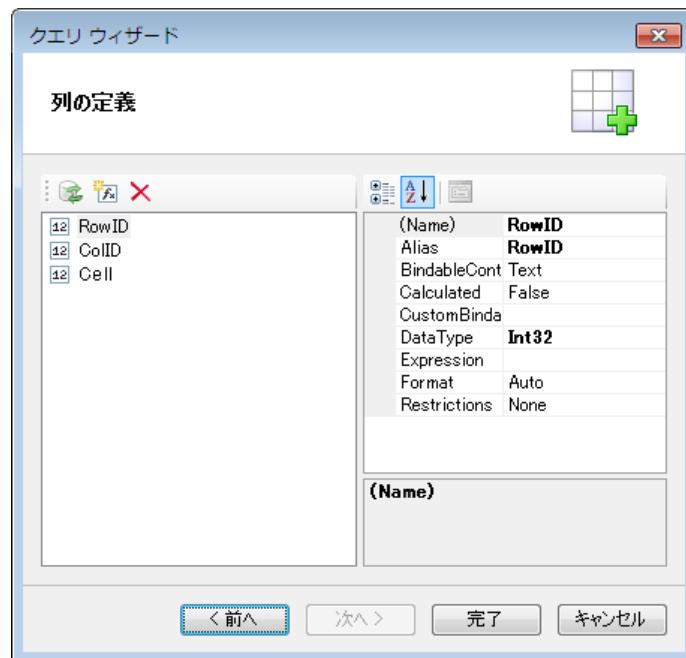


2 番目のステップでは、SQL 言語でクエリを入力します。ご利用の MSDB でサポートされる言語を使用します。クエリを視覚的に作成する

場合は、クエリビレーダーを使用することができます。これを行うには、[クエリビレーダー]ボタンをクリックします。クエリビレーダーについては後で詳しく説明します。

クエリテキストを入力したら、[次へ]ボタンをクリックします。3番目のステップでは、クエリパラメーターを定義することができます。クエリにパラメーターがある場合はこの定義が必要です。パラメーターについては後で詳しく説明します。

最後のステップでは、クエリによって返される列を設定できます。



クエリテキストまたはパラメーター定義で間違いがあった場合、ウィザードの最後のページに進むときにエラー メッセージが表示されます。

一般に、クエリによって該当する列すべてが返されることが保証されます。このステップで、以下のことができます。

- [削除] ボタンを使用して不要な列を削除する。
- [更新] ボタンを使用して列をリセットする。
- [計算列の追加] ボタンをクリックして計算列を追加する。新しい例には、Name、DataType および Expression プロパティを設定する必要があります。

[完了] ボタンをクリックしてウィザードを閉じると、[データウィザード] ウィンドウに戻ります。

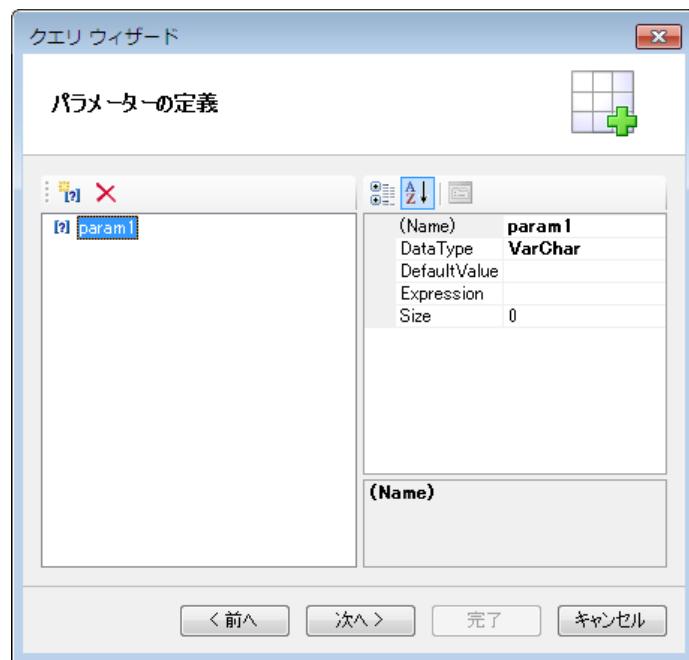
## クエリパラメーター

クエリテキストにパラメーターを含めることができます。次のクエリを見てください。

```
select * from DVDs  
where Title = @param1
```

これは MS SQL デモデータベースへのクエリです。クエリには "param1" という名前のパラメーターが定義されています。クエリにおけるパラメーターの記述方式は DBMS によって異なるので注意してください。MS SQL の場合、パラメーターは "@" 記号で指定されます。MS Access の場合、パラメーターは名前を持たず、"?" 記号で指定されます。

SQL クエリにパラメーターが含まれる場合、それらを宣言する必要があります。この宣言は、前のセクションで説明したクエリウィザードの 3 番目のステップで行うことができます。パラメーターを作成するには、[パラメーターの追加] ボタンをクリックします。新しいパラメーターが作成されます。



以下のパラメーターのプロパティを [プロパティ] ウィンドウで設定します。

プロパティ	説明
名前	パラメーターの名前。これはクエリテキストで使用する同じ名前を指定する必要があります。いくつかの DBMS (MS Access など) では名前付きパラメーターがサポートされません。この場合は、このプロパティを変更しないでください。
DataType	パラメーターのデータ型。
DefaultValue	Expression プロパティが指定されていない場合、または計算できない値の場合 (レポートのデザインモードでクエリを操作する場合など) に使用する値。

Expression	パラメーターの値を返す式。この式はレポートの実行時に処理されます。このプロパティにはあらゆる式を指定することができます(詳細については「式」の章を参照してください)。
Size	パラメーターのデータサイズ。このプロパティは、パラメーターのデータ型が文字列の場合に指定する必要があります。

パラメーターのプロパティを正しく設定しないと、ウィザードの最後のページへ進むときにエラーが発生します。

## パラメーターへ値を渡す

パラメーターは多くの場合、ユーザーから値を求めるために使用されます。値をケリパラメーターに渡す2つの方法について説明します。

第1の方法 - 値をプログラムで渡します。値をケリパラメーターへ直接渡すための簡単な方法がないため、コードから簡単に設定できるレポートパラメーターを使用する必要があります。以下のことを行います。

- レポートパラメーターを作成します(レポートパラメーターについては後で詳しく説明します)。レポートパラメーターの DataType には、ケリパラメーターで使用される同じ DataType を設定します。
- ケリパラメーターの Expression プロパティで、たとえば次のようにレポートパラメーターを参照する。

[MyReportParameter]

- レポートパラメーターへ値を渡す。

```
report1.SetParameterValue("MyReportParameter", 10);
```

第2の方法 - ダイアログフォームを使用して値を渡します(ダイアログフォームについては後で詳しく説明します)。たとえば、文字列型の値を求める場合は次のようにします。

- ダイアログをレポートに追加する。
- そのダイアログに TextBoxControl を置く。このコントロールは文字列値を入力するために使用します。
- パラメーターを以下のように設定します。

```
Name=param1
DataType=VarChar
DefaultValue= (空文字列)
Expression=TextBox1.Text
Size=255
```

TextBox1 は、ユーザーが入力した値を含むコントロールです。

## 接続の編集

データ ウィザードを使用して作成されたデータ接続は編集することができます。これを行うには、[データ] ウィンドウでデータ接続項目を選択し、ツールバーにある [編集] ボタンをクリックします。接続の作成時に使用した [データ ウィザード] ウィンドウが表示されます。ここで、[接続の編集] ボタンをクリックして接続の設定を変更することができます。接続の種類は変更できません。

ウィザードの次のステップでは、レポートに含めるテーブルを選択できます。作業が終了したら、[完了] ボタンをクリックします。

## データ ソースの編集

データ ウィザードを使用して作成されたデータ ソースは編集することができます。これを行うには、[データ] ウィンドウでデータ ソース項目を選択し、ツールバーにある [編集] ボタンをクリックします。データ ソースの追加時にも見られる [ケリ ウィザード] ウィンドウが表示されます。このウィンドウで、SQL ケリ テキストを変更したり、ケリ パラメータやデータ列を設定することができます。

データ ソースを削除するには、[データ] ウィンドウでそのデータ ソース項目を選択し、ツールバーにある [削除] ボタンをクリックします。これはデータ ソースを物理的に削除するものではなく、アクセスできないよう変更するだけです。[データ] メニューの [レポートデータの選択] を選択して、そのデータ ソースを再び有効にすることができます。しかし、これが問題になることはありません。なぜなら、削除されたデータ ソースはレポート ファイルに保存されていないため、次回レポートが読み込まれるときには復元されないからです。

## エイリアス

各データ要素 (データ ソースと列) はそれぞれ独自の名前を持っています。既定では、これはデータベースで定義された名前です。時にはその名前 (たとえば ProdID) の意味を理解することが難しいこともあります。

データ要素はエイリアスという第 2 の名前を持っています。エイリアスを使用すれば、要素の名前を変更することができます。たとえば PROD\_ID という名前を持つ CATEGORY\_TABLE データ ソースがある場合、次のようなエイリアスを指定することができます。

CATEGORY\_TABLE --> Categories

PROD\_ID --> Product ID

このデータ列は以下の方法で参照することができます。

[Categories.Product ID]

データ要素を参照する場合、エイリアスが定義されていない場合はそのエイリアスを使用する必要があります。この場合は要素を元の名前で参照しないでください。

この場合にデータ要素の名前を変更するには、[データ] ウィンドウでその要素を選択し、F2 キーを押します。また、オブジェクトのコンテキストメニューで [名前の変更] を選択することもできます。その操作後、適切な名前を入力して Enter キーを押します。

[プロパティ] ウィンドウを使用して要素の名前を変更することもできます。[データ] ウィンドウで要素を選択しておいて、[プロパティ] ウィンドウでその要素の Alias プロパティの値を変更します。

エイリアスを削除する(元の名前にリセットする)には、その要素を選択してコンテキストメニューから [エイリアスの削除] を選択します。

## 階層データソース

今まで見たデータソースは相関関係、つまりリレーションナルDBMS (RDBMS と呼ばれることが多い)に基づいたものです。FastReport では、階層データソースという異種のデータもサポートします。このようなデータソースは、いわゆるビジネスオブジェクトと呼ばれるものです。これは多くの場合、リレーションナルデータソースを .Net クラスとして表すためにアプリケーションで使用されます。

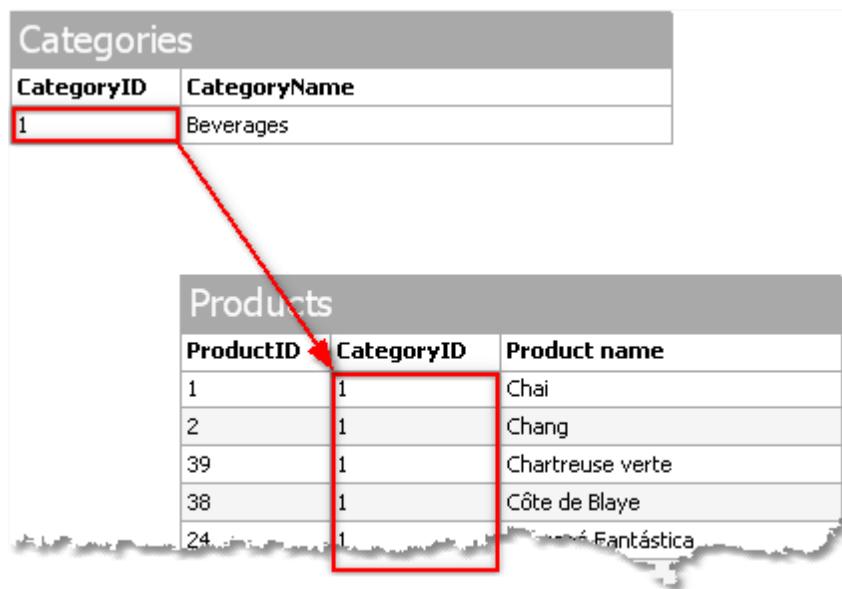
レポートに階層データソースを追加する唯一の方法は、プログラムによる登録です。これについては、『FastReport .Net Programmer's Manual』で説明しています。ここでは、一般的なデータソースと階層データソースの違いについて説明します。下図には Categories BusinessObject と Products という 2 つのデータソースが見られます。ご覧のとおり Products データソースはその親である Categories BusinessObject 内に含まれています。



これは、この 2 つのデータソースが互いに関連付けられており、マスター/詳細レポートタイプで使用できることを意味しています。また、これらのデータソースは单一リストレポートタイプでそれぞれ個別に使用することもできます。

## 関係

2 つのデータソース間の関係を設定することができます。関係は "マスター/詳細" リレーションシップを定義するために使用します。たとえば Categories テーブルの 1 レコードに対し Products テーブルには複数のエントリが存在する可能性があります。



Categories

CategoryID	CategoryName
1	Beverages

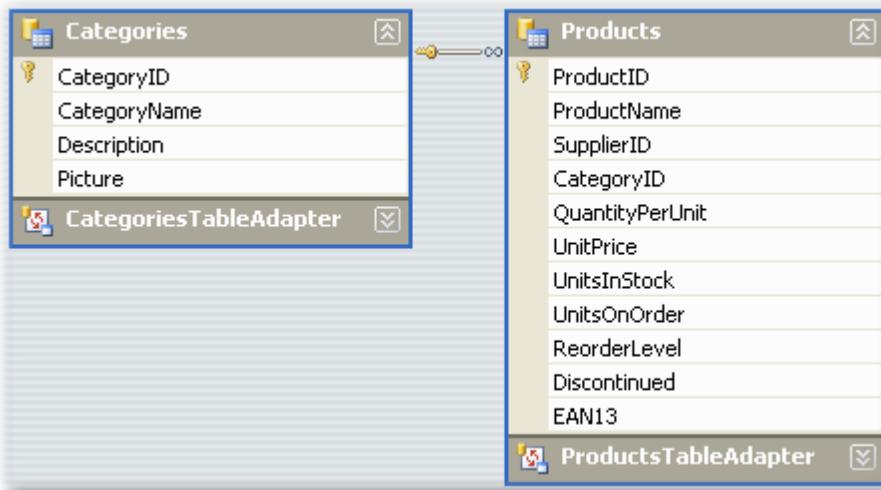
Products

ProductID	CategoryID	Product name
1	1	Chai
2	1	Chang
39	1	Chartreuse verte
38	1	Côte de Blaye
24	1	Fantástica

関係を作成するには、以下の項目を指定する必要があります。

- 親テーブル
- 子テーブル
- 親テーブルのキー列の設定
- 子テーブルのキー列の設定

例として、サンプルデータベースの Categories および Products テーブルを見てみましょう。これらのテーブルは次のよう構造になっています。



どちらのテーブルにもCategoryID フィールドがあり、このフィールドに対して関係を設定できます。このため、1 つのカテゴリに対し複数の製品が含まれる可能性があります。

FastReport で使用されるデータソースはどのように関連付けられるのでしょうか？これには2 つの方法があります。

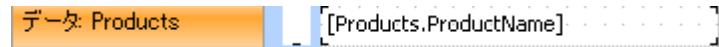
1つ目の方法では、マスター/詳細タイプのレポートを作成することができます。これにはデータバンドを使用します。マスター バンドはマスター データソースに接続され、詳細バンドは詳細データソースに接続されます。レポートデザインは次のようになります。



このレポートを実行すると、次のようにカテゴリ別の製品リストが表示されます。

Category	Product
Beverages	Chai
	Chang
	Chartreuse verte
	Côte de Blaye
	Guaraná Fantástica
	Ipoh Coffee
	Lakkalikööri
	Laughing Lumberjack Lager
	Outback Lager
	Rhönbräu Klosterbier
	Sasquatch Ale
	Steeleeye Stout
Condiments	Aniseed Syrup
	Chef Anton's Cajun Seasoning
	Chef Anton's Gumbo Mix
	Genen Shouyu
	Grandma's Boysenberry Spread
	Gula Malacca

2つ目の方法では、詳細データソースからマスターを参照することができます。例を挙げて説明します。すべての製品を一覧表示するします。そのためには、Products テーブルへ接続されるデータバンドが1つ必要です。



このようなレポートでは全カテゴリの製品がすべて表示されます。各製品に加え、その製品が属するカテゴリ名も表示します。この作業は関係を使用しないと困難です。製品のカテゴリについてわかっていることはその ID (Products テーブルの CategoryID )だけです。出力したカテゴリ名は、Categories テーブルの CategoryName 列に格納されています。次のように関係を使用すればカテゴリ名を参照することができます。

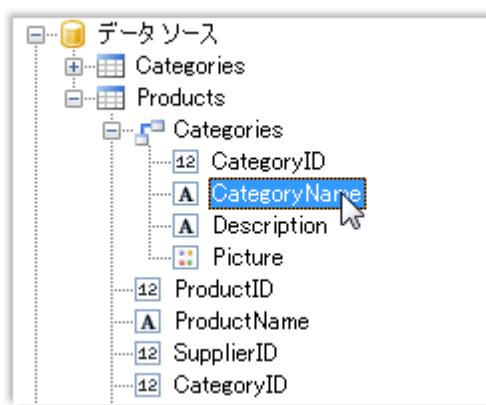
[Products.Categories.CategoryName]

Products テーブルの現在行に対し FastReport は Categories テーブル内でそれは対応する親行を見つけ、そのCategoryName 列の値を返します。

一般的に、親テーブルのフィールドを参照する場合、次のように参照先のテーブルの先祖レベルは無制限に指定できます。

[Child\_table.Its\_parent.Parent\_of\_a\_parent.And\_so\_on.Column\_name]

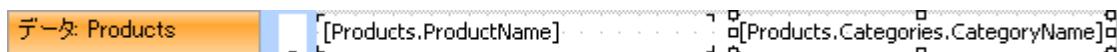
このようなデータ列をレポートに追加するには、[データ] ウィンドウで Products テーブルを開きます。その列の中で、Categories テーブルへのリンクがあります。



上図で示される列をレポートへドラッグすると、次のようなテキストを持つテキストオブジェクトが作成されます。

[Products.Categories.CategoryName]

レポートデザインは次のようにになります。

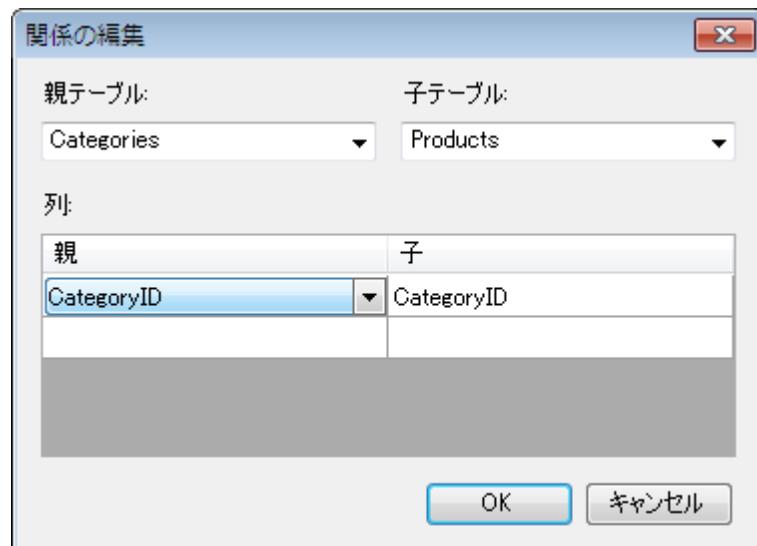


レポートを実行すると、次のように表示されます。

Alice Mutton	Meat/Poultry
Aniseed Syrup	Condiments
Boston Crab Meat	Seafood
Camembert Pierrot	Dairy Products
Carnarvon Tigers	Seafood
Chai	Beverages
Chang	Beverages

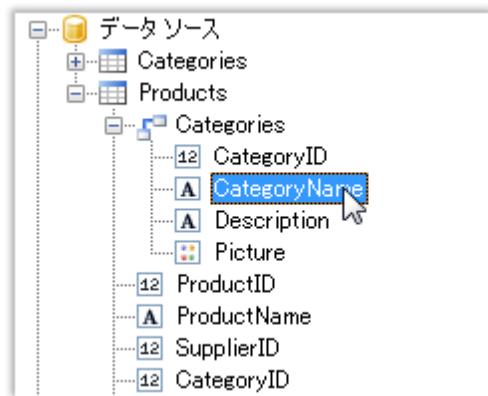
## 関係の作成

関係を作成するには、[データ]ウインドウのツールバーにある[操作]ボタンをクリックし、メニューから[新し関係]を選択します。関係エディター([関係の編集]ダイアログ)が表示されます。



まず始めに、親テーブルと子テーブルを選択する必要があります。次に、[親]フィールドと[子]フィールドで、関連付けるデータ列を選択します。1つまたは複数のデータ列によってテーブルを関連付けることができます。列を設定したら[OK]ボタンをクリックして関係エディターを閉じます。

[データ]ウインドウ内で、子データソースを選択して列の一覧を展開すると、作成した関係が表示されます。展開された列の中で、親ソースデータとの関係性が表示されます。



親ソースデータの列は、ドラッグアンドドロップでレポート上に挿入することができます。図で示されている列を選択し、それをレポートページ上へドラッグした場合は、次のテキストを含むテキストオブジェクトが作成されます。

[Products.Categories.CategoryName]

## 関係の編集

関係を編集するには、[データ] ウィンドウで子データソースの列の一覧から編集対象の関係を選択し、ツールバーにある [編集] ボタンをクリックします。関係の作成時に使用した関係エディターが開きます。

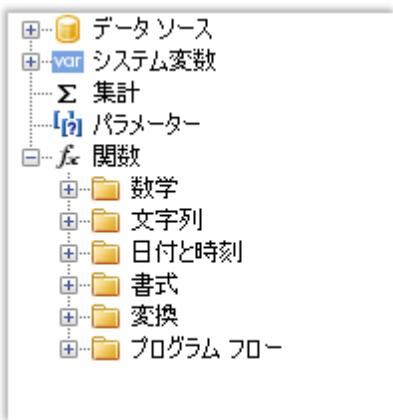
## システム変数

レポートで使用できるシステム変数は次のとおりです。

変数	説明
Date	レポートを開始した日時。
Page	現在のページ番号。
TotalPages	レポートの総ページ数。この変数を使用するには、レポートのダブルバスを有効にしておく必要があります。これは [レポートオプション] ダイアログ ([レポート] メニューの [オプション] を選択) で設定できます。
PageN	"ページN" 形式のページ番号。
PageNofM	"Page N of M" 形式のページ番号。
Row#	グループ内のデータ行番号。この値は、新しいグループの開始時にリセットされます。
AbsRow#	データ行の絶対番号。この値は、新しいグループの開始時にもリセットされません。
Page#	現在のページ番号。複数の実行レポートを1つのパッケージに結合する場合、この変数はパッケージ内における現在のページ番号を返します。 この変数は、実際にはマクロです。コンポーネントがプレビュー ウィンドウに表示されるときに、値が代入されます。つまり、これを式で使用することはできません。
TotalPages#	レポートの総ページ数。複数の実行レポートを1つのパッケージに結合する場合、この変数はパッケージに含まれるページ数を返します。正しい値を取得するためにダブルバスを使用する必要はありません。 この変数は、実際にはマクロです。コンポーネントがプレビュー ウィンドウに表示されるときに、値が代入されます。つまり、これを式で使用することはできません。
HierarchyLevel	階層レポートにおける現在の階層レベル ( <a href="#">階層の出力</a> を参照してください)。最上位レベルは1です。
HierarchyRow#	階層レポートの完全な行番号 ("1.2.1" など)。

## 関数

FastReport.Net には (60 個を超える)多くの組み込み関数があります。関数はすべてカテゴリ別に区分けされ、[データ] ウィンドウからアクセスすることができます。



式、スクリプト内 ([スクリプト](#)の章を参照) で関数を使用する またはその値を "テキスト" オブジェクトに出力することができます。たとえば、"テキスト" オブジェクトに次のテキストを入力します。

```
[Sqrt(4)]
```

これは "2" (4 の平方根) として出力されます。

次の式を指定した場合は、4 が返ります。

```
Sqrt(4) + 2
```

レポート内に関数を挿入する方法を見てみましょう

- [データ] ウィンドウからレポートページへ関数をドラッグアンドドロップすることができます。関数呼び出しを含む "テキスト" オブジェクトが作成されます。テキストを編集して、関数呼び出しにパラメーターを追加します。
- 関数をスクリプトコードへドラッグアンドドロップできます。
- 式エディターでは、[データ] ウィンドウと同様に動作するペインが表示されます。そのペインから式テキストへ関数をドラッグアンドドロップできます。

次に、各関数について詳しく説明します。

## 数学

## Abs

関数	パラメーター	戻り値
Abs	sbyte value	sbyte
Abs	short value	short
Abs	int value	int
Abs	long value	long
Abs	float value	float
Abs	double value	double
Abs	decimal value	decimal

指定した値 (value) の絶対値を返します。

例 :

$$\text{Abs}(-2.2) = 2.2$$

## Acos

関数	パラメーター	戻り値
Acos	double d	double

コサインの値 (d) の角度 (ラジアン) を返します。d は -1 ~ 1 の範囲で指定します。

ラジアンを度に変換するには、戻り値に  $180 / \text{Math.PI}$  を掛けます。

例 :

$$\text{Acos}(0) * 180 / \text{Math.PI} = 90$$

## Asin

関数	パラメーター	戻り値
Asin	double d	double

サインの値 (d) の角度 (ラジアン) を返します。d は -1 ~ 1 の範囲で指定します。

ラジアンを度に変換するには、戻り値に  $180 / \text{Math.PI}$  を掛けます。

例：

$\text{Asin}(0) = 0$

## Atan

関数	パラメーター	戻り値
Atan	double d	double

タングエントの値 (d) の角度 (ラジアン) を返します。

ラジアンを度に変換するには、戻り値に  $180 / \text{Math.PI}$  を掛けます。

例：

$\text{Atan}(1) * 180 / \text{Math.PI} = 45$

## Ceiling

関数	パラメーター	戻り値
Ceiling	double d	double
Ceiling	decimal d	decimal

指定した値 (d) 以上で最も小さな整数を返します。

例：

$\text{Ceiling}(1.7) = 2$

## Cos

関数	パラメーター	戻り値
Cos	double d	double

指定した角度 (d) のコサインを返します。角度はラジアンを単位として指定します。

度をラジアンに変換するには Math.PI / 180 を掛けます。

例：

$\text{Cos}(90 * \text{Math.PI} / 180) = 0$

## Exp

関数	パラメーター	戻り値
Exp	double d	double

$e (2.71828)$ を指定した回数 (d) がべき乗して返します。

例：

$\text{Exp}(1) = 2.71828$

## Floor

関数	パラメーター	戻り値
Floor	double d	double
Floor	decimal d	decimal

指定した値 (d) 以下で最も大きい整数を返します。

例：

$\text{Floor}(1.7) = 1$

## Log

関数	パラメーター	戻り値
Log	double d	double

指定した数 (d) の対数を返します。

例：

$\text{Log}(2.71828) = 1$

## Maximum

関数	パラメーター	戻り値
Maximum	int val1, int val2	int
Maximum	long val1, long val2	long
Maximum	float val1, float val2	float
Maximum	double val1, double val2	double
Maximum	decimal val1, decimal val2	decimal

指定した2つの値 (val1 とval2 )のうち大きい方の値を返します。

例：

Maximum(1,2) = 2

## Minimum

関数	パラメーター	戻り値
Minimum	int val1, int val2	int
Minimum	long val1, long val2	long
Minimum	float val1, float val2	float
Minimum	double val1, double val2	double
Minimum	decimal val1, decimal val2	decimal

指定した2つの値 (val1 とval2 )のうち小さい方の値を返します。

例：

Minimum(1,2) = 1

## Round

関数	パラメーター	戻り値
Round	double d	double
Round	decimal d	decimal

指定した値 (d) を四捨五入して最も近い整数にします。

例：

Round(1.47) = 1

関数	パラメーター	戻り値
Round	double d, int digits	double
Round	decimal d, int digits	decimal

指定した値 (d) を digits パラメーターで指定した(小数部)桁数で四捨五入します。

例：

Round(1.478, 2) = 1.48

## Sin

関数	パラメーター	戻り値
Sin	double d	double

指定した角度 (d) のサインを返します。角度はラジアンを単位として指定します。

度をラジアンに変換するには Math.PI / 180 を掛けます。

例：

Sin(90 \* Math.PI / 180) = 1

## Sqrt

関数	パラメーター	戻り値
Sqrt	double d	double

指定した値 (d) の平方根を返します。

例：

$$\text{Sqrt}(4) = 2$$

## Tan

関数	パラメーター	戻り値
Tan	double d	double

指定した角度 (d) のタンジェントを返します。角度はラジアンを単位として指定します。

度をラジアンに変換するには Math.PI / 180 を掛けます。

例：

$$\text{Tan}(45 * \text{Math.PI} / 180) = 1$$

## Truncate

関数	パラメーター	戻り値
Truncate	double d	double
Truncate	decimal d	decimal

指定した値 (d) の整数部を算出します。

例：

$$\text{Truncate}(1.7) = 1$$

## 文字列

メモ:

- これらの関数では渡された文字列値自体は変更されません。その代わりに、新たに変更された文字列を返します。
- 文字列の先頭文字のインデックスは0です。Insertなど文字のインデックスを用いる関数で作業する場合は、この点に留意してください。

### Asc

関数	パラメーター	戻り値
Asc	char c	int

文字に応する文字コードを表す整数値を返します。

例:

Asc('A') = 65

### Chr

関数	パラメーター	戻り値
Chr	int i	char

指定された文字コードに応する文字を返します。

例:

Chr(65) = 'A'

### Insert

関数	パラメーター	戻り値
Insert	string s, int startIndex, string value	string

文字列 (s) 内の 指定したインデックス位置 (startIndex) に部分文字列 (value) を挿入し、新しい文字列を返します。

例：

Insert("ABC", 1, "12") = "A12BC"

## Length

関数	パラメーター	戻り値
Length	string s	int

文字列 (s) の長さを返します。

例：

Length("ABC") = 3

## LowerCase

関数	パラメーター	戻り値
LowerCase	string s	string

文字列 (s) のすべての文字を小文字に変換し、その結果を返します。

例：

LowerCase("ABC") = "abc"

## PadLeft

関数	パラメーター	戻り値
PadLeft	string s, int totalWidth	string

文字列 (s) 内の文字を右寄せにし、指定した文字数 (totalWidth) によるまで、スペースを左側に埋め込みます。

例：

PadLeft("ABC", 5) = " ABC"

関数	パラメーター	戻り値
PadLeft	string s, int totalWidth, char paddingChar	string

文字列 (s) 内の文字を右寄せにし、指定した文字数 (totalWidth) によるまで、指定した文字 (paddingChar) を左側に埋め込みます。

例：

PadLeft("ABC", 5, '0') = "00ABC"

## PadRight

関数	パラメーター	戻り値
PadRight	string s, int totalWidth	string

文字列 (s) 内の文字を左寄せにし、指定した文字数 (totalWidth) によるまで、スペースを右側に埋め込みます。

例：

PadRight("ABC", 5) = "ABC "

関数	パラメーター	戻り値
PadRight	string s, int totalWidth, char paddingChar	string

文字列 (s) 内の文字を左寄せにし、指定した文字数 (totalWidth) によるまで、指定した文字 (paddingChar) を右側に埋め込みます。

例：

PadRight("ABC", 5, '0') = "ABC00"

## Remove

関数	パラメーター	戻り値
Remove	string s,	string

```
int startIndex
```

指定の位置 (startIndex) から終端位置までのすべての文字を、文字列 (s) から削除します。

例：

```
Remove("ABCD", 3) = "ABC"
```

関数	パラメーター	戻り値
Remove	string s, int startIndex, int count	string

指定の位置 (startIndex) から、指定した文字数分 (count) を、文字列 (s) から削除します。

例：

```
Remove("A00BC", 1, 2) = "ABC"
```

## Replace

関数	パラメーター	戻り値
Replace	string s, string oldValue, string newValue	string

文字列 (s) 内の、指定した部分文字列 (oldValue) を別の部分文字列 (newValue) に置きかえ、その結果を返します。

例：

```
Replace("A00", "00", "BC") = "ABC"
```

## Substring

関数	パラメーター	戻り値
Substring	string s, int startIndex	string

文字列 (s) から部分文字列を取得します。部分文字列は startIndex パラメーターに指定した文字位置から開始します。

例：

Substring("ABCDEF", 4) = "EF"

関数	パラメーター	戻り値
Substring	string s, int startIndex, int length	string

文字列 (s) から部分文字列を取得します。部分文字列は startIndex パラメーターに指定した文字位置から開始し、その長さは length パラメーターで指定されます。

例：

Substring("ABCDEF", 1, 3) = "BCD"

## TitleCase

関数	パラメーター	戻り値
TitleCase	string s	string

指定された文字列 (s) をタイトルケースに変換します。

例：

TitleCase("john smith") = "John Smith"

## Trim

関数	パラメーター	戻り値
Trim	string s	string

文字列 (s) の先頭や末尾にあるスペースをすべて削除します。

例：

Trim(" ABC ") = "ABC"

## UpperCase

関数	パラメーター	戻り値
UpperCase	string s	string

文字列 (s) のすべての文字を大文字に変換し、その結果を返します。

例：

UpperCase("abc") = "ABC"

## 日付と時刻

### AddDays

関数	パラメーター	戻り値
AddDays	DateTime date, double value	DateTime

日付 (date) に、指定の日数 (value) を加算し、結果を新しい日付として返します。

例：

AddDays(#7/29/2009#, 1) = #7/30/2009#

### AddHours

関数	パラメーター	戻り値
AddHours	DateTime date, double value	DateTime

日付/時刻 (date) に、指定の時数 (value) を加算し、結果を新しい日付/時刻として返します。

例：

AddHours(#7/29/2009 1:30#, 1) = #7/29/2009 2:30#

## AddMinutes

関数	パラメーター	戻り値
AddMinutes	DateTime date, double value	DateTime

日付/時刻 (date) に、指定の分数 (value) を加算し、結果を新しい日付/時刻として返します。

例：

AddMinutes(#7/29/2009 1:30#, 1) = #7/29/2009 1:31#

## AddMonths

関数	パラメーター	戻り値
AddMonths	DateTime date, int value	DateTime

日付 (date) に、指定の月数 (value) を加算し、結果を新しい日付として返します。

例：

AddMonths(#7/29/2009#, 1) = #8/29/2009#

## AddSeconds

関数	パラメーター	戻り値
AddSeconds	DateTime date, double value	DateTime

日付/時刻 (date) に、指定の秒数 (value) を加算し、結果を新しい日付/時刻として返します。

例：

AddSeconds(#7/29/2009 1:30:01#, 1) = #7/29/2009 1:30:02#

## AddYears

関数	パラメーター	戻り値

AddYears	DateTime date, int value	DateTime
----------	-----------------------------	----------

日付 (date) に、指定の年数 (value) を加算し、結果を新しい日付として返します。

例：

AddYears(#7/29/2009#, 1) = #7/29/2010#

## DateDiff

関数	パラメーター	戻り値
DateDiff	DateTime date1, DateTime date2	TimeSpan

2 つの日付の間隔 (日数、時間数、分数、秒数) を返します。

例：

DateDiff(#1/2/2009#, #1/1/2009#) = 1.00:00:00

## DateSerial

関数	パラメーター	戻り値
DateSerial	int year, int month, int day	DateTime

指定した年、月および日から DateTime 型の値を作成します。

例：

DateSerial(2009, 7, 29) = #7/29/2009#

## Day

関数	パラメーター	戻り値
Day	DateTime date	int

指定された日付の日 (1 ~ 31) を取得します。

例：

Day(#7/29/2009#) = 29

## DayOfWeek

関数	パラメーター	戻り値
DayOfWeek	DateTime date	string

指定された日付の曜日名を取得します。

例：

DayOfWeek(#7/29/2009#) = "wednesday"

## DayOfYear

関数	パラメーター	戻り値
DayOfYear	DateTime date	int

指定された日付で表される年に入ってからの日数(1 ~ 365)を取得します。

例：

DayOfYear(#7/29/2009#) = 210

## DaysInMonth

関数	パラメーター	戻り値
DaysInMonth	int year, int month	int

指定された年月の日数を返します。

例：

DaysInMonth(2009, 7) = 31

## Hour

関数	パラメーター	戻り値
Hour	DateTime date	int

指定された日付/時刻の時 (0 ~ 23) を取得します。

例 :

Hour(#7/29/2009 1:30#) = 1

## Minute

関数	パラメーター	戻り値
Minute	DateTime date	int

指定された日付/時刻の分 (0 ~ 59) を取得します。

例 :

Minute(#7/29/2009 1:30#) = 30

## Month

関数	パラメーター	戻り値
Month	DateTime date	int

指定された日付の月 (1 ~ 12) を取得します。

例 :

Month(#7/29/2009#) = 7

## MonthName

関数	パラメーター	戻り値
MonthName	int month	string

指定された月 (1 ~ 12) の月名を取得します。

例：

MonthName(1) = "January"

## Second

関数	パラメーター	戻り値
Second	DateTime date	int

指定された日付/時刻の秒 (0 ~ 59)を取得します。

例：

Second(#7/29/2009 1:30:05#) = 5

## Year

関数	パラメーター	戻り値
Year	DateTime date	int

指定された日付の年を取得します。

例：

Year(#7/29/2009#) = 2009

## 書式

## Format

関数	パラメーター	戻り値
Format	string format, params object[] args	string

format 文字列に指定された書式項目を、指定された配列 (args) のオブジェクトインスタンスに対応する値で置き換えます。

たとえば、次のようないい関数呼び出しがあるとします。

```
Format("Name = {0}, hours = {1:hh}", myName, DateTime.Now)
```

これには "{0}" および "{1:hh}" という書式項目が含まれます。これらは myName および DateTime.Now の値で置き換えられます。実行結果は次のようになります。

```
Name = Alex, hours = 12
```

各書式項目は次の書式を使用します。

```
{[index[,alignment][:formatString]}
```

- index - ゼロから始まる整数。書式設定するオブジェクトのリスト内の要素を示します。
- alignment - 任意の整数。書式設定された値を含める領域の最小幅を示します。書式設定される値の長さが alignment より小さい場合、その空いた範囲はスペースで埋められます。alignment が負数の場合、書式設定された値は範囲内で左揃えになります。alignment が正数の場合、書式設定された値は範囲内で右揃えになります。
- formatString - 書式指定子の任意の文字列。

次の表では、標準の数値書式指定文字列を示します。

書式指定子	名前	説明
C または c	通貨	数値は金額を表す文字列に変換されます。
		Format("{0:C}", 10) = "\$10.00"
D または d	10 進数	この書式がサポートされるのは整数型のみです。数値は 10 進数 (0 ~ 9) の文字列に変換されます。
		Format("{0:D}", 10) = "10"
E または e	指数 (科学技術計算)	数値は "-d.ddd...E+ddd" または "-d.ddd...e+ddd" という形式の文字列に変換されます。この "d" は 1 衔の数字 (0 ~ 9) を示します。
		Format("{0:E}", 10) = "1,000000E+001"
F または f	固定小数点	数値は "-ddd.ddd..." という形式の文字列に変換されます。この "d" は 1 衔の数字 (0 ~ 9) を示します。
		Format("{0:F}", 10) = "10.00"
G または g	全般	数値は最も簡潔な形式に変換されます。

		Format("{0:G}", 10) = "10"
N またはn	数値	数値は"-d,ddd,ddd.ddd..." という形式の文字列に変換されます。この "d" は1桁の数字(0 ~ 9)を示します。  Format("{0:N}", 1234.56) = "1,234.56"
P またはp	パーセント	数値はパーセントを表す文字列へ変換されます。変換する値はパーセントを表すため100を乗します。  Format("{0:P}", 0.15) = "15.00%"
X またはx	16進数	数値は16進数の文字列に変換されます。書式指定子の大文字と小文字によって9よりも大きい16進数値を示すアルファベット文字が大文字と小文字のどちらで表示されるかが決まります。たとえば 'X' を指定すると"ABCDEF"となり 'x' を指定すると"abcdef"となります。  Format("{0:X}", 26) = "1A"

浮動小数点数値を書式設定する場合は、次のように書式指定文字列の後に小数部の桁数を指定することができます。

Format("{0:C1}", 12.23) = "\$12.2"

標準の数値書式指定子の中に必要な書式設定のタイプがない場合は、カスタム書式指定文字列を使用することができます。

書式指定文字列	説明
0	ゼロプレースホルダー書式が設定される値で、書式指定文字列の'0'に対応する位置に数字がある場合には、この数字が結果の文字列にコピーされます。整数部の左端の'0'の位置と 小数部の右端の'0'の位置によって、常に結果の文字列に示される桁数が決まります。
#	桁プレースホルダ書式が設定される値で、書式指定文字列の'#'に対応する位置に数字がある場合には、この数字が結果の文字列にコピーされます。それ以外の場合は、結果の文字列のこの位置には何も格納されません。
.	小数点書式指定文字列の1番目の'.'文字によって、書式設定後の値での小数点の位置が決定します。
,	桁区切り記号書式指定文字列に","文字が含まれている場合は、出力の整数部分で各数値グループの間に区切り記号文字が挿入されます。
%	パーセントプレースホルダー書式指定文字列に '%' 文字が含まれていると、書式設定前に数値に100が乗算されます。
;	セカンショ区切り記号';'文字によって、書式指定文字列の正の数値部分、負の数値部分、ゼロの部分が区切られます。

使用例：

```
Format("{0:$#,##0.00}", 1024.25) = "$1,024.25"
Format("{0:00%}", 0.25) = "25%"
Format("{0:$#,##0.00;($#,##0.00);Zero}", 1024.25) = "$1,024.25"
Format("{0:$#,##0.00;($#,##0.00);Zero}", -1024.25) = "($1,024.25)"
Format("{0:$#,##0.00;($#,##0.00);Zero}", 0) = "Zero"
```

---

注意: \記号を文字列に含めるには \ を追加してエスケープさせる必要があります。

VBスクリプトの場合：

```
Format("{0:\#,##0.00}", 1024.25) = "\1,024.25"
```

C#スクリプトの場合：

```
Format("{0:\#\#\#0.00}", 1024.25) = "\1,024.25"
```

---

次の表では、日付時刻 (DateTime) の値を書式設定するための標準書式指定子について説明します。

書式指定子	名前	例
d	短い形式の日付パターン	"8/9/2009"
D	長い形式の日付パターン	"Sunday, August 09, 2009"
f	完全な日付と時刻のパターン (短い形式の時刻)	"8/9/2009 2:44 PM"
F	完全な日付と時刻のパターン (長い形式の時刻)	"8/9/2009 2:44 PM"
g	一般の日付と時刻のパターン (短い形式の時刻)	"8/9/2009 2:44 PM"
G	一般の日付と時刻のパターン (長い形式の時刻)	"8/9/2009 2:44 PM"
t	短い形式の時刻パターン	"2:44 PM"
T	長い形式の時刻パターン	"2:44 PM"

次の表では、日付時刻 (DateTime) の値を書式設定するためのカスタム書式指定子について説明します。

書式指定子	説明
d	月の日にちを1 ~ 31の数値として表します。1桁(1 ~ 9)の日にちの場合、先行ゼロなしで表します。

dd	月の日にちを1～31の数値として表します。1桁(1～9)の日にちの場合、先行ゼロ付き(01～09)で表します。
ddd	曜日の省略名を表します。
ddd	曜日の完全な名前を表します。
f または F	秒の端数の最上位桁を表します。
h	時間を1～12の数値として表します。1桁(1～9)の時間の場合、先行ゼロなしで表します。
hh	時間を1～12の数値として表します。1桁(1～9)の時間の場合、先行ゼロ付き(01～09)で表します。
H	時間を0～23の数値として表します。1桁(1～9)の時間の場合、先行ゼロなしで表します。
HH	時間を0～23の数値として表します。1桁(1～9)の時間の場合、先行ゼロ付き(01～09)で表します。
m	分を0～59の数値として表します。1桁(1～9)の分の場合、先行ゼロなしで表します。
mm	分を0～59の数値として表します。1桁(1～9)の分の場合、先行ゼロ付き(01～09)で表します。
M	月を1～12の数値として表します。1桁(1～9)の月の場合、先行ゼロなしで表します。
MM	月を1～12の数値として表します。1桁(1～9)の月の場合、先行ゼロ付き(01～09)で表します。
MMM	月の省略名を表します。
MMMM	月の完全な名前を表します。
s	秒を0～59の数値として表します。1桁(1～9)の秒の場合、先行ゼロなしで表します。
ss	秒を0～59の数値として表します。1桁(1～9)の秒の場合、先行ゼロ付き(01～09)で表します。
t	A.M./P.M. 指定子の最初の文字を表します。
tt	A.M./P.M. 指定子を表します。
y	年を最大2桁の数値として表します。年の上2桁は省略されます。1桁(1～9)の年の場合、先行ゼロなしで表します。
yy	年を最大2桁の数値として表します。年の上2桁は省略されます。1桁(1～9)の年の場合、先行ゼロ付き(01～09)で表します。
yyy	年を4桁の数値として表します。年が4桁に満たない場合は、4桁になるまで数値が先行ゼロで埋められます。
z	グレゴリオ標準時(GMT)からのシステムのタイムゾーンオフセットを、符号と共に時間単位で表します。このオフセットは、常に先頭の符号と共に表示されます(ゼロは"+0"として表示されます)。正符号(+)はGMTより時間が進んでいることを示し、負符号(-)はGMTより時間が遅れていることを示します。オフセットの値の範囲は-12～+13です。1桁(1～9)のオフセットの場合、先行ゼロなしで表します。先頭には適切な符号が付きます。
zz	グレゴリオ標準時(GMT)からのシステムのタイムゾーンオフセットを、符号と共に時間単位で表します。このオフセット

は、常に先頭または後続の符号と共に表示されます(ゼロは"+00"として表示されます)。正符号(+)はGMTより時間が進んでいることを示し、負符号(-)はGMTより時間が遅れていることを示します。オフセットの値の範囲は-12 ~ +13です。1桁(1 ~ 9)のオフセットの場合、先行ゼロ付き(01 ~ 09)で表します。先頭には適切な符号が付きます。

zzz グローバル標準時(GMT)からのシステムのタイムゾーンオフセットを、符号と共に時間と分の単位で表します。このオフセットは、常に先頭または後続の符号と共に表示されます(ゼロは"+00:00"として表示されます)。正符号(+)はGMTより時間が進んでいることを示し、負符号(-)はGMTより時間が遅れていることを示します。オフセットの値の範囲は-12:00 ~ +13:00です。1桁(1 ~ 9)のオフセットの場合、先行ゼロ付き(01 ~ 09)で表します。先頭には適切な符号が付きます。

: 時刻の区切り記号です。

/ 日付の区切り記号です。

使用例 :

```
Format("{0:d MMM yyyy}", DateTime.Now) = "9 Aug 2009"
Format("{0:MM/dd/yyyy}", DateTime.Now) = "08/09/2009"
Format("{0:MMMM, d}", DateTime.Now) = "August, 9"
Format("{0:HH:mm}", DateTime.Now) = "16:07"
Format("{0:MM/dd/yyyy hh:mm tt}", DateTime.Now) = "08/09/2009 04:07 PM"
```

## FormatCurrency

関数	パラメーター	戻り値
FormatCurrency	object value	string

指定された値(value)を通貨として書式設定します。形式はWindowsの地域設定を使用します。

例 :

```
FormatCurrency(1.25) = "$1.25"
```

関数	パラメーター	戻り値
FormatCurrency	object value, int decimalDigits	string

指定された値(value)を通貨として書式設定します。decimalDigitsパラメーターには、小数点以下に表示する桁数を指定します。

例：

FormatCurrency(1.25, 1) = "\$1.3"

## FormatDateTime

関数	パラメーター	戻り値
FormatDateTime	DateTime value	string

指定された値 (value) を日付/時刻として書式設定します。形式は Windows の地域設定を使用します。この関数の実行結果の文字列に、ニュートラル(中立)の値が含まれることはできません。

例：

FormatDateTime(#1/1/2009#) = "01/01/2009"  
FormatDateTime(#1/1/2009 1:30#) = "01/01/2009 1:30:00 AM"  
FormatDateTime(#1:30#) = "1:30:00 AM"

関数	パラメーター	戻り値
FormatDateTime	DateTime value, string format	string

指定された値 (value) を日付/時刻として書式設定します。形式は format パラメーターで指定される名前付きの書式を使用します。このパラメーターには以下の値を指定できます。

"Long Date"  
"Short Date"  
"Long Time"  
"Short Time"

例：

FormatDateTime(#1/1/2009 1:30#, "Long Date") = "Thursday, January 01, 2009"  
FormatDateTime(#1/1/2009#, "Short Date") = "01/01/2009"  
FormatDateTime(#1:30#, "Short Time") = "01:30 AM"  
FormatDateTime(#1:30#, "Long Time") = "1:30:00 AM"

## FormatNumber

関数	パラメーター	戻り値

FormatNumber	object value	string
--------------	--------------	--------

指定された値 (value) を数値として書式設定します。形式は Windows の地域設定を使用します。

例：

FormatNumber(1234.56) = "1,234.56"

関数	パラメーター	戻り値
FormatNumber	object value, int decimalDigits	string

指定された値 (value) を数値として書式設定します。decimalDigits パラメーターには、小数点以下に表示する桁数を指定します。

例：

FormatNumber(1234.56, 1) = "1,234.6"

## Format Percent

関数	パラメーター	戻り値
FormatPercent	object value	string

指定された値 (value) を割合 (パーセンテージ) として書式設定します。形式は Windows の地域設定を使用します。

例：

FormatPercent(0.15) = "15.00%"

関数	パラメーター	戻り値
FormatPercent	object value, int decimalDigits	string

指定された値 (value) を割合 (パーセンテージ) として書式設定します。decimalDigits パラメーターには、小数点以下に表示する桁数を指定します。

例：

FormatPercent(0.15, 0) = "15%"

## 変換

### ToBoolean

関数	パラメーター	戻り値
ToBoolean	object value	bool

指定した値をブール値に変換します。

例：

ToBoolean(1) = true  
ToBoolean(0) = false

### ToByte

関数	パラメーター	戻り値
ToByte	object value	byte

指定された値をバイトに変換します。

例：

ToByte("55") = 55

### ToChar

関数	パラメーター	戻り値
ToChar	object value	char

指定した値を文字に変換します。

例：

ToChar(65) = 'A'

## ToDateTime

関数	パラメーター	戻り値
ToDateTime	object value	DateTime

指定した値を日付/時刻に変換します。

例：

```
ToDateTime("1/1/2009") = #1/1/2009#
```

## ToDecimal

関数	パラメーター	戻り値
ToDecimal	object value	decimal

指定された値を10進数に変換します。

例：

```
ToDecimal(1) = 1  
ToDecimal("1") = 1
```

## ToDouble

関数	パラメーター	戻り値
ToDouble	object value	double

指定した値を倍精度浮動小数点数値に変換します。

例：

```
ToDouble(1) = 1  
ToDouble("1") = 1
```

## ToInt32

関数	パラメーター	戻り値
ToInt32	object value	int

指定した値を整数に変換します。

例：

```
ToInt32(1f) = 1  
ToInt32("1") = 1
```

### ToRoman

関数	パラメーター	戻り値
ToRoman	object value	string

指定した数値をローマ表記に変換します。この値は1 ~ 3998 の範囲で指定する必要があります。

例：

```
ToRoman(9) = "IX"
```

### ToSingle

関数	パラメーター	戻り値
ToSingle	object value	float

指定された値を単精度浮動小数点数値に変換します。

例：

```
ToSingle(1m) = 1  
ToSingle("1") = 1
```

### ToString

関数	パラメーター	戻り値
ToString	object value	string

指定した値 (value) を文字列に変換します。

例：

```
ToString(false) = "False"  
ToString(DateTime.Now) = "08/09/2009 4:45:00 PM"
```

## ToWords

関数	パラメーター	戻り値
ToWords	object value	string

指定した通貨の値を言葉に変換します。

例：

```
ToWords(1024.25) = "One thousand and twenty-four dollars and 25 cents"
```

関数	パラメーター	戻り値
ToWords	object value, string currencyName	string

指定した通貨の値を言葉に変換します。currencyName パラメーターには通貨を指定します。このパラメーターには以下の値を指定できます。

"USD"  
"EUR"  
"GBP"

例：

```
ToWords(1024.25, "EUR") = "One thousand and twenty-four euros and 25 cents"
```

関数	パラメーター	戻り値
ToWords	object value, string one, string many	string

指定した整数值を言葉に変換します。one パラメーターには単数形の呼称、many パラメーターには複数形の呼称を指定します。

例：

```
ToWords(124, "page", "pages") = "One hundred and twenty-four pages"
```

ToWords(1, "page", "pages") = "One page"

## ToWordsEnGb

関数	パラメーター	戻り値
ToWordsEnGb	object value	string

指定した通貨の値をイギリス英語の言葉に変換します。この関数とToWords 関数では次のような違いがあります。

- GBP 通貨が既定値として使用されます。
- 10 億および1 兆単位を変換するときに使用される語が異なります。

例：

ToWordsEnGb(121) = "One hundred and twenty-one pounds and 00 pence"

関数	パラメーター	戻り値
ToWordsEnGb	object value, string currencyName	string

指定した通貨の値をイギリス英語の言葉に変換します。currencyName パラメーターには通貨を指定します。このパラメーターには以下の値を指定できます。

"USD"  
"EUR"  
"GBP"

例：

ToWordsEnGb(1024.25, "EUR") = "One thousand and twenty-four euros and 25 cents"

関数	パラメーター	戻り値
ToWordsEnGb	object value, string one, string many	string

指定した整数值をイギリス英語の言葉に変換します。one パラメーターには単数形の呼称、many パラメーターには複数形の呼称を指定します。

例：

ToWordsEnGb(124, "page", "pages") = "One hundred and twenty-four pages"  
ToWordsEnGb(1, "page", "pages") = "One page"

## ToWordsRu

関数	パラメーター	戻り値
ToWordsRu	object value	string

指定した通貨の値をロシア語の言葉に変換します。

例：

ToWordsRu(1024.25) = "25 "

関数	パラメーター	戻り値
ToWordsRu	object value, string currencyName	string

指定した通貨の値をロシア語の言葉に変換します。currencyName パラメーターには通貨を指定します。このパラメーターには以下の値を指定できます。

"RUR"  
"UAH"  
"USD"  
"EUR"

例：

ToWordsRu(1024.25, "EUR") = "25 "

関数	パラメーター	戻り値
ToWordsRu	object value, bool male, string one, string two, string many	string

指定した整数值をロシア語の言葉に変換します。male パラメーターはその名前の性を示します。one、two、およびfive パラメーターにはそれぞれ 1 個、2 個および5 個の場合に使用される形(单数、双数、複数)を指定します。

例：

```
// "        " は女性名詞であり、male = false
ToWordsRu(122, false, "        ", "        ", "        ") =
"        "

// "        " は男性名詞であり、male = true
ToWordsRu(122, true, "        ", "        ", "        ", "        ") =
"        "
```

## プログラム フロー

### Choose

関数	パラメーター	戻り値
Choose	double index, params object[] choice	object

配列 (choice) のうち index パラメーターで指定されたインデックスに対応する要素を返します。配列の先頭要素のインデックスは1 です。

例：

```
Choose(2, "one", "two", "three") = "two"
```

### lif

関数	パラメーター	戻り値
lif	bool expression, object truePart, object falsePart	object

式 (expression) がTrue の場合、truePart の値を返します。False の場合はfalsePart の値を返します。

例：

```
lif(2 > 5, "true", "false") = "false"
```

## Switch

関数	パラメーター	戻り値
Switch	params object[] expressions	object

expressions には、式と値を組み合わせた引数を指定します。Switch 関数は、引数に指定された項目の奇数番目の式を左から順に評価し、True として評価される最初の式に関連付けられる偶数番目の値を返します。

例：

```
// a の値に応じて "a greater than 0"、"a less than 0"、  
// "equals to 0" のいずれかの値を返す  
Switch(  
    a > 0, "a greater than 0",  
    a < 0, "a less than 0",  
    a == 0, "a equals to 0")
```

## 集計

多くのレポートで集計情報 (グループ単位の集計、リスト内の行数など) を表示する必要があります。FastReport は集計を使用してこのタスクを実行します。集計の作成には、以下のパラメーターを指定する必要があります。

- 集計関数の種類
- 計算される式 Count (総数) 関数には式を指定する必要はありません。
- 条件。条件を満たす場合に関数が実行されます。条件の設定は必須ではありません。
- 関数の処理対象となるデータバンド。
- 集計値が出力されるバンド。

集計関数の一覧を以下に示します。

関数	説明
Sum (合計)	式の合計を計算します。
Min (最小値)	式の最小値を計算します。
Max (最大値)	式の最大値を計算します。
Average (平均)	式の平均値を計算します。
Count (総数)	行数を返します。

## 集計の作成

集計関数の使用方法を例を挙げて説明します。Categories およびProducts という2つのテーブルを使用して次のようなマスター/詳細レポートを作成します。

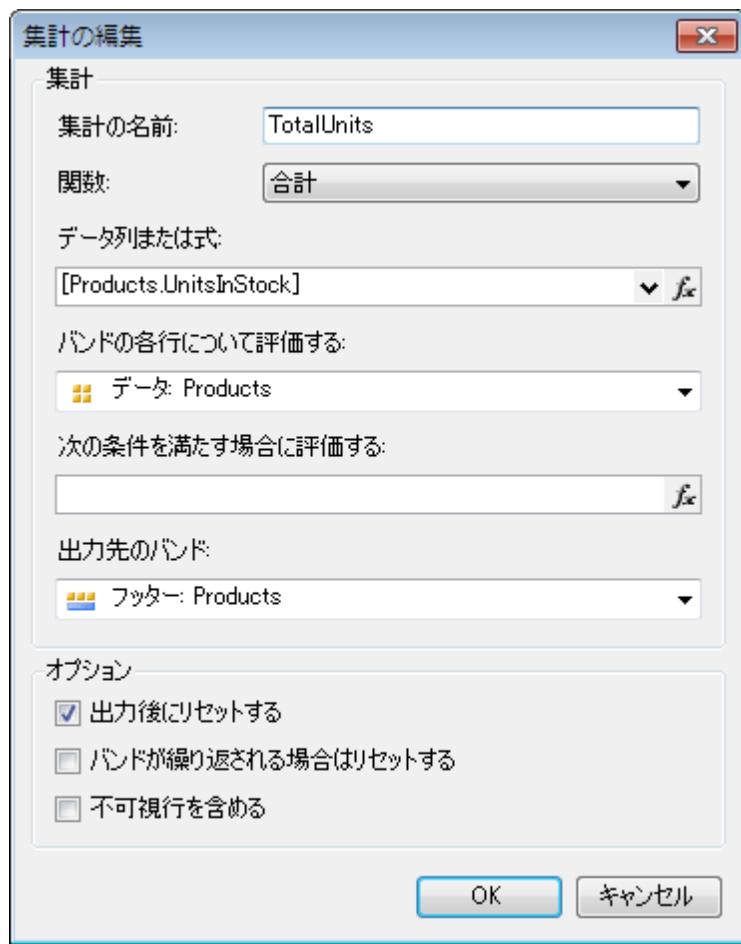


実行されたレポートは次のようになります。

Beverages		Condiments	
Product Name	Units In Stock	Product Name	Units In Stock
Chai	39	Aniseed Syrup	13
Chang	17	Chef Anton's Cajun Seasoning	53
Chartreuse verte	69	Chef Anton's Gumbo Mix	0
Côte de Blaye	17	Genen Shouyu	39
Guaraná Fantástica	20	Grandma's Boysenberry Spread	120
Ipoh Coffee	17	Gula Malacca	27
Lakkalikööri	57	Louisiana Fiery Hot Pepper Sauce	76
Laughing Lumberjack Lager	52	Louisiana Hot Spiced Okra	4
Outback Lager	15	Northwoods Cranberry Sauce	6
Rhönbräu Klosterbier	125	Original Frankfurter grüne Soße	32
Sasquatch Ale	111	Siroop d'étable	113
Steeleye Stout	20	Vegie-spread	24

このレポートに、カテゴリ別の在庫総数 (UnitsInStock データ列の合計) を出力する集計を追加します。集計はデータフッターバンドに出力します。

集計値を出力するには、まずその集計を作成する必要があります。[データ] ウィンドウのツールバーの [操作] ボタンをクリックし、[削除] [集計] を選択します。別の方として、データツリーの "集計" 要素を右クリックし [削除] [集計] を選択することもできます。次のような集計エディター (集計の編集) ウィンドウが表示されます。

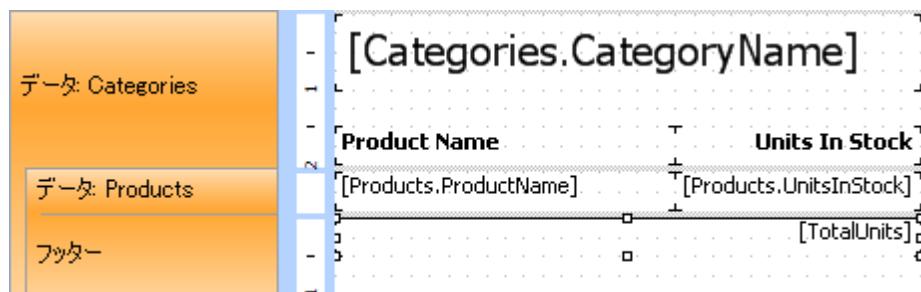


まず、集計の名前を指定します。この名前によって集計が参照されるので、計算される対象がわかりやすい名前を付けておきましょう。ここでは "TotalUnits" という名前を付けます。

集計関数には"合計"を選択します。

次に、集計を計算するデータ範囲を指定します。[「バンドの各行について評価する」]フィールドには、製品の一覧が出力されるデータバンドを選択します。[「出力先のバンド」]フィールドには、集計が出力されるバンド、つまりデータフッターバンドを選択します。

[OK]ボタンをクリックしてエディターを閉じます。[データ]ウインドウには新しい集計項目が表示されます。その集計項目をレポートへドラッグできます。



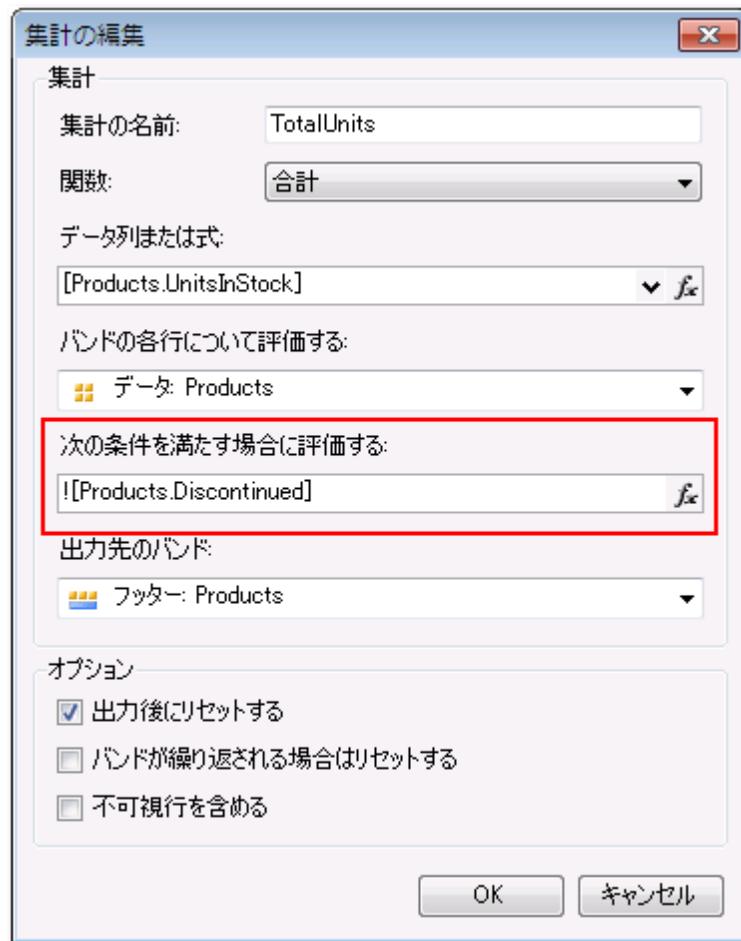
レポートを実行すると、次のように出力されます。

Beverages		Condiments	
Product Name	Units In Stock	Product Name	Units In Stock
Chai	39	Aniseed Syrup	13
Chang	17	Chef Anton's Cajun Seasoning	53
Chartreuse verte	69	Chef Anton's Gumbo Mix	0
Côte de Blaye	17	Genen Shouyu	39
Guaraná Fantástica	20	Grandma's Boysenberry Spread	120
Ipoh Coffee	17	Gula Malacca	27
Lakkalikööri	57	Louisiana Fiery Hot Pepper Sauce	76
Laughing Lumberjack Lager	52	Louisiana Hot Spiced Okra	4
Outback Lager	15	Northwoods Cranberry Sauce	6
Rhönbräu Klosterbier	125	Original Frankfurter grüne Soße	32
Sasquatch Ale	111	Sirop d'éable	113
Steeleye Stout	20	Vegie-spread	24
	559		507

## 条件付き集計

前のセクションで挙げた例では、集計は全データ行を対象に計算されています。集計エディターで条件を指定することで、この対象範囲を制限することができます。集計の計算は、条件が満たす行のみを対象とします。

たとえば、次のような条件を設定できます。



これは、集計を計算する対象が Discontinued フラグがセットされていない 製品のみであることを示しています。

## 累計

サンプルの集計では、データフッターバンドの出力後に集計がリセットされます。これは、集計エディターで集計の出力後にリセットするよう指定しているためです。この結果、カテゴリ単位の集計値が出力されます。

[出力後にリセットする] チェックボックスをオフにすると、集計は出力後もリセットされません。これは「累計」と呼ばれます。

2種類の集計 (一般的な集計と累計) を同時に表示する必要がある場合、同じ設定の集計をもう1つ作成し、その [出力後にリセットする] チェックボックスをオフにします。

## ページ集計

ページフッターバンドに表示される集計を作成するには、[出力先のバンド] フィールドで「ページフッター」を指定します。

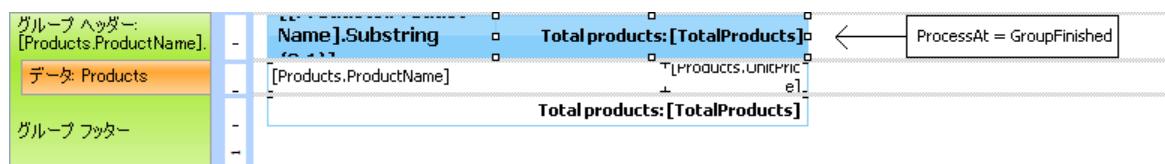
## ヘッダーに集計を出力する

通常、集計値はフッターバンド(データフッター、グレープフッターなど)に出力します。これは、集計を出力するときに、その値が適切に計算されそのまま使用できるので、自然な出力位置と言えます。しかし場合によっては、集計をヘッダー(グレープフッターなど)に出力する必要があるかもしれません。ヘッダーに集計を出力しようとすると、その値はゼロで表示されてしまいます。この時点では、集計はまだ計算されていません。

この問題を解決するため、FastReportには"出力の遅延"機能があります。テキストオブジェクトにはProcessAtというプロパティがあり、以下のいずれかの値を持ちます。

値	説明
Default	既定の出力モード。これが既定値です。
ReportFinished	オブジェクトの値は、レポートの最後に計算されます。
ReportPageFinish	オブジェクトの値は、ページのすべてのバンドが終了したときに計算されます。
ed	
PageFinished	オブジェクトの値は、ページの最後に計算されます。
ColumnFinished	オブジェクトの値は、列の最後に計算されます。
DataFinished	オブジェクトの値は、データバンドの最後(そのデータのフッターが出力されたとき)に計算されます。
GroupFinished	オブジェクトの値は、グレープの最後(そのグレープのフッターが出力されたとき)に計算されます。

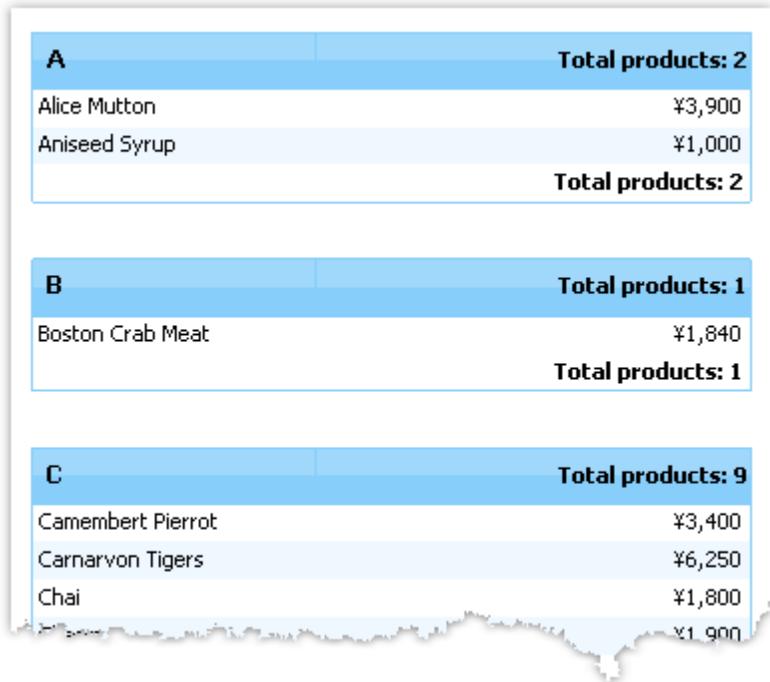
どのように動作するか見てみましょう。集計を出力するテキストオブジェクトをグレープフッターに置きます。テキストオブジェクトのProcessAtプロパティに"GroupFinished"を設定します。



レポートの実行時、FastReportは以下のことを行います。

- グレープヘッダーを出力します。集計値は0(誤り)として出力されますが、FastReportは後でこのオブジェクトを処理することを覚えておきます。
- すべてのデータ行を出力します。
- グレープフッターを出力します。この時点で、FastReportはグレープヘッダーに出力されるオブジェクトを再び処理して正しい集計値を出力します。

実行されたレポートは次のようになります。



ProcessAt プロパティにほかの値を使用すると、レポートタイトルにレポート集計を出力したり (ProcessAt = ReportFinished)、ページヘッダーにページ集計を出力 (ProcessAt = PageFinished) したりすることができます。

レポートファイルのキャッシュをオンにした場合 ([レポート]メニューで [オプション]を選択、[全般]タブの [ファイルキャッシュを使用する]チェックボックス)、この出力の遅延機能は動作しません。

## レポートパラメーター

レポートごとにパラメーターを定義することができます。パラメーターは変数です。その値はレポート内およびレポート外 (レポートの呼び出しでパラメーターの値をレポートに転送するプログラム) のどちらからでも定義することができます。詳細については、『FastReport.NET Programmer's Manual』を参照してください。パラメーターは、式で使用したり、テキストオブジェクトなどのレポートオブジェクトに表示したりすることができます。

最も一般的なパラメーターの使用方法は次のとおりです。

- パラメーターに設定した条件でデータをフィルター処理する
- レポートにパラメーター値を出力する

パラメーターには以下のプロパティがあります。

プロパティ	説明
名前	パラメーターの名前にはドット(.)以外の記号も使用できます。

DataType	パラメーターのデータ型。
Expression	パラメーターの値を返す式。式の詳細については「式」の章を参照してください。この式はパラメーターを呼び出したときに処理されます。
Value	パラメーターの値。このプロパティはデザイナーでは使用できません。これはプログラムで指定できます。

Name および DataType プロパティは設定する必要があります。Expression プロパティは空にしておくことができます。この場合、パラメーターの値はプログラムで渡してください。

## パラメーターの作成

パラメーターを作成するには、[データ] ウィンドウで "パラメーター" 要素を右クリックし、コンテキストメニューから新し [パラメーター] を選択します。



F2 キーを押してパラメーターに名前を付け、[プロパティ] ウィンドウでそのパラメーターの DataType プロパティを設定します。

パラメーターは入れ子にすることができます。入れ子のパラメーターを作成するには、親パラメーターを右クリックし、コンテキストメニューから新し [パラメーター] を選択します。



親パラメーターと入れ子のパラメーターの両方が参照可能です。入れ子のレベルに制限はありません。

## レポート内でのパラメーターの使用

次のように、角かっこを使用した式からパラメーターを参照することができます。

[Parameter name]

入れ子のパラメーターには、次の方法を使用して参照する必要があります。

[Parent parameter.Child parameter]

パラメーターは DataType プロパティによってデータ型が明確に定義されているため、パラメータを使用する場合は、そのデータ型に応じたアクションを実行できます。このため、文字列型のパラメーターは次のように式内で使用することができます。

```
[StringParameter].Substring(0, 2)
```

パラメーターの使用例を挙げてみましょう Employees テーブルを出力するレポートがあるとします。このレポートを編集して、指定した番号 (ID) の従業員に関する情報のみを出力するようします。これを行うには、EmployeeID データ列のデータをフィルター処理する必要があります。"EmployeeID" という名前のパラメーターを作成します。パラメーターのデータ型には EmployeeID データ列と同じ Int32 を指定します。指定した ID の従業員をフィルター処理するには、データバンドのエディター ([データバンドの編集] ダイアログ) の [フィルター] タブで次の式を指定します。

```
[Employees.EmployeeID] == [EmployeeID]
```

パラメーター値をプログラムからレポートへ渡すには、次のようなコードを使用します。

```
report1.SetParameterValue("EmployeeID", 2);
```



# 第4章

式

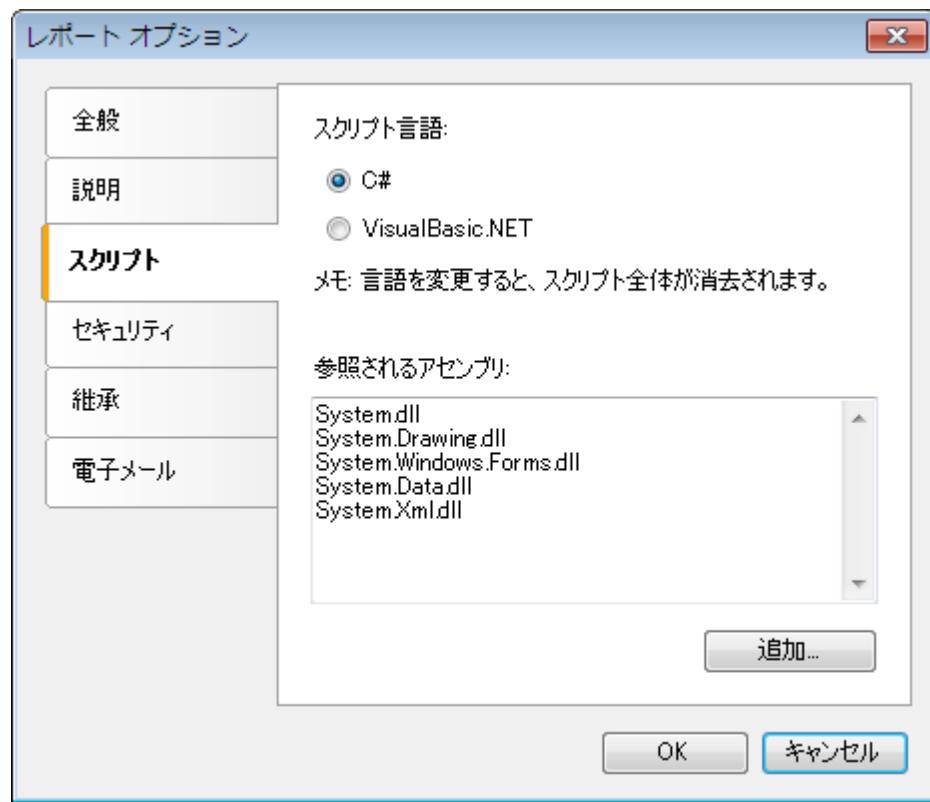
## 式

FastReport では多くの場所で式が使用されます。たとえば テキストオブジェクトには角かっこで囲まれた式を含めることができます。

式はC# またはVB.NET 言語のコードであり 値を返します。たとえば 次のよう指定します。

2 + 2

式はレポートのスクリプトとして選択した言語で記述する必要があります。言語の既定値はC# です。この言語は[レポートオプション]ダイアログ([レポート]メニューの[オプション]を選択)の[スクリプト]タブで変更できます。

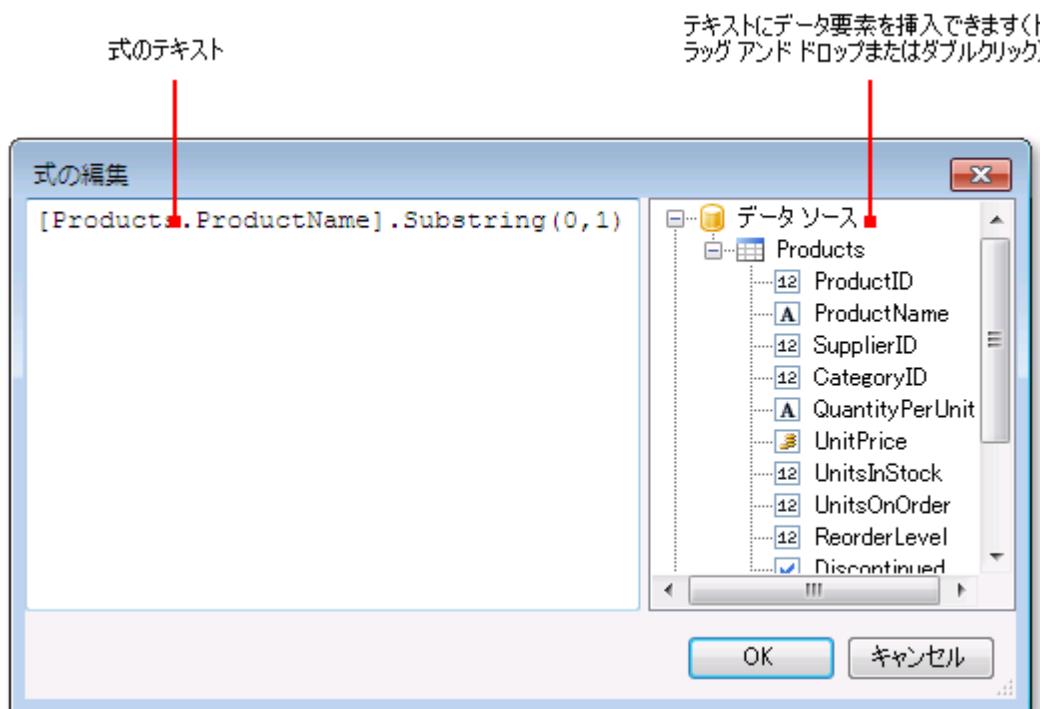


## 式エディター

式をすばやく作成するには、式エディターを使用します。これはFastReport ユーザー インターフェイスで、式を入力できる場所から起動できます。



式エディターはウインドウとして表示され、式を入力したり 式内にデータ要素を挿入することができます。



## レポートオブジェクトの参照

レポートオブジェクトを参照するには そのオブジェクトの名前を使用します。次の例は Text1 オブジェクトの高さを返します。

Text1.Height

レポートのプロパティを参照するには Report 変数を使用します。次の例はレポートを読み込むファイル名を返します。

Report.FileName

さらに、入れ子のオブジェクトのプロパティを参照することもできます。次の例はレポート名を返します。

## .Net 関数の使用

式ではあらゆる .Net オブジェクトが使用できます。次のコードは Max 関数の使用例を示しています。

```
Math.Max(5, 10)
```

既定では、レポートは以下の .Net アセンブリを使用します。

```
System.dll
System.Drawing.dll
System.Windows.Forms.dll
System.Data.dll
System.Xml.dll
```

これらのアセンブリで宣言されるすべての .Net オブジェクトにアクセスできます。別のアセンブリにアクセスする必要がある場合は、そのアセンブリの名前をレポートのアセンブリリストに追加します。これは [レポートオプション] ダイアログ ([レポート] メニューの [オプション] を選択) の [スクリプト] タブで行うことができます。



たとえば、アプリケーション内で宣言した関数をレポートで使用する場合、アプリケーションアセンブリ (exe または .dll) をレポートのアセンブリリストに追加します。これで、アプリケーションの名前空間を使用することでその関数を呼び出すことができます。たとえば、次の関数がアプリケーションで定義されています。

```
namespace Demo
{
    public static class MyFunctions
    {
        public static string Func1()
        {
            return "Hello!";
        }
    }
}
```

```
 }  
 }
```

これをレポートで次のように使用できます。

```
Demo.MyFunctions.Func1()
```

レポートのスクリプトの先頭で "using Demo" 行を追加すると、次のように構文を短縮することができます。

```
MyFunctions.Func1()
```

スクリプトで定義された関数または変数を参照するには、単にその名前を使用します。

```
myPrivateVariableThatIHaveDeclaredInScript
```

```
MyScriptFunction()
```

式で使用できるのは、値を返す関数のみです。

## データ要素の参照

式では、標準の言語要素だけでなく以下のレポート要素が使用できます。

- データソースの列
- システム変数
- 集計値
- レポートパラメーター

これらの要素はすべて [データ] ウィンドウに含まれています。詳細については [データ](#) の章を参照してください。これらの要素は角かっこで囲むことによって、式で使用することができます。たとえば、次のように指定します。

```
[Page] + 1
```

この式は次に出力されるページ番号を返します。ここでは現在のレポートページ番号を返すシステム変数 Page が使用されています。この変数が角かっこで囲まれています。

## データ ソースの参照

データソースの列を参照するには、以下の書式を使用します。

```
[DataSource.Column]
```

データソース名と列名はピリオドで区切ります。たとえば、次のように指定します。

```
[Employees.FirstName]
```

関係を使用してデータソースを参照する場合は、データソース名を含めることができます。詳細については、「データ」の章を参照してください。たとえば、関連付けられたデータソースの列を参照するには次のように指定されます。

```
[Products.Categories.CategoryName]
```

次に、式で列を使用する例を見てみましょう。

```
[Employees.FirstName] + " " + [Employees.LastName]
```

注意事項: 各列にはデータ型が定義されています。これはそれぞれの DataType プロパティで設定されています (このプロパティは「データ」ウインドウで「データ列」を選択したときに「プロパティ」ウインドウで表示されます)。式内で列を使用する場合、その使用法はその列のデータ型によって異なります。たとえば、前述の例で示した FirstName および LastName の列はいずれも文字列データ型であるため、それに応じた使用が可能です。次の例では、数値型の Employees.Age 列を使用しようとしています。これはエラーを引き起します。

```
[Employees.FirstName] + " " + [Employees.Age]
```

文字列と数値を混在させてしまうため、この式はエラーになります。このため、次のように数値を文字列に変換する必要があります。

```
[Employees.FirstName] + " " + [Employees.Age].ToString()
```

この場合、Employees.Age 列を整数型の変数であるかのように参照します。これで適切な式として成立します。式はすべてコンパイルされます。コンパイラの観点から、(データ列の参照のような) 非標準的なものがすべて、コンパイラにわかりやすい別のデータ型に変換されます。このため、最後の式は次のような形に変換されます。

```
(string)(Report.GetColumnValue("Employees.FirstName")) + " " +  
(int)(Report.GetColumnValue("Employees.Age")).ToString()
```

ご覧のとおり、FastReport データ列への参照を次のように変更します。

```
[Employees.FirstName] --> (string)(Report.GetColumnValue("Employees.FirstName"))  
[Employees.Age] --> (int)(Report.GetColumnValue("Employees.Age"))
```

つまり、式でデータ列を使用する場合、その列をデータ型が定義された変数のようご使用することができます。たとえば、次の式では従業員名の先頭文字を返します。

```
[Employees.FirstName].Substring(0, 1)
```

## システム変数の参照

以下のシステム変数を式で使用することができます (この変数は [データ] ウィンドウからアクセス可能です)。

変数	.Net データ型	説明
Date	DateTime	レポートを開始した日時。
Page	int	現在のページ番号。
TotalPages	int	レポートの総ページ数。この変数を使用するには、レポートのダブルパスを有効にしておく必要があります。これは [レポートオプション] ダイアログ ([レポート] メニューの [オプション] を選択) で設定できます。
PageN	string	"ページ N" 形式のページ番号。
PageNofM	string	"Page N of M" 形式のページ番号。
Row#	int	グループ内のデータ行番号。この値は、新しいグループの開始時にリセットされます。
AbsRow#	int	データ行の絶対番号。この値は、新しいグループの開始時にもリセットされません。

各変数にはデータ型が定義されます。このデータ型に応じて、式での使用方法が異なります。次の式の例では、日付が使用されています。

```
[Date].Year
```

この式は現在の年を返します。Date 变数は DateTime 型であるため、その Year プロパティを参照できます。同様の式で現在の月も取得できます ([Date].Month)。

FastReport は Date 变数などのシステム変数への参照を次のように変換します。

```
((DateTime)Report.GetVariableValue("Date"))
```

## 集計値の参照

集計値を参照するには、次のようにその名前を使用します。

```
[TotalSales]
```

FastReport は集計への参照を次のように変換します。

```
Report.GetTotalValue("TotalSales")
```

ご覧のとおり、ここではデータ型が使用されていません。これは集計値が FastReport.Variant 型の値だからです。集計値は、自動的にあらゆるデータ型に変換されるため、どのように式でも直接使用できます。たとえば、次のように指定します。

```
[TotalSales] * 0.2f
```

## レポートパラメーターの参照

レポートパラメーターを参照するには、次のようにそのパラメータ名を使用します。

```
[Parameter1]
```

パラメーターは入れ子にすることができます。この場合、次のような書式で親パラメーターおよび子パラメーターの両方を使用する必要があります。

```
[ParentParameter.ChildParameter]
```

パラメーターにはデータ型が定義されます。これはそのパラメーターの `DataType` プロパティで設定されています。このため、パラメーターを式で使用する方法は、そのパラメーターのデータ型によって異なります。

FastReport はレポートパラメーターへの参照を次のように変換します。

```
((string)Report.GetParameterValue("Parameter1"))
```

# 第5章

## スクリプト

# スクリプト

スクリプトは レポートの一部を成す高度なプログラミング言語です。スクリプトは次のいずれかの .Net 言語で記述されます。

- C#
- VisualBasic.Net

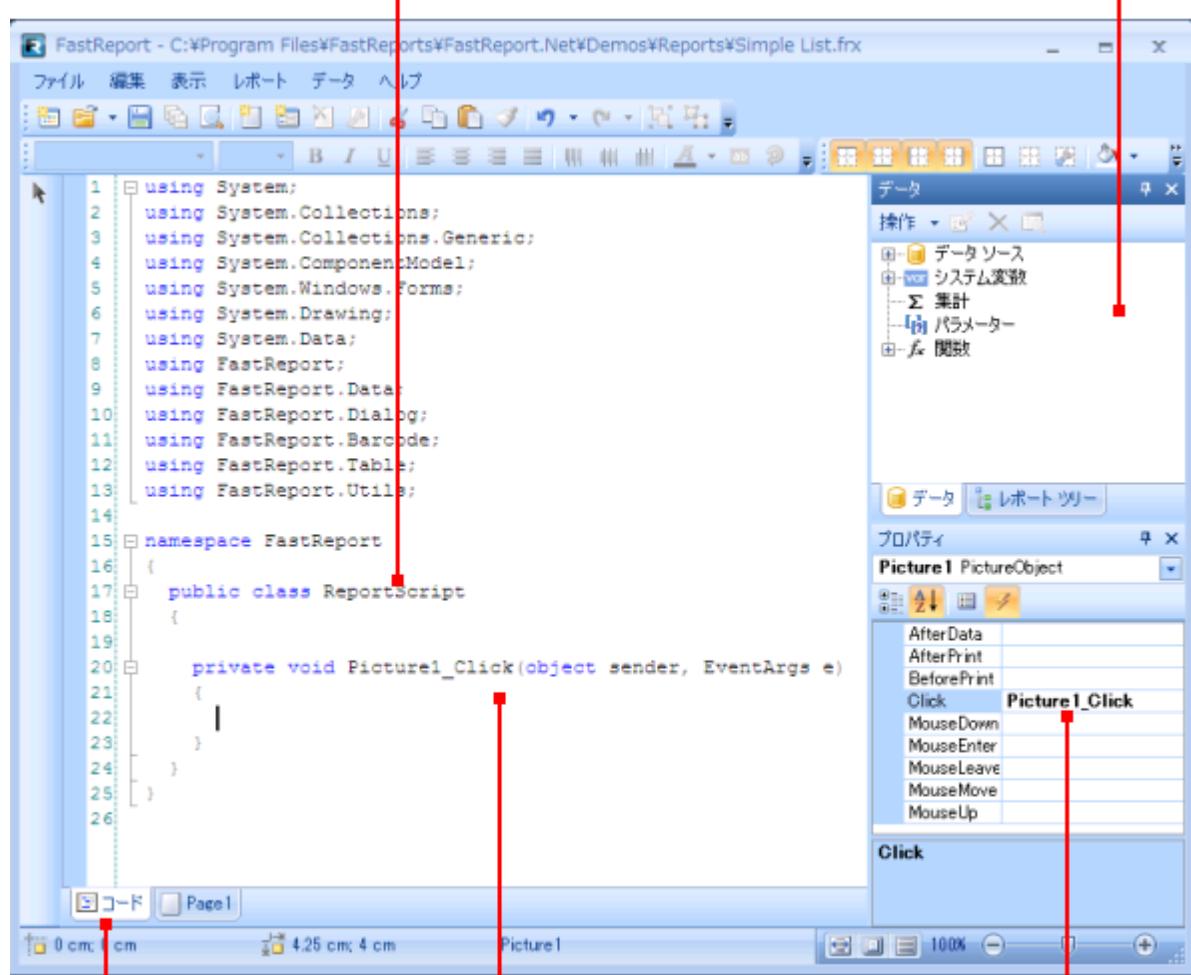
スクリプトは多くの場所で適用されます。スクリプトを使用すると 以下のことを行えます。

- FastReport エンジンの標準手段では行うことのできないデータ処理を実行する
- レポートページおよびページ上のバンドの出力を制御する
- ダイアログフォーム上の要素間の相互作用を制御する
- 動的なテーブルオブジェクトの形成を制御する
- その他多数

レポートのスクリプトを表示するには デザイナーで [コード] タブ切り替えます。

## メインクラス

スクリプトにデータ要素をドラッグできます



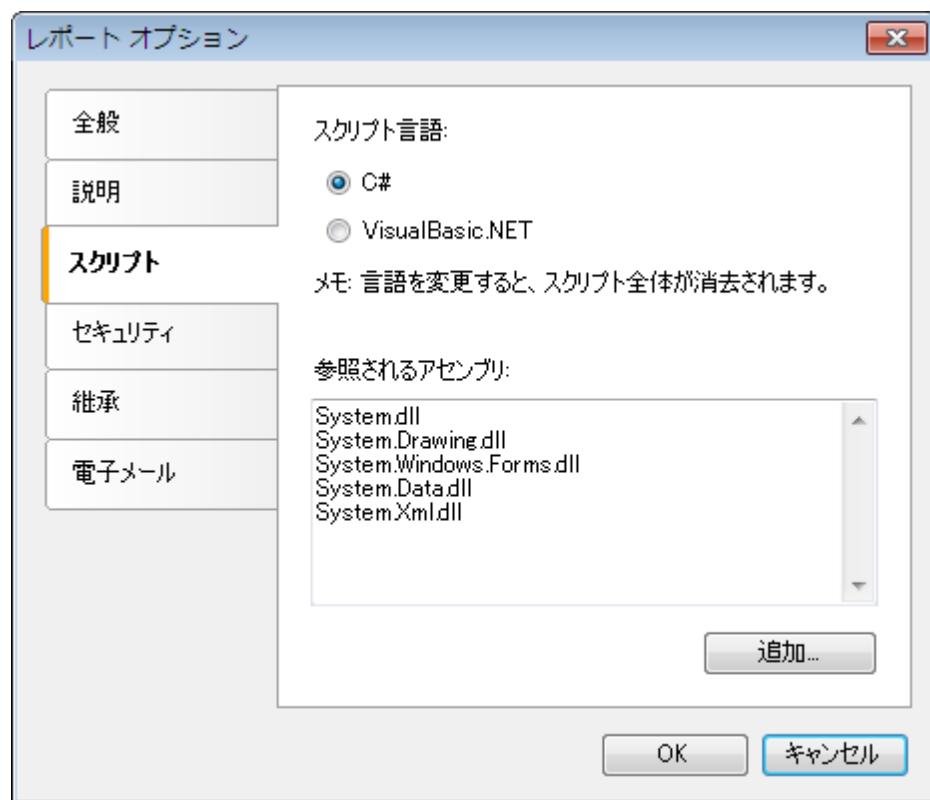
[コード]タブ

[プロパティ]ウインドウで作成された [プロパティ]ウインドウで、イベントをダブルクリック

イベントハンドラー

することにより、イベントハンドラーを作成できます

スクリプト言語は、[レポート]メニューの [オプション] を選択して設定することができます。言語が変更されると既存のスクリプトが削除されてしまうため、言語の設定はレポートを新規作成した直後に行なうようにしてください。



## 全般情報

ほかのレポートジェネレーターとは異なり FastReport のスクリプトには自分で記述したものが含まれます。スクリプトでは、以下のことができます。

- 変数、メソッドおよびプロパティをスクリプトのメインクラスに追加する。
- レポートオブジェクトのイベントハンドラーを作成する。
- 必要に応じて、新しいクラスをスクリプトへ追加する。クラスは ReportScript メインクラスの前または後のどちらにも追加できます。

以下のことを行うことはできません。

- ReportScript メインクラスの可視領域を削除、リネームまたは変更する。
- メインクラスが置かれてる名前空間を変更する。

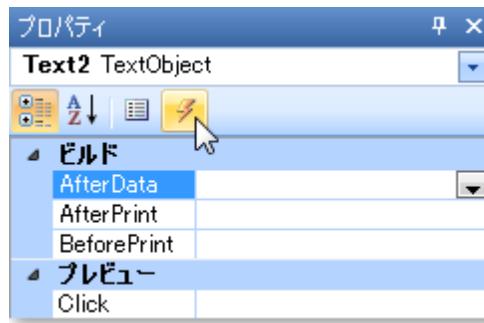
レポートの実行時には、以下のことが行われます。

- FastReport によってスクリプトに変数のリストが追加されます。変数の名前はレポートオブジェクトの名前と一致します。これはスクリプトのコンパイル前に行われるので、名前によってレポートオブジェクトを参照することができます。
- 式ハンドラーがスクリプトへ追加されます。このハンドラーはレポート内にあるすべての式を処理します。
- スクリプトが空でなければコンパイルされます。
- スクリプトクラスが初期化されます。

- レポートを実行します。

## イベントハンドラー

スクリプトは主にオブジェクトのイベントハンドラーを作成する場合に使用されます。イベントハンドラーを作成する場合は、対象となるオブジェクトを選択します。[プロパティ] ウィンドウで  ボタンをクリックし、イベントの一覧に切り替えます。



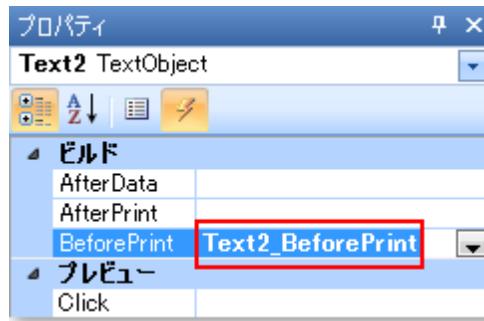
目的のイベントを選択し、ダブルクリックします。FastReport によって、レポートのコードに空のイベントハンドラーが追加されます。

```
private void Text2_BeforePrint(object sender, EventArgs e)
{
}
```

レポートオブジェクトにもイベントがあります。このオブジェクトは以下の方法で選択できます。

- [レポートツリー] ウィンドウで "Report" を選択します。
- [プロパティ] ウィンドウの上部にあるドロップダウンリストから "Report" を選択します。

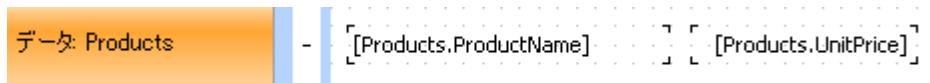
イベントハンドラーを削除するには、[プロパティ] ウィンドウでそのイベントを選択し、そのフィールドのテキストを選択して Delete キーを押します。



## レポートのイベント

レポートの制御に最大限の柔軟性を持たせるため、各レポートオブジェクトごとにいくつかのイベントがあります。たとえば、あるイベントハンドラーではデータバンドと接続し、レコードをフィルター処理、つまり特定の条件に応じてバンドを表示/非表示にすることができます。

レポート生成の処理時に発生するイベントについて見てみましょう。例として、1つのデータバンドとそのバンド上に2つのテキストオブジェクトがある1ページのシンプルなレポートを使います。



レポートの始めに、レポートオブジェクトによってStartReport イベントが発生します。レポートページを形成する前にはStartPage イベントが発生します。このイベントは各テンプレートページ（実行されるレポートページと混同しないでください）に對し1回発生します。この例の場合、テンプレートに含まれるレポートは1ページなので、実行されるレポートのページ数に関係なくこのイベントの発生は1回です。

さらに、データバンド行の出力が開始されます。これは次のように行います。

1. バンドのBeforePrint イベントが発生します。
2. そのバンドに置かれているすべてのオブジェクトのBeforePrint イベントが発生します。
3. すべてのオブジェクトがデータで埋められます。
4. そのバンドに置かれているすべてのオブジェクトのAfterData イベントが発生します。
5. バンドのBeforeLayout イベントが発生します。
6. オブジェクトがバンド上に置かれ、そのバンドの高さが計算され（可能な場合は）バンドが拡張されます。
7. バンドのAfterLayout イベントが発生します。
8. バンドがページ上の空き領域に収まらない場合は、新しいページが形成されます。
9. バンドおよびそのすべてのオブジェクトが実行レポートページに表示されます。
10. バンドのAfterPrint イベントが発生します。
11. すべてのバンドオブジェクトのAfterPrint イベントが発生します。

データソースにあるすべてのデータがバンド行として出力されます。これで、この例のレポートの形成は終了です。ページのFinishPage イベントが発生し、最後にReport オブジェクトのFinishReport イベントが発生します。

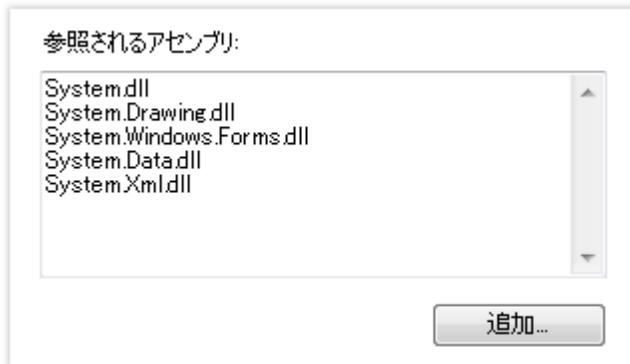
このため、異なるオブジェクトのイベントを使用することで、レポート形成の各手順を制御することができます。イベントを正しく使用するためには、上記の11個の手順で述べたバンドの出力プロセスについて十分に理解していることが重要です。多くの操作はバンドのBeforePrint イベントの使用だけ行えます。オブジェクトに対して行われる変更も表示されます。しかし、このイベントでは、バンドの出力でそのバンドを拡張する場合のページ分析を行うことはできません。これはバンドの高さが手順6で計算されるためです。これは、手順7でAfterLayout イベント、または手順10でAfterPrint イベントを使用して行えますが、後者のAfterPrint の場合は、バンドが既に表示されておりオブジェクトの操作は何も割り当てられません。要するに、各イベントをいつ発生させるか、また指定したタスクに適したイベントの使用について明確に提示する必要があります。

## .Net オブジェクトの使用

スクリプトでは、.Net オブジェクトを使用することができます。これらは以下のアセンブリで定義されています。

```
System.dll
System.Drawing.dll
System.Windows.Forms.dll
System.Data.dll
System.Xml.dll
```

そのほか FastReport アセンブリで定義されているオブジェクトも使用できます。別のアセンブリにアクセスする必要がある場合は、それをアセンブリのリストに追加します。これは [レポートオプション] ダイアログ ([レポート] メニューの [オプション] を選択) の [スクリプト] タブで行なうことができます。



たとえば、アプリケーション内で宣言した関数をレポートで使用する場合、アプリケーションアセンブリ(exe または.dll)をレポートのアセンブリリストに追加します。これで、アプリケーションの名前空間を使用することでその関数を呼び出すことができます。たとえば、次の関数がアプリケーションで定義されています。

```
namespace Demo
{
    public static class MyFunctions
    {
        public static string Func1()
        {
            return "Hello!";
        }
    }
}
```

スクリプトでの呼び出しあは次のように行なうことができます。

```
string hello = Demo.MyFunctions.Func1();
```

レポートのスクリプトの先頭で "using Demo" 行を追加すると、次のように構文を短縮することができます。

```
string hello = MyFunctions.Func1();
```

## レポートオブジェクトの参照

レポートオブジェクト(テキストオブジェクトなど)を参照する場合は、そのオブジェクトの名前を使用します。次の例はText1 オブジェクトの高さを返します。

```
float height = Text1.Height;
```

レポートのネイティブの測定単位はピクセルであることに留意してください。オブジェクトで Left、Top、Width および Height のようなプロパティを使用する場合はこの単位に留意してください。ピクセルをセンチメートリに変換および逆変換するには、次のように Units クラスで定義されている定数を使用します。

```
float heightInPixels = Text1.Height;  
float heightInCM = heightInPixels / Units.Centimeters;  
Text1.Height = Units.Centimeters * 5; // 5 cm
```

## Report および Engine オブジェクト

レポートに含まれるオブジェクトとは別に、スクリプトで定義される Report および Engine という2 つの変数があります。

Report 変数は現在のレポートを参照します。次の表は Report オブジェクトのメソッドの一覧を示します。

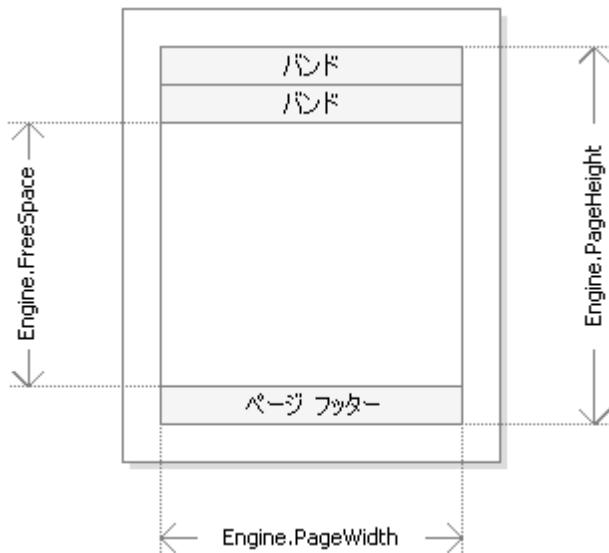
メソッド	説明
object Calc( string expression)	式を計算して、その結果の値を返します。このメソッドを初めて呼び出すとき、必要となる式がコンパイルされます。
object GetColumnValue( string complexName)	データ列の値を返します。データ列の名前は "DataSource.Column" ("データソース列" )の形式で表します。列に Null 値がある場合は、そのデータ型に応じた既定値 (0、空文字列、False )に変換されます。
object GetColumnValueNullable( string complexName)	データ列の値を返します。前のメソッドとは異なり、列に Null 値があっても既定値には変換されません。
Parameter GetParameter( string complexName)	指定された名前のレポートパラメーターを返します。入れ子のパラメーターを参照する場合、名前は "MainParam.NestedParam" ("メインパラメーター.入れ子のパラメーター" )の形式で指定することができます。
object GetParameterValue( string complexName)	指定された名前のレポートパラメーターの値を返します。

<code>void SetParameterValue(     string complexName,     object value)</code>	指定された名前のレポートパラメーターの値を設定します。
<code>object GetVariableValue(     string complexName)</code>	システム変数 (Date など)の値を返します。
<code>object GetTotalValue(     string name)</code>	集計の値を返します。この集計は名前が付けられ [データ] ウィンドウで定義されています。
<code>DataSourceBase GetDataSource(     string alias)</code>	レポートに名前で定義されているデータソースを返します。

Engine オブジェクトはレポートの作成を制御するエンジンです。このエンジンのメソッドやプロパティを使用すると、バンドをページ上に配置する処理を管理することができます。Engine オブジェクトの以下のプロパティを使用することができます。

プロパティ	説明
<code>float CurX</code>	X 軸における現在の座標。このプロパティに値を割り当て、出力オブジェクトを移動させることができます。
<code>float CurY</code>	Y 軸における現在の座標。このプロパティに値を割り当て、出力オブジェクトを移動させることができます。
<code>int CurColumn</code>	複数列 レポートにおける現在の列の番号。先頭の列の番号は 0 です。
<code>int CurPage</code>	出力されるページの番号。この値は Page システム変数を用いて取得することができます。
<code>float PageWidth</code>	左余白と右余白のサイズを差し引いたページの幅。
<code>float PageHeight</code>	上余白と下余白のサイズを差し引いたページの高さ
<code>float PageFooterHeight</code>	ページ フッター (およびそのすべての子バンド) の高さ
<code>float ColumnFooterHeight</code>	列 フッター (およびそのすべての子バンド) の高さ
<code>float FreeSpace</code>	ページ上の空き領域のサイズ
<code>bool FirstPass</code>	最初の (または唯一の) レポートパスが実行された場合は <code>True</code> を返します。パスの数は <code>Report.DoublePass</code> プロパティを用いて取得できます。
<code>bool FinalPass</code>	最後の (または唯一の) レポートパスが実行された場合は <code>True</code> を返します。

次の図で、上記のいくつかのプロパティについて理解を深めることができます。



`Engine.PageWidth` および `Engine.PageHeight` プロパティは出力領域のサイズを決定します。この領域はほとんどの場合ページの実際のサイズよりも小さくなっています。出力領域のサイズはページ余白によって決まります。この余白は `LeftMargin`、`TopMargin`、`RightMargin` および `BottomMargin` プロパティで指定されます。

`Engine.FreeSpace` プロパティはページ上の空き領域の高さを決定します。ページ上にページ フッター バンドがある場合、その高さは空き領域 (FreeSpace) の計算時に考慮されます。バンドの出力後の空き領域が縮小されるので注意してください。

実行されるレポートのページがどのように形成されるかについて説明します。FastReport エンジンは、ページ上にバンドを出力する十分なスペース (空き領域) がある間はバンドを表示します。空き領域がなくなると、ページ フッター バンドが outputされ新しい空のページが形成されます。X および Y 座標によって指定される現在の位置から開始するバンドを表示します。この位置は `Engine.CurX` および `Engine.CurY` プロパティで返されます。1 つのバンドの出力後、その出力されたバンドの高さが自動的に `CurY` へ加算されます。新しいページが形成されると `CurY` の位置は 0 に設定されます。複数列のレポートを出力する場合、`CurX` の位置は変更します。

`Engine.CurX` および `Engine.CurY` プロパティは取得だけでなく設定にも使用できます。これは適切なイベントを使用することでバンドを手動で移動させることができるということです。これらのプロパティの使用例については、「例」セクションを参照してください。

サイズや位置を返すプロパティを使って作業する場合、これらのプロパティの値はピクセル単位で測定されます。

Engine オブジェクトでは、以下のメソッドが定義されています。

メソッド	説明
<code>void AddOutline(string text)</code>	要素をレポートアウトラインに追加し ( <a href="#">対話型レポート</a> を参照)、その追加された要素を現在の位置として設定します。
<code>void OutlineRoot()</code>	現在の位置をアウトラインのルートに設定します。
<code>void OutlineUp()</code>	現在の位置を 1 つ上のレベルのアウトライン要素に移動させます。

void AddBookmark(string name)	ブックマークを追加します (対話型レポートを参照)。
int GetBookmarkPage(string name)	指定された名前のブックマークがあるページ番号を返します。
void StartNewPage()	新しいページを開始します。複数列のレポートの場合、新しい列が開始されます。

AddOutline、OutlineRoot、OutlineUp メソッドを使用すれば、レポートアウトラインを手動で形成することができます。通常、これは各バントおよびレポートページにあるOutlineExpression プロパティを使用して自動的に行われます。

AddOutline メソッドは現在のアウトライン要素に子要素を追加し、その要素に現在の位置設定します。現在のレポートページおよびそのページ上の現在の位置が新しい要素と関連付けられます。AddOutline メソッドを複数回呼び出した場合は、以下のよう構造になります。

Item1

  Item2

    Item3

現在の要素を制御する場合は、OutlineUp およびOutlineRoot メソッドを使用します。OutlineUp メソッドは、ポインターを1つ上のレベルにある要素へ移動させます。これは次のようなスクリプトになります。

```
Engine.AddOutline("Item1");
Engine.AddOutline("Item2");
Engine.AddOutline("Item3");
Engine.OutlineUp();
Engine.AddOutline("Item4");
```

これによって次のようなアウトラインが作成されます。

Item1

  Item2

    Item3

      Item4

OutlineRoot メソッドは、現在の要素をアウトラインのルートへ移動させます。たとえば、次のようなスクリプトを記述します。

```
Engine.AddOutline("Item1");
Engine.AddOutline("Item2");
Engine.AddOutline("Item3");
Engine.OutlineRoot();
Engine.AddOutline("Item4");
```

これによって次のようなアウトラインが作成されます。

Item 1

Item 2

Item 3

Item 4

ブックマークを使って作業する場合、Engine オブジェクトの AddBookmark および GetBookmarkPage メソッドが使用されます。通常、ブックマークは Bookmark プロパティを使用すると自動的に追加されます。このプロパティはレポートのすべてのオブジェクトにあります。

AddBookmark メソッドを使用することで、プログラムでブックマークを追加することができます。このメソッドは現在のページの現在の出力位置にブックマークを作成します。

GetBookmarkPage メソッドはブックマークが置かれるページ番号を返します。このメソッドは目次の作成時、ページ番号を表示するためによく使用されます。この場合、レポートはダブルパスを持つ必要があります。

## データ ソースの参照

FastReport の式 (式の章を参照) とは異なり、データソースの参照には角かっこをスクリプトで使用しないでください。その代わりに、列の値を返す Report オブジェクトの GetColumnValue メソッドを使用します。

```
string productName = (string)Report.GetColumnValue("Products.Name");
```

このように、ソースおよびその列の名前を指定する必要があります。関係を使用してデータソースを参照している場合は、データソース名を含めることができます。関係の詳細については、データの章の関係を参照してください。たとえば、次のように記述すると、関連付けられたデータソースの列を参照することができます。

```
string categoryName = (string)Report.GetColumnValue("Products.Categories.CategoryName");
```

この作業は [データ] ウィンドウを使用するもとと簡単に行えます。[データ] ウィンドウからデータ要素をスクリプトへドラッグすることができます。このとき、FastReport によってその要素の参照用コードが自動的に作成されます。

データソース自身を参照する場合は、Report オブジェクトの GetDataSource メソッドを使用します。

```
DataSourceBase ds = Report.GetDataSource("Products");
```

DataSourceBase クラスのプロパティやメソッドの詳細については、FastReport.NET Class Reference ヘルプを参照してください。一般に、このオブジェクトはスクリプトで以下のように使用されます。

```
// データソースへの参照を取得する
DataSourceBase ds = Report.GetDataSource("Products");

// データソースを初期化する
```

```

ds.Init();

// すべての行を列挙する
while (ds.HasMoreRows)
{
    // 現在の行からデータ列の値を取得する
    string productName = (string)Report.GetColumnValue("Products.Name");
    // その値を使ってなんらかの作業を行う
    // ...

    // 次のデータ行へ
    ds.Next();
}

```

## システム変数の参照

システム変数を参照する場合は Report オブジェクトの GetVariableValue メソッドを使用します。

```
DateTime date = (DateTime)Report.GetVariableValue("Date");
```

システム変数の一覧は [データ] ウィンドウに表示されます。そこから変数をスクリプトへドラッグすることができます。このとき FastReport によってその変数の参照用コードが自動的に作成されます。

## 集計値の参照

集計値を参照する場合は Report オブジェクトの GetTotalValue メソッドを使用します。

```
float sales = Report.GetTotalValue("TotalSales");
```

集計の一覧は [データ] ウィンドウに表示されます。そこから集計をスクリプトへドラッグすることができます。このとき FastReport によってその集計の参照用コードが自動的に作成されます。

集計値には FastReport.Variant 型が定義されます。この FastReport.Variant 型は自動的にあらゆるデータ型に変換されるため、集計値は式で直接使用することができます。たとえば 次のように指定します。

```
float tax = Report.GetTotalValue("TotalSales") * 0.2f;
```

集計値の参照は、それが処理される時点で行うことができます。通常、集計はレポートでそれが置かれているバンドを出力する際にすぐに使用できます。

## レポートパラメーターの参照

レポートパラメーターを参照する場合は Report オブジェクトの GetParameterValue メソッドを使用します。

```
int myParam = (int)Report.GetParameterValue("MyParameter");
```

パラメーターは入れ子にすることができます。この場合、次のように親パラメーターの名前、ピリオド、子パラメーターの名前の順で指定します。

```
Report.GetParameterValue("ParentParameter.ChildParameter")
```

パラメーターにはデータ型が定義されます。これはそのパラメーターの DataType プロパティで指定されています。パラメーターを参照する場合はこのデータ型を考慮する必要があります。パラメーターの一覧は、[データ] ウィンドウに表示されます。そこからパラメーターをスクリプトへドロップすることができます。このとき、FastReport によってそのスクリプトの参照用コードが自動的に作成されます。

パラメーターの値を変更するには Report オブジェクトの SetParameterValue メソッドを使用します。

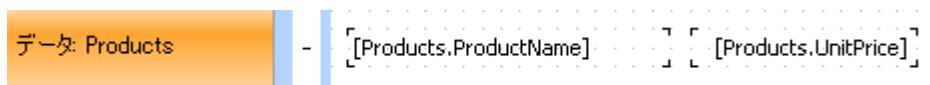
```
Report.SetParameterValue("MyParameter", 10);
```

## 例 1 :オブジェクトの外観を変更する

この例では、オブジェクトに出力される値に応じてテキストの色を変更する方法を示します。ここで使用するものは次のとおりです。

- BeforePrint イベント
- スクリプトからデータ列への参照

次のような外観のシンプルなレポートを作成します。



UnitPrice 列を出力するオブジェクトを選択し、BeforePrint イベントハンドラーを作成します。

```
private void Text2_BeforePrint(object sender, EventArgs e)
{
    if (((Decimal)Report.GetColumnValue("Products.UnitPrice")) > 2000)
        Text2.TextColor = Color.Red;
}
```

Products.UnitPrice データ列をスクリプトへ挿入するには、[データ] ウィンドウからそのデータ列をドラッグします。データ列がドラッグされるごとに、次の文字列がスクリプトへ追加されます。

```
((Decimal)Report.GetColumnValue("Products.UnitPrice"))
```

レポートを実行すると、製品の一覧が出力され、価格が2000 を超えるテキストは赤色で強調表示されます。

Chai	1,800
Chang	1,900
Aniseed Syrup	1,000
Chef Anton's Cajun Seasoning	2,200
Chef Anton's Gumbo Mix	2,135
Grandma's Boysenberry Spread	2,500

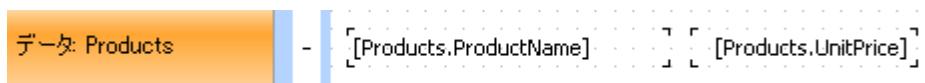
強調表示の条件 (「レポートの作成」の章の[条件付き強調表示](#)を参照)を使用するとこれと同じように出力することができます。

## 例 2 : バンドの偶数行を強調表示する

この例では、データバンドにおける偶数行の塗りつぶしの色を変更する方法を示します。ここで使用するものは次のとおりです。

- ・ バンドの BeforePrint イベント
- ・ スクリプトから Row# システム変数への参照

次のように外観のシンプルなレポートを作成します。



次のように、バンドの BeforePrint イベントハンドラーを作成します。

```
private void Data1_BeforePrint(object sender, EventArgs e)
{
    if (((Int32)Report.GetVariableValue("Row#")) % 2 == 0)
        Data1.FillColor = Color.Gainsboro;
}
```

Row# システム変数は出力されるバンドの行番号を返します。変数の参照をスクリプトへ挿入するには、[データ] ウィンドウからその変数をドラッグします。変数をドラッグすると、次の文字列がスクリプトへ追加されます。

```
((Int32)Report.GetVariableValue("Row#"))
```

レポートを実行すると、次のように偶数行が淡い灰色で強調表示されます。

Chai	1,800
Chang	1,900
Aniseed Syrup	1,000
Chef Anton's Cajun Seasoning	2,200
Chef Anton's Gumbo Mix	2,135

データバンドの EvenStyle プロパティを使用すると、これと同じように出力することができます。この詳細については、「レポートの作成」の章

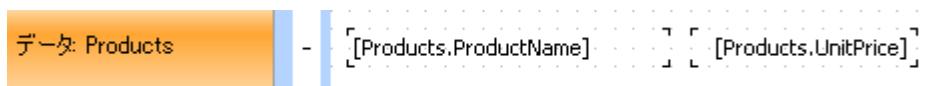
のデータの偶数行を強調表示するを参照してください。

### 例 3 :データをフィルター処理する

この例では、指定した条件に従ってデータバンド行を非表示にする方法を示します。ここで使用するものは次のとおりです。

- ・ バンドの BeforePrint イベント
- ・ スクリプトからデータソースへの参照

次のように外観のシンプルなレポートを作成します。



次のように、バンドの BeforePrint イベントハンドラーを作成します。

```
private void Data1_BeforePrint(object sender, EventArgs e)
{
    if (((Decimal)Report.GetColumnValue("Products.UnitPrice")) > 2000)
        Data1.Visible = false;
}
```

この場合、レポートを実行すると UnitPrice の値が 2000 を超えるバンド行が表示されなくなります。

Chai	1,800
Chang	1,900
Aniseed Syrup	1,000
Konbu	600
Genen Shouyu	1,550
Pavlova	1,745

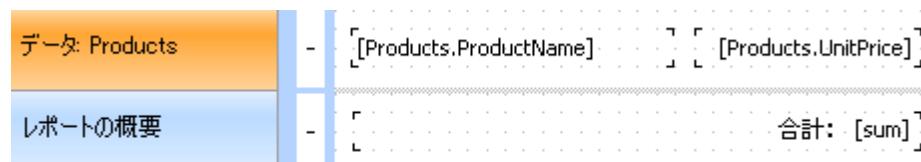
データバンドエディターで設定できるデータフィルターを使用すると、これと同じように出力することができます。

## 例 4 :合計を計算する

この例では、プログラミング方法を使用して合計を計算する方法を示します。ここで使用するものは次のとおりです。

- ・バンドのBeforePrint イベント
- ・スクリプトからデータ列への参照
- ・ローカル変数 (レポートに outputされる値を含む)

次のような外観のレポートを作成します。



スクリプトで、"sum" という変数を宣言し、このバンドに属するBeforePrint イベントハンドラーを作成します。

```
public class ReportScript
{
    private decimal sum;

    private void Data1_BeforePrint(object sender, EventArgs e)
    {
        sum += (Decimal)Report.GetColumnValue("Products.UnitPrice");
    }
}
```

Products.UnitPrice データ列をスクリプトへ挿入するには、[データ] ウィンドウからそのデータ列をドロップします。

レポートを実行すると、次のように出力されます。

Röd Kaviar	1,500
Longlife Tofu	1,000
Rhönbräu Klosterbier	775
Lakkalikööri	1,800
Original Frankfurter grüne Soße	1,300
合計: 222271	

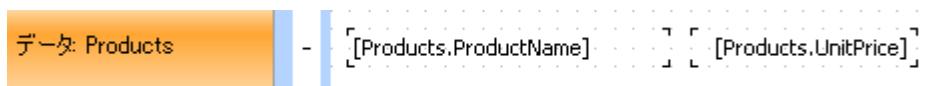
集計を使用すると、これと同じように出力することができます。

## 例 5 :出力位置を移動する

この例では、Engine オブジェクトを使用し、手動でバンドの位置を移動する方法を示します。ここで使用するものは次のとおりです。

- バンドの BeforePrint イベント
- Engine オブジェクト

次のように外観のシンプルなレポートを作成します。



次のように、バンドの BeforePrint イベントハンドラーを作成します。

```
private void Data1_BeforePrint(object sender, EventArgs e)
{
    Engine.CurX = ((Int32)Report.GetVariableValue("Row#")) * 10;
}
```

レポートを実行すると、次のように出力されます。

Chai	1,800
Chang	1,900
Aniseed Syrup	1,000
Chef Anton's Cajun Seasoning	2,200
Chef Anton's Gumbo Mix	2,135



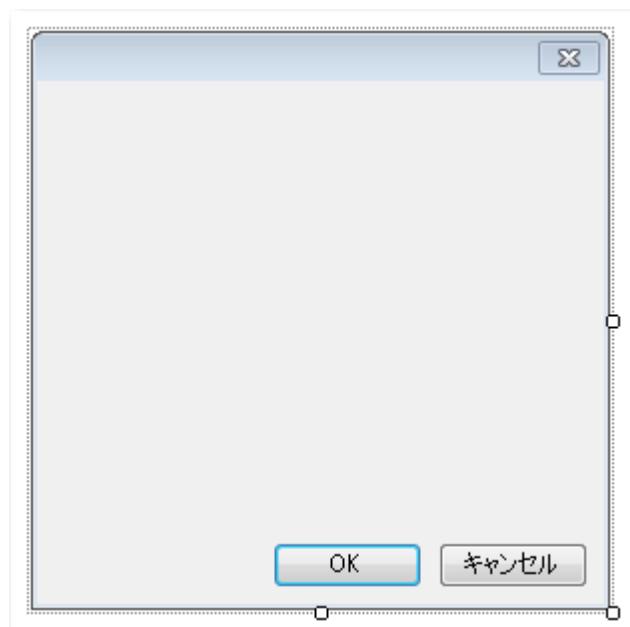
# 第6章

## ダイアログ フォーム

## ダイアログ フォーム

レポートには、通常のページとは別に1つまたは複数のダイアログ フォームを含めることができます。このようなダイアログ フォームはレポートの実行時に表示されます。ダイアログでは、レポートの作成に必要なあらゆる情報を入力できます。またダイアログを使用すれば、レポートに表示されるデータをフィルター処理することもできます。

ダイアログをレポートに追加するには、デザイナーのツールバーにある  ボタンをクリックします。次のような新しいダイアログが表示されます。



1つまたは複数のダイアログを持つレポートは次のように動作します。

- レポートを実行すると、最初のダイアログが表示されます。
- [OK] ボタンをクリックしてそのダイアログを閉じると、次のダイアログが表示されます。[キャンセル] ボタンをクリックすると、レポートの実行作業は停止します。
- すべてのダイアログが表示された後、レポートが作成されます。

## コントロール

ダイアログ フォーム上で、以下のコントロールを使用することができます。

アイコン	名前	説明
	ButtonControl	Windows Button コントロールを表します。

	CheckBoxControl	Windows CheckBox コントロールを表します。
	CheckedListBoxControl	リストボックスを表示します。ボックス内の各リスト項目の左側にはチェックボックスが付きます。
	ComboBoxControl	Windows ComboBox コントロールを表します。
	DataGridViewControl	カスタマイズ可能なグリッドにデータを表示します。
	DataSelectorControl	2つのリストを表示します。リスト間で項目を一方のリストから他方のリストへ移動させることができます。
	DateTimePickerControl	ユーザーが日時を選択し、書式を指定してその日時を表示できるWindows DateTimePicker コントロールを表します。
	GroupBoxControl	コントロールのグループの周りにフレームを表示する、キャプション付きまたはキャプションなしのWindows GroupBox コントロールを表します。
	LabelControl	Windows の標準ラベルを表します。
	ListBoxControl	項目の一覧を表示するためのWindows ListBox コントロールを表します。
	ListViewControl	これは4つの異なるビューのうちの1つで項目を表示するWindows ListView コントロールを表します。
	MaskedTextBoxControl	ユーザー入力が適切か不適切かを区別するマスクを使用します。
	MonthCalendarControl	グラフィカルな月カレンダーを使用してユーザーが日付を選択できるWindows MonthCalendar コントロールを表します。
	NumericUpDownControl	数値を表示するWindows スピンボックス(アップダウンコントロール)を表します。
	PanelControl	コントロールのコレクションをグリーブ化するために使用されます。
	PictureBoxControl	イメージを表示するためのWindows PictureBox コントロールを表します。
	RadioButtonControl	ほかのRadioButton コントロールと組み合わせることで、ユーザーは選択肢のグループから1つの選択肢を選ぶことができます。
	RichTextBoxControl	Windows のRichTextBox コントロールを表します。
	TextBoxControl	Windows のTextBox コントロールを表します。
	TreeViewControl	ラベル付き項目の階層コレクションを表示します。

DataSelectorControl 以外のコントロールはすべて Windows フォームの標準 コントロールと同じ機能を持ちます。コントロールの名前は重複を避けるため、末尾に"Control"というサフィックスが付けられています。このため、FastReport のButtonControl は Windows の標準

標準 Button コントロールに対応します。

## コードからのコントロールの参照

コントロールは、次のようにその名前を使用して参照することができます。

```
TextBoxControl1.Text = "my text";
```

実際、FastReport のコントロールは .Net 標準コントロールのラッパーです。標準コントロールの多くのプロパティをラップします（すべてのプロパティをラップするわけではありません）。FastReport コントロールでインプリメントされないプロパティが必要な場合は、以下の方法で、ラップされる標準コントロールにアクセスできます。

- Control プロパティを使用します。これは System.Windows.Forms.Control 型です。

```
(TextBox1.Control as TextBox).ShortcutsEnabled = false;
```

- コントロール自身と同じ名前を持つプロパティを使用します。ただし、"Control" サフィックスは付きません。たとえば、TextBoxControl には TextBox プロパティがあります。これは System.Windows.Forms.TextBox 型で、ラップされた TextBox コントロールを返します。

```
TextBox1.TextBox.ShortcutsEnabled = false;
```

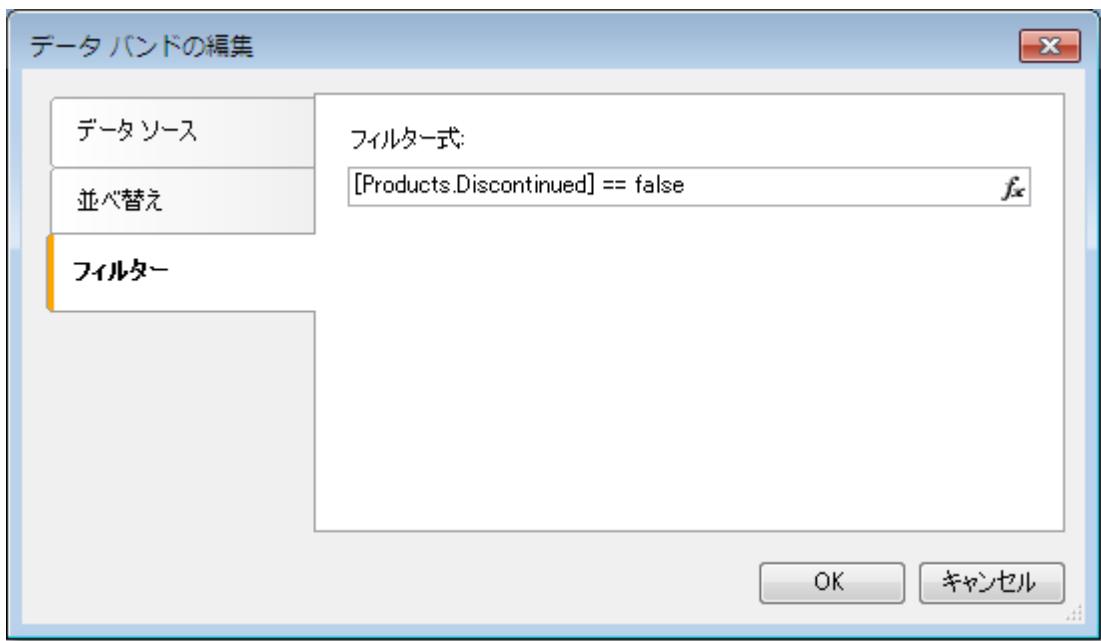
コントロールのプロパティおよびメソッドの詳細については、MSDN を参照してください。

## データのフィルター処理

ダイアログを使用して、レポートに出力されるデータをフィルター処理することができます。たとえば、全従業員の一覧を出力するレポートがあるとします。ダイアログを使用すれば、全従業員から 1 人または複数人の従業員を選択することができます。その結果、レポートの作成時に選択した従業員のみが表示されるようにデータがフィルター処理されます。

データのフィルター処理を行うため、初期レポートにはすべてのデータが含まれている必要があります。"フィルター処理" は、レポートの作成時に不要なデータが表示されないようにすることを前提とします。

データのフィルター処理を準備するには、データバンドの Filter プロパティを使用するのが最も簡単な方法です。バンドエディター（[データバンドの編集]）には、次のようなフィルター式を指定できます。



ダイアログを使用すれば、ユーザーに値の指定を要求し、その値をフィルター式で使用することができます。シンプルなフィルター処理の例については、[例 5 シンプルなフィルター](#)を参照してください。

単一の値が必要な場合は、この方法が使用できます。値の一覧を表示し、その一覧から1つまたは複数の値を照会するタスクの場合、この実装は難しくなります。これは ListBoxControl コントロール要素に従業員の一覧を表示し、その中から1つまたは複数の値を選択するという単純なタスクだと考えることができます。これを実装するためには、以下のことを行うスクリプトを使用する必要があります。

- データソースを名前で取得する
- データを初期化する
- ListBoxControl をデータソースのデータで埋める
- 従業員の選択後、データバンドで使用されるフィルター式を作成する

FastReport ではこれを自動的に行うことができます。このため、次のセクションで説明する自動フィルタリングが使用されます。

## 自動フィルタリング - 動作方法

コントロールは DataColumn プロパティを使用してデータ列へ接続されます。値のリストを表示できるコントロールの場合 (ListBoxControl など)、そのコントロールは指定されたデータ列の値で埋められます。これはダイアログが表示されたときに自動的に行われます。さらに、ユーザーがこのダイアログを使用し、コントロール内で1つまたは複数の項目を選択してダイアログを閉じます。このとき、DataColumn プロパティは指定されたデータソースは自動的にフィルター処理されます。

この方法の利点は、コードを記述することなくレポートで自動フィルタリングを使用できるということです。

自動フィルタリングは以下のコントロールでサポートされています。

アイコン	名前
	CheckBoxControl
	CheckedListBoxControl
	ComboBoxControl
	DataSelectorControl
	DateTimePickerControl
	ListBoxControl
	MaskedTextBoxControl
	MonthCalendarControl
	NumericUpDownControl
	RadioButtonControl
	TextBoxControl

## フィルター演算

既定では FastReport はコントロールの値と等しい値が含まれまるデータ行をフィルター処理します。この動作はコントロールの FilterOperation プロパティで定義されます。以下の演算を使用できます。

演算	演算子	効果
Equal	=	コントロールの値と等しい値をフィルター処理します。
NotEqual	<>	コントロールの値と等しくない値をフィルター処理します。
LessThan	<	コントロールの値よりも小さい値をフィルター処理します。
LessThanOrEqual	<=	コントロールの値以下の値をフィルター処理します。
GreaterThan	>	コントロールの値よりも大きい値をフィルター処理します。
GreaterThanOrEqual	>=	コントロールの値以上の値をフィルター処理します。

たとえば、コントロールの FilterOperation プロパティに "LessThanOrEqual" を設定し、コントロールに値 "5" を入力した場合は、列の値が 5 以下のデータ行がすべて選択されます。

データが文字列型の場合、さらに以下の演算を使用することができます。

演算	効果
Contains	コントロールの値が含まれる値をフィルター処理します。
NotContains	コントロールの値が含まれない値をフィルター処理します。
StartsWith	コントロールの値で始まる値をフィルター処理します。
NotStartsWith	コントロールの値で始まらない値をフィルター処理します。
EndsWith	コントロールの値で終わる値をフィルター処理します。
NotEndsWith	コントロールの値で終わらない値をフィルター処理します。

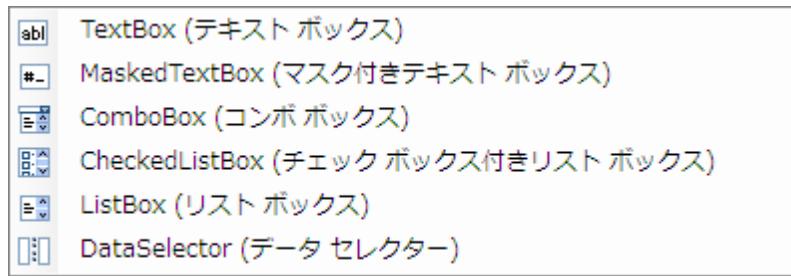
たとえば、コントロールのFilterOperation プロパティに"StartsWith" を設定し、コントロールに値 "A" を入力した場合は、列の値の先頭文字が"A" のデータ行がすべて選択されます。

## レポートへのフィルターの追加

FastReport のダイアログデザイナーでは、データのフィルター処理をサポートするコントロールを簡単に追加することができます。コントロールを追加するには、[データ] ウィンドウからダイアログ フォームへデータ列をドラッグアンドドロップします。このとき、FastReport によってヘッダー (LabelControl コントロール) と データのフィルター処理に使用される実際のコントロールが作成されます。



コントロールの種類はデータ列のデータ型に応じて決定します。データ列が文字列型の場合は、フォーム上にそのデータ列が挿入されたときに、コントロールの種類の選択を要求されます。



同一のデータソースに接続される2つの同じコントロールを挿入した場合、FastReportはFilterOperationプロパティを使用して自動的にデータ範囲を設定します。最初のコントロールのFilterOperationプロパティには"GreaterThanOrEqual"、2番目のコントロールのFilterOperationプロパティには"LessThanOrEqual"が設定されます。これが設定されるのは、文字列型以外のデータ列を挿入した場合です。

レポートにデータのフィルター処理を追加する場合は、以下のことを行う必要があります。

- 新しいダイアログをレポートに追加します。
- レポートでフィルター処理の対象とするデータ列をダイアログ上にドロップします。

## データ範囲によるフィルタリング

このフィルタリング方法は、価格などの定量的な特性を持つ値を扱う場合に利用できます。指定した価格より低いまたは高い価格の商品をフィルター処理することができます。要素に入力された値を解析する方法を指定するには、前述のFilterOperationプロパティを使用します。

2つの同じコントロールを使用するとき、これらが同じデータ列に接続され、かつFilterOperationプロパティの設定がそれぞれ異なる場合は、データ範囲の開始と終了を指定できます。最初のコントロールのFilterOperationプロパティには"GreaterThanOrEqual"、2番目のコントロールのFilterOperationプロパティには"LessThanOrEqual"を指定する必要があります。

## 関連付けられたデータ列によるフィルタリング

既に説明したとおり、2つのデータソース間では関係を設定することができます。詳細については、[データの章](#)を参照してください。関係を使用すると、データソースのデータを、別のデータソースのデータ列を使用してフィルター処理することができます。

ダイアログにListBoxControlが置かれ、DataColumnプロパティに次のデータ列が指定されていると仮定します。

Products.Categories.CategoryName

フィルター処理は次のように動作します。

- コントロールを値で埋める場合、CategoriesデータソースのCategoryName列が使用されます。
- フィルターはProductsテーブルに適用されます。そのデータ行は次の条件を対象にフィルター処理されます。

ユーザーが選択した値の1つが[Products.Categories.CategoryName]に含まれる

## カスケードリストを使用したフィルタリング

カスケードリストは、他のリストボックスで選択された値に従って選択肢が変化するリストボックスです。たとえば、フォーム上に2つのリストがあるとします。最初のリストにはカテゴリ、2番目のリストには製品が含まれています。最初のリストからカテゴリを選択すると、2番目のリストには選択したカテゴリに属する製品がリストアップされます。



カスケードリストを作成するには、マスター/詳細関係を持つ2つのデータソースを使用する必要があります（データソースおよび関係の詳細については、データの章の関係を参照してください）。マスター リストはマスター データソースの列に接続し、詳細 リストは詳細 データソースの列に接続します。また、マスターのリストの DetailControl プロパティには、詳細 リストを設定します。

## コードからのフィルタリング制御

自動フィルタリングで十分間に合うことが多いとしても、手動でフィルタリングを管理する可能性もあります。このため、以下のメソッドおよびプロパティが使用されます。

AutoFill プロパティは、コントロールをデータで埋めるのを制御します。これは値の一覧を表示できるコントロール（ListBoxControl など）で使用されます。ダイアログを表示する前に、そのコントロールにはデータが挿入されます。既定では、このプロパティには True が設定されます。これが無効（False）の場合、コントロールにはデータが挿入されないため、次のように FillData メソッドを呼び出して手動で行う必要があります。

```
ListBox1.FillData();
```

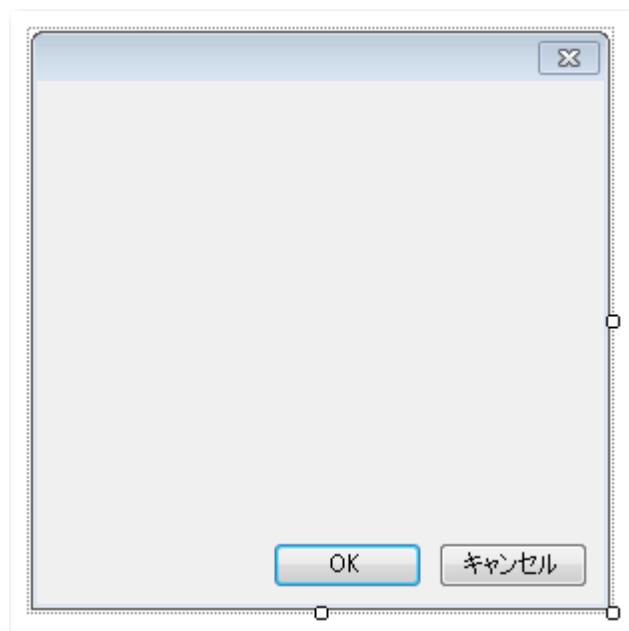
AutoFilter プロパティは、データのフィルター処理を制御します。これはすべてのコントロールで使用されます。[OK] ボタンをクリックしてダイアログを閉じると、FastReport はデータのフィルターを自動的に適用します。既定では、このプロパティには True が設定されます。これが無効（False）の場合、フィルター処理は実行されないので、次のように FilterData メソッドを呼び出して手動で行う必要があります。

```
ListBox1.FilterData();
```

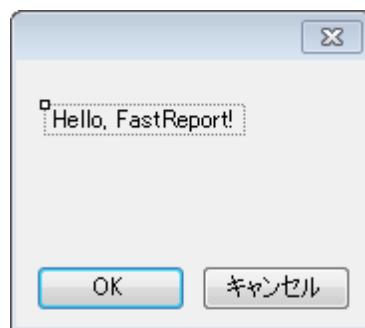
## 例

### 例 1 :"Hello, FastReport!"

この例では、グリーティングメッセージを表示するダイアログを作成します。新しいレポートを作成し、ダイアログを追加するには、ツールバーにある  ボタンをクリックします。



ダイアログ上にLabelControl を置き、[プロパティ]ウインドウでそのText プロパティにメッセージを設定します。



レポートを実行すると、次のようなダイアログが表示されます。[OK] ボタンをクリックしてダイアログを閉じるとレポートが作成されます。[キャンセル] ボタンまたは [X] ボタンをクリックしてダイアログを閉じると、レポートの作業は停止し、デザイナーに戻ります。

## 例 2 :ユーザーにテキスト入力を求める

この例では、ダイアログを作成します。このダイアログでユーザーに対しテキストの入力を求め、その入力した値をレポートに出力します。

新しいレポートを作成し、ダイアログを追加します。ダイアログ上にLabelControl および TextBoxControl コントロールを配置します。



この場合、入力した値は TextBoxControl の Text プロパティに格納されます。この値をレポートに出力するには、レポートタイトルバンドに Text オブジェクトを追加し、そのエディターで次のような式を入力します。

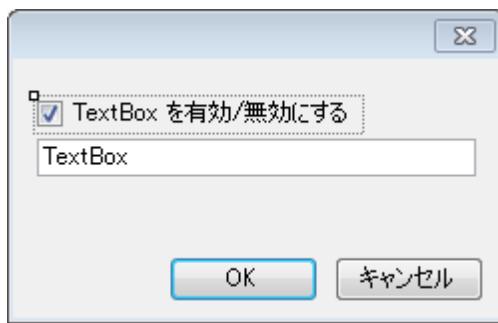
入力した値 {TextBox1.Text}

この TextBox1 は TextBoxControl コントロールの名前です。

## 例 3 :ダイアログ コントロールを処理する

スクリプトおよびコントロールのイベントを使用すると、Visual Studio のようにコントロールを処理することができます。この例では、 CheckBoxControl を使用して TextBoxControl へのアクセスを処理する方法を示します。

新しいレポートを作成し、ダイアログを追加します。図のように、ダイアログ上に CheckBoxControl および TextBoxControl コントロールを配置します。



CheckBoxControl コントロールを選択し、[プロパティ] ウィンドウで ボタンをクリックします。CheckedChanged イベントをダブルクリックします。これはチェックボックスの状態が変わったとき発生するイベントです。FastReport によって、このイベント用に空のハンドラーが作成されます。このハンドラーで以下のコードを記述します。

```
private void CheckBox1_CheckedChanged(object sender, EventArgs e)
{
    TextBox1.Enabled = CheckBox1.Checked;
}
```

レポートを実行すると、チェックボックスのオン/オフによって TextBoxControl を有効または無効にすることができます。

#### 例 4 : レポートオブジェクトを処理する

この例では、カテゴリの一覧および各カテゴリの製品を出力するレポートについて説明します。

**Beverages**  
Soft drinks, coffees, teas, beers, and ales

Product name	Unit price
Chai	¥1,800
Chang	¥1,900
Chartreuse verte	¥1,800
Côte de Blaye	¥26,350
Guaraná Fantástica	¥450
Ipoh Coffee	¥4,600
Lakkalikööri	¥1,800
Laughing Lumberjack Lager	¥1,400
Outback Lager	¥1,500
Rhönbräu Klosterbier	¥775
Sasquatch Ale	¥1,400
Steeleye Stout	¥1,800

**Condiments**  
Sweet and savory sauces, relishes, spreads, and seasonings

Product name	Unit price
Aniseed Syrup	¥1,000

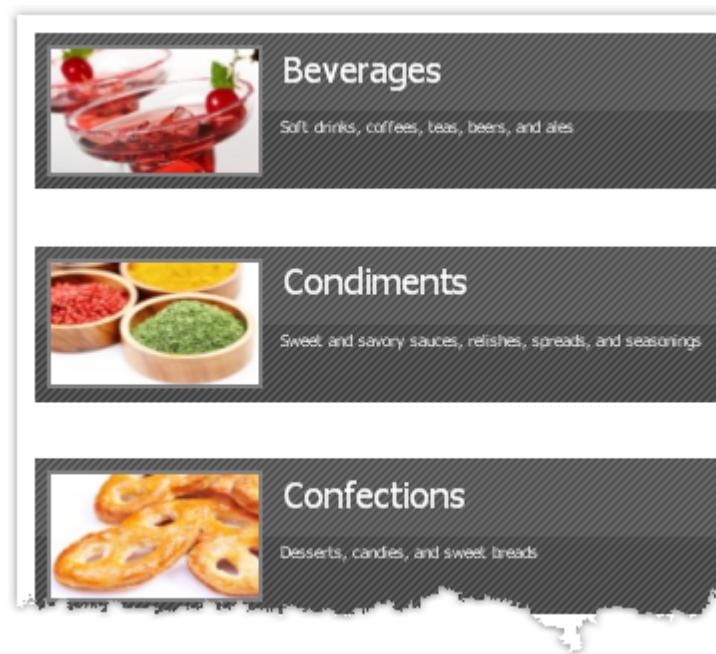
ダイアログを使用して、製品の出力を停止し、カテゴリのみを出力する方法を示します。このため、レポートに1つのダイアログを追加します。



[OK] ボタンをダブルクリックします。FastReport によって Click イベント用に空のイベントハンドラーが追加されます。このハンドラーで以下のコードを記述します。

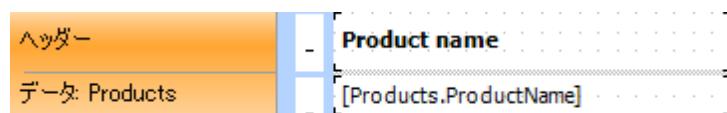
```
private void btnOk_Click(object sender, EventArgs e)
{
    Data2.Visible = CheckBox1.Checked;
}
```

製品の一覧を出力するバンドの表示/非表示を制御します。この例では、このバンドに "Data2" という名前が付けられています。レポートを実行して、チェックボックスのチェックをオフにすると、次のように出力されます。



## 例 5 : シンプルなフィルター

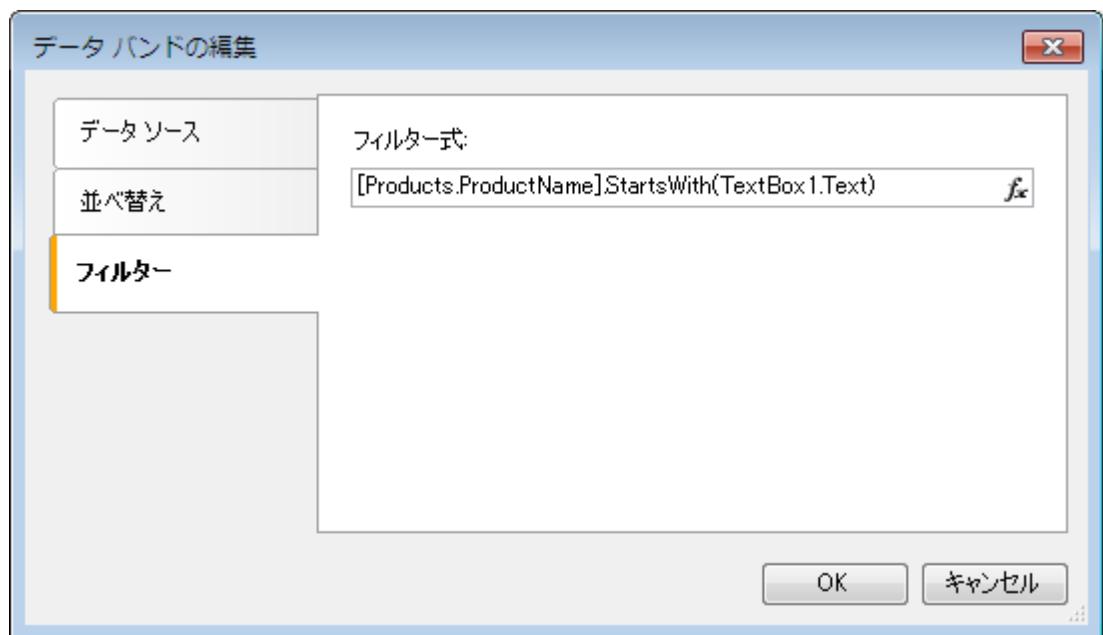
この例では、製品の一覧を出力するレポートについて説明します。



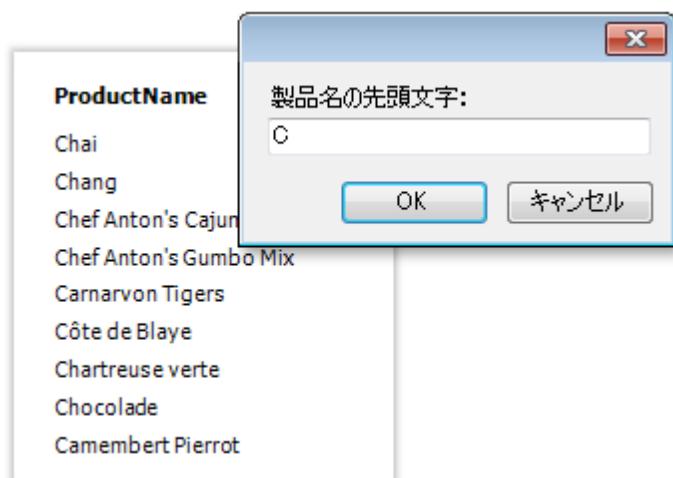
製品名の先頭文字に応じて製品一覧をフィルター処理する方法を示します。ここでは、データの自動フィルタリング機能は使用しません。このため、レポートに1つのダイアログを追加し、LabelControl および TextBoxControl という2つのコントロールを次のように配置および設定します。



データバンドエディターを開き、[フィルター]タブで次のフィルター式を指定します。



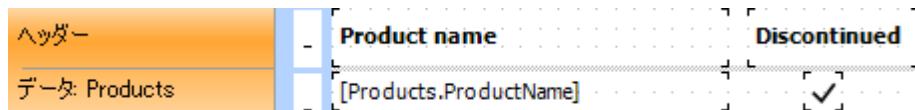
レポートを実行し、適切に動作することを確認してください。



## 例 6 :自動フィルタリング

この例では、Products テーブルから製品の一覧を出力するレポートにフィルターを追加する方法を示します。Products.Discontinued 列に対してデータをフィルター処理します。

レポートデザインは次のようになります。



ツールバーの ボタンをクリックしてレポートにダイアログを追加し、[データ] ウィンドウからダイアログ フォームへ Products.Discontinued 列をドラッグします。



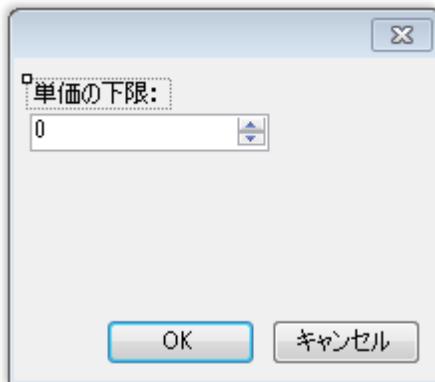
必要な作業はこれだけです。2回のクリックのみで完了しました。コントロールは FastReport によって自動的にデータ列へ接続されます。

レポートを実行し、Discontinued フラグを有効（チェックをオン）にします。その後、[OK] ボタンをクリックすると Discontinued フラグが設定されている製品のみを列挙するレポートが表示されます。

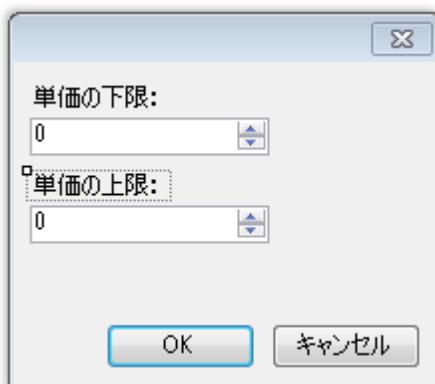
Product name	Discontinued
Alice Mutton	✓
Chef Anton's Gumbo Mix	✓
Guaraná Fantástica	✓
Mishi Kobe Niku	✓
Perth Pasties	✓
Rössle Sauerkraut	✓
Singaporean Hokkien Fried Mee	✓
Thüringer Rostbratwurst	✓

## 例 7 範囲による自動フィルタリング

この例では、前の例のレポートデザインを参考にして、指定した範囲の価格 (UnitPrice)を持つ製品を出力する方法を示します。このため、レポートにダイアログを追加し、[データ] ウィンドウからダイアログフォームへ Products.UnitPrice 列をドラッグします。ドラッグした列のラベルテキストを次のように修正します。



次に、Products.UnitPrice データ列を同様に追加し、そのラベルテキストを次のように修正します。



これで必要な作業は終りです。あとは FastReport によってダイアログフォーム上のコントロールがデータ列へ接続され、各コントロールの FilterOperation プロパティが設定されます。最初のコントロールの FilterOperation プロパティには "GreaterThanOrEqual"、2 番目のコントロールの FilterOperation プロパティには "LessThanOrEqual" が設定されます。

レポートを実行し、値を指定します (たとえば 2000 から 3000)。[OK] ボタンをクリックするとレポートが作成されます。このレポートに指定した範囲の値を持つ製品が表示されます。

Product name	UnitPrice
Chef Anton's Cajun Seasoning	2,200
Chef Anton's Gumbo Mix	2,135
Fløtemysost	2,150
Grandma's Boysenberry Spread	2,500
Gravad lax	2,600
Gustaf's Knäckebröd	2,100
Louisiana Fiery Hot Pepper Sauce	2,105
Maxilaku	2,000
Nord-Ost Matjeshering	2,589
Pâté chinois	2,400
Queso Cabrales	2,100
Sirop d'érable	2,850
Tofu	2,325
Uncle Bob's Organic Dried Pears	3,000

## 例 8 関連付けられたデータ列によるフィルタリング

この例では、関連付けられたデータソースの列を使用してデータのフィルター処理を実行します。

まず、製品リストを出力するシンプルなリストを作成します。カテゴリ名は各製品の横に出力されます。これは次のようないくつかの関係を使用して出力します。

[Products.Categories.CategoryName]

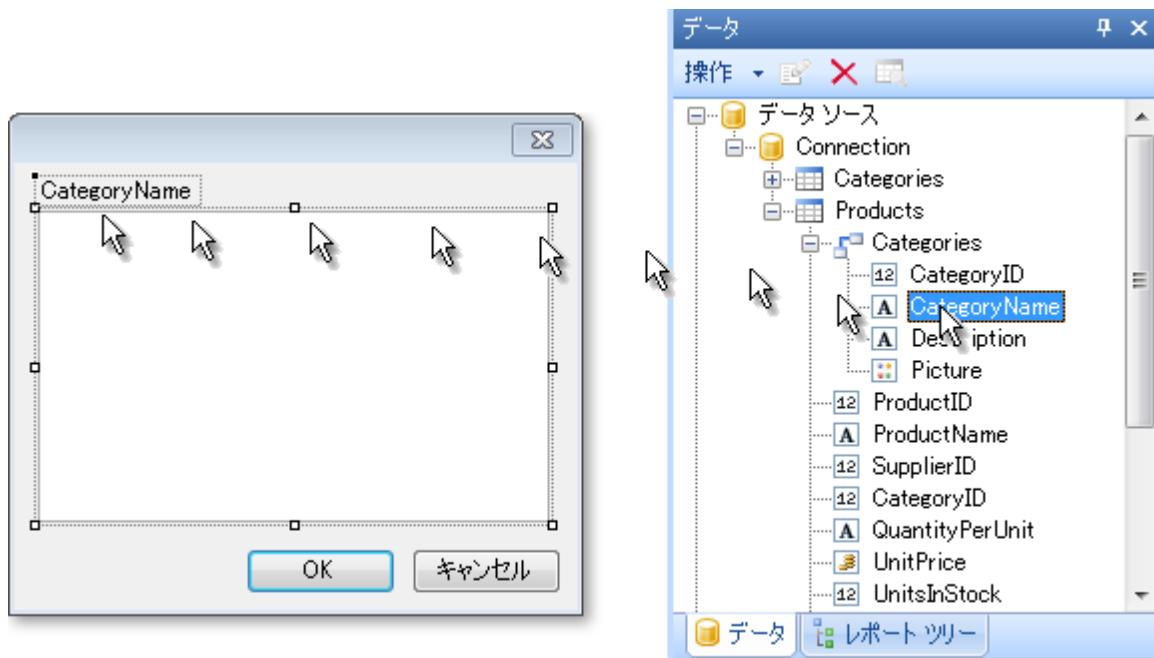
レポートデザインは次のようになります。

レポートを実行すると、次のように出力されます。

## PRODUCT CATALOG

Product name	Category name	Unit price
Alice Mutton	Meat/Poultry	3,900
Aniseed Syrup	Condiments	1,000
Boston Crab Meat	Seafood	1,840
Camembert Pierrot	Dairy Products	3,400
Carnarvon Tigers	Seafood	6,250
Chai	Beverages	1,800
Chang	Beverages	1,900
Chartreuse verte	Beverages	1,800
Chef Anton's Cajun Seasoning	Condiments	2,200
Chef Anton's Gumbo Mix	Condiments	2,135
Chocolate	Confections	1,275

この出力処理に、カテゴリ名によるフィルター処理を追加します。このため、ダイアログを追加し、[データ] ウィンドウからダイアログ フォームへ Products.Categories.CategoryName 列をドラッグします。



フォーム上でコントロールが作成されると、その種類を選択する必要があります。"CheckedListBox (チェックボックス付きリストボックス)" を選択します。レポートを実行すると、次のようなダイアログが表示されます。



いくつかのカテゴリを選択して [OK] ボタンをクリックします。これによりデータがフィルター処理され、次のようなレポートが表示されます。

PRODUCT CATALOG		
Product name	Category name	Unit price
Camembert Pierrot	Dairy Products	3,400
Chai	Beverages	1,800
Chang	Beverages	1,900
Chartreuse verte	Beverages	1,800
Côte de Blaye	Beverages	26,350
Fløtemysost	Dairy Products	2,150
Geitost	Dairy Products	250
Gorgonzola Telino	Dairy Products	1,250
Guaraná Fantástica	Beverages	450
Gudbrandsdalsost	Dairy Products	3,600
Ipoh Coffee	Beverages	4,600
Uht kelikööt	Beverages	1,800

このレポートには、選択したカテゴリの製品のみが表示されます。

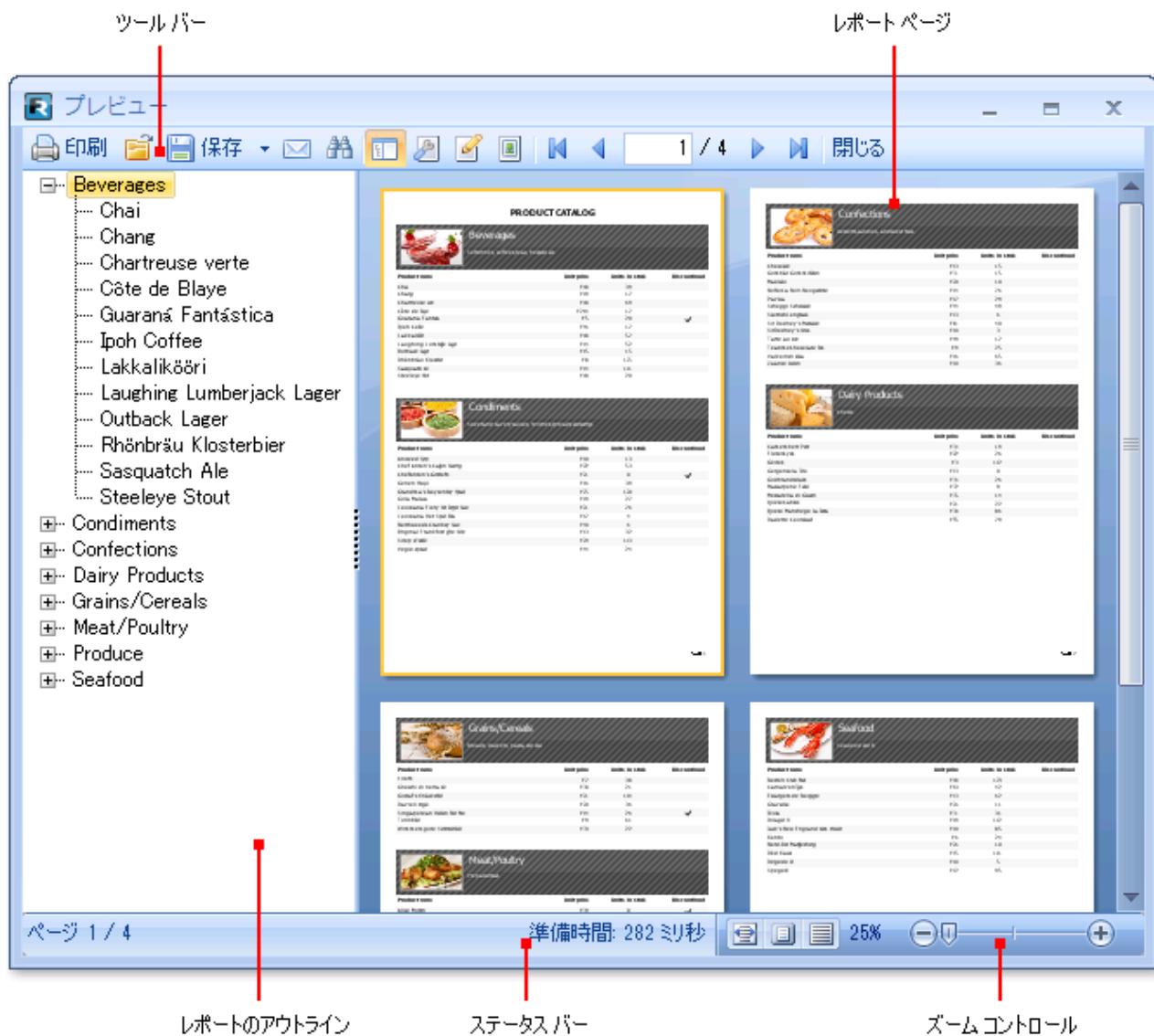




## プレビュー、印刷、エクスポート

## プレビュー、印刷、エクスポート

作成したレポートは、画面上に表示、プリンターで印刷またはサポートされる形式へエクスポートすることができます。これらの作業はすべてプレビュー ウィンドウで行うことができます。



ツールバーには次のようなボタンがあります。

ボタン	説明
	レポートを印刷します。
	FPX 形式のデータ保存 レポートを開きます。

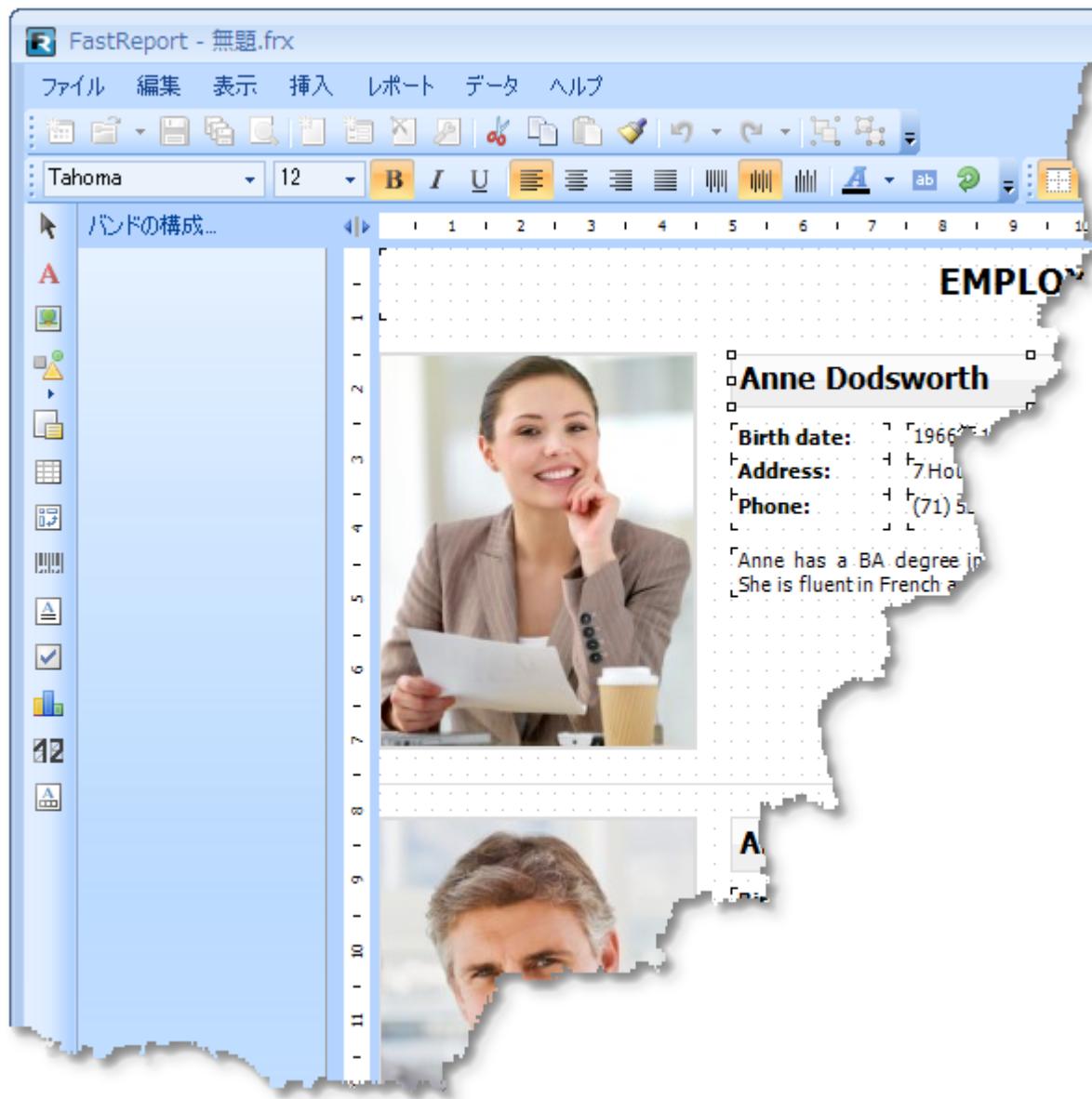
	サポートされる形式の中から1つ指定してレポートを保存します。
	レポートを電子メールで送信します。
	レポート内でテキストを検索します。
	レポートのアウトライン表示または非表示にします。
	ページ設定。
	現在のレポートページを編集します。
	ウォーターマークの設定。
	先頭ページへ移動します。
	前のページへ移動します。
1	指定したページへ移動します。ページ番号を入力してEnterキーを押します。
	次のページへ移動します。
	最終ページへ移動します。

以下のキーボードショートカットを使用できます。

キー	説明
Ctrl+P	レポートを印刷します。
Ctrl+F	テキストを検索します。
矢印キー	プレビューをスクロールします。
PageUp, PageDown	ページアップ/ページダウン。
Home	先頭ページへ移動します。
End	最終ページへ移動します。
Esc	プレビュー ウィンドウを閉じます。

## レポートの編集

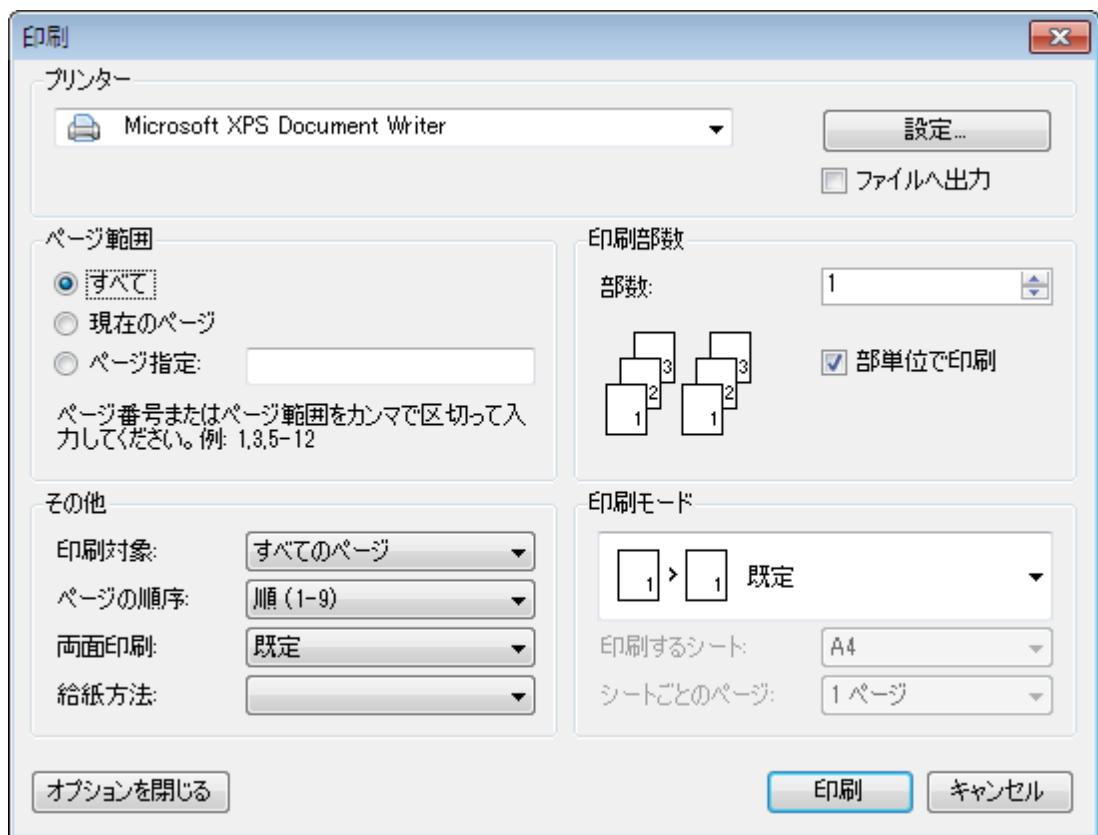
実行したレポートのページを編集する場合は プレビュー ウィンドウの  ボタンをクリックします。これで、現在のページがレポートデザイナーに読み込まれます。ここでページの編集が自由に行えます。



編集が完了したら、デザイナーを閉じます。デザイナーを閉じるときには、変更をレポートページに保存するかどうか尋ねられます。

## レポートの印刷

レポートを印刷するには  ボタンをクリックします (または Ctrl+P キーを押します)。次のような [印刷] ダイアログが表示されます。



このダイアログで利用できる設定を見てみましょう。[詳細オプション/オプションを閉じる] ボタンは、それぞれ詳細設定 (ダイアログ全体) または基本設定のみの表示切り替えに使用します。既定では、ダイアログは基本設定フォームで表示されます。

[プリンター] グループ: プリンターの選択、その設定の変更 (設定 ボタン) およびファイルへの出力の選択が行えます。

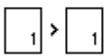
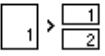
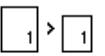
[ページ範囲] グループ: 印刷するページの範囲 (すべて、現在のページまたはページ指定) を選択できます。

[印刷部数] グループ: 印刷部数を設定し、印刷するページの順序 (部単位で印刷) を選択します。

[その他] グループ: 印刷対象のページ (すべて、奇数、偶数)、ページの順序 (順、逆)、両面印刷 (プリンターが両面印刷に対応している場合)、および給紙方法を選択できます。

[印刷モード] グループ: 以下の印刷モードのいずれかを選択できます。

モード	説明
1 > 1	既定

 >  既定	プリンタは、レポートに適応する用紙に印刷します。レポートの1ページが1枚の印刷シートに対応します。
 >  1 大きなページを分割	A3 サイズのレポートをA4 サイズの用紙に印刷する必要がある場合は、このモードを使用します。レポートの1ページを2枚の印刷シートに分割して印刷します。このモードを使用する場合、[印刷するシート]のリストから用紙の種類を選択する必要があります。
 >  1 縮小	A4 サイズのレポートをA3 サイズの用紙に印刷する必要がある場合は、このモードを使用します。1枚の印刷シート上に、1、2、4、または8ページのレポートページを印刷できます。このモードを使用する場合、[印刷するシート]のリストから用紙の種類を選択する必要があります。また、[シートごとのページ]のリストでページ数を指定する必要もあります。

[印刷]ボタンをクリックすると、レポートの印刷が開始します。[ファイルへ出力]フラグを選択した場合は、ファイルの名前を指定するよう要求され、そのレポートは (PRNという拡張子で) そのファイルに保存されます。

## レポートのエクスポート

FastReport では、作成したレポートを別の形式にエクスポートすることができます。現時点では、11種類の形式 (PDF、RTF、HTML、MHT、Excel(XML)、Excel 2007、PowerPoint 2007、OpenOffice Calc、CSV、TXT) へのエクスポートがサポートされています。エクスポートを行うには、プレビュー ウィンドウで [保存] ボタンをクリックし、一覧からエクスポート先を選択します。



## FPX 形式で保存

FPX 形式は FastReport のネイティブな形式です。この形式の利点は次のとおりです。

- 品質を損なうことなくレポートを保存します。既に保存されているレポートファイルを開いた場合は、そのレポートに対してすべての操作（印刷、エクスポート、編集など）が行えます。
- ZIP を使用して圧縮される XML ベースの圧縮形式です。
- レポートファイルは、必要に応じて ZIP 形式をサポートするアーカイバーで解凍（展開）し、テキストエディターで手動で修正することができます。

ただし、この形式の場合はレポートの表示に FastReport.NET が必要です。

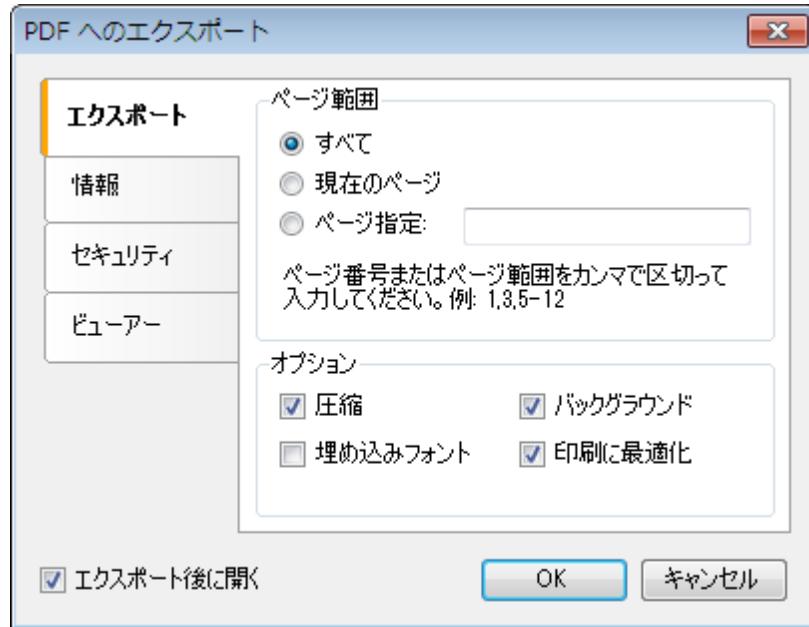
FPX 形式で保存するには、プレビュー ウィンドウのツールバーにある [保存] ボタンをクリックし、ファイルの種類一覧で [データ保存レポート] を選択します。既に保存されているファイルを開く場合は、[開く] ボタンをクリックします。

## Adobe Acrobat (PDF)へのエクスポート

PDF (Portable Document Format) は Adobe Systems によって作成された、プラットフォーム依存の電子ドキュメント形式です。このドキュメントの表示には無料の Acrobat Reader パッケージが使用されます。この形式はかなり柔軟性があります。必要なフォント、ベクタおよびピットマップイメージを含めることができます。表示や印刷に応じたドキュメントの変換および保存も可能です。

エクスポート方法はレイヤー方式です。

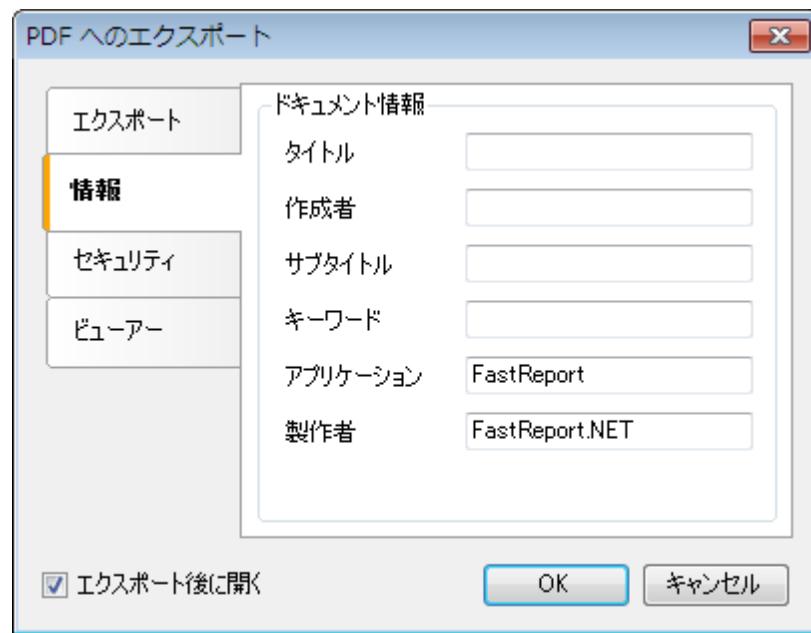
PDF にエクスポートする場合は、出力ファイルのパラメーターを設定できるダイアログ ウィンドウがあります。



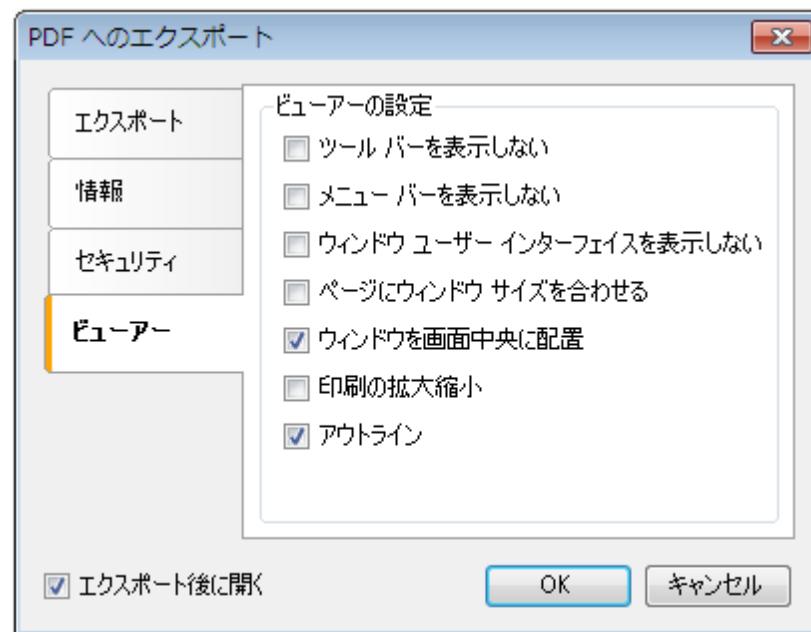
以下のエクスポートパラメーターがあります。

- ・ [圧縮]- 出力ファイルは圧縮されます。ファイルサイズは小さになりますが、エクスポート時間が長くなります。
- ・ [埋め込みフォント]- レポートで使用されるすべてのフォントがPDFファイルに含まれるようになります。これにより、ファイルサイズはかなり大きくなります。
- ・ [バックグラウンド]- ページのウォーターマークはイメージとしてエクスポートされます。これにより、ファイルサイズはかなり大きくなります。
- ・ [印刷に最適化]- すべてのグラフィックオブジェクト(図やグラフなど)を印刷向けの高解像度で出力します。

[情報]タブでは、ドキュメントに関する情報を各フィールドに入力することができます。



[ビューアー]タブでは、Adobe Acrobat ドキュメントビューアーに関するオプションを設定できます。

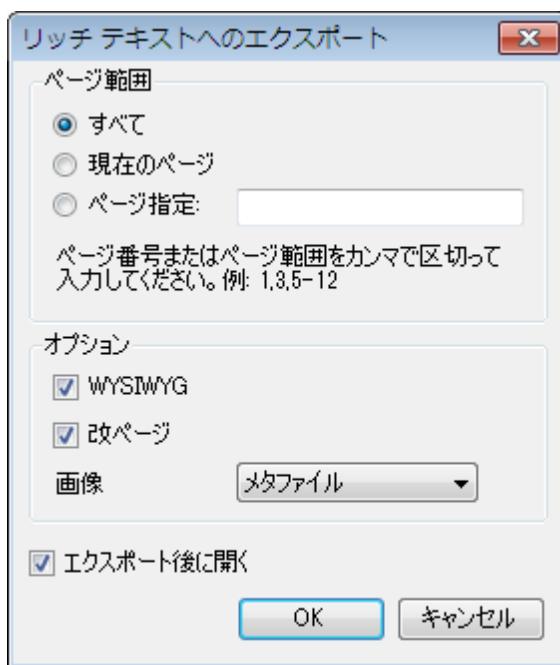


## Word (RTF)へのエクスポート

RTF (リッチテキスト形式) は、テキスト情報の交換に使用される標準形式で、Microsoft によって開発されました。現在、RTF ドキュメントは多くの新しいテキストエディターおよびオペレーティングシステムとの互換性があります。

### エクスポート方法 表形式

RTF にエクスポートする場合は、出力ファイルのパラメーターを設定できるダイアログウィンドウがあります。



以下のエクスポートパラメーターがあります。

- 「WYSIWYG」- エクスポートの結果は、可能な限りレポートと近い状態になります。このオプションを無効にすると、結果ファイルで行数および列数が削減されます。
- 「改ページ」- RTF ファイルで改ページを有効にします。
- 「画像」- RTF ファイルにおける画像の形式を選択します。「メタファイル」形式は、MSChartObject およびShapeObject などのレポートオブジェクトの表示に最も適しています。

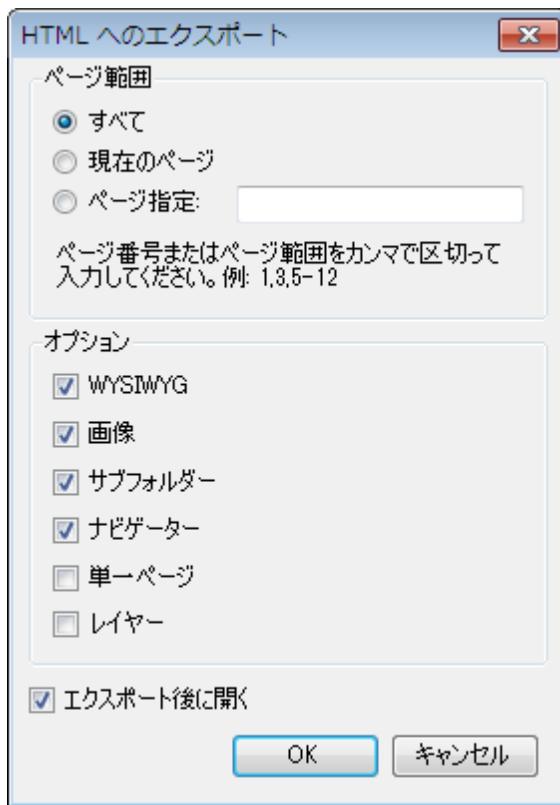
エクスポートの結果ファイルの外観やサイズは、レポートテンプレートによって異なります（[レポートデザインにおける推奨事項](#)を参照）。

## HTML へのエクスポート

HTML (HyperText Markup Language) は Web ページ向けに使用される主要なマークアップ言語です。これはドキュメントに含まれる情報の構造をテキストベースで記述する方法を提供します。この方法で、特定のテキストをリンク、見出し、段落、リストとして表示したり、そのテキストに对话型フォーム、埋め込みイメージおよびその他のオブジェクトを加えることができます。

### エクスポート方法 表形式

HTML へエクスポートする場合、パラメーターを設定するためのダイアログ ウィンドウが提供されます。



以下のエクスポートパラメーターがあります。

- 「WYSIWYG」- エクスポートの結果は、可能な限りレポートと近い状態になります。
- 「画像」- 画像のエクスポートを有効にします。
- 「サブフォルダー」- 当該ファイル以外の追加ファイルはすべて「files」という別フォルダーに保存されます。
- 「ナビゲーター」- ページの移動用に特別なナビゲーターを作成します。
- 「単一ページ」- すべてのページを1つのファイルに保存します。

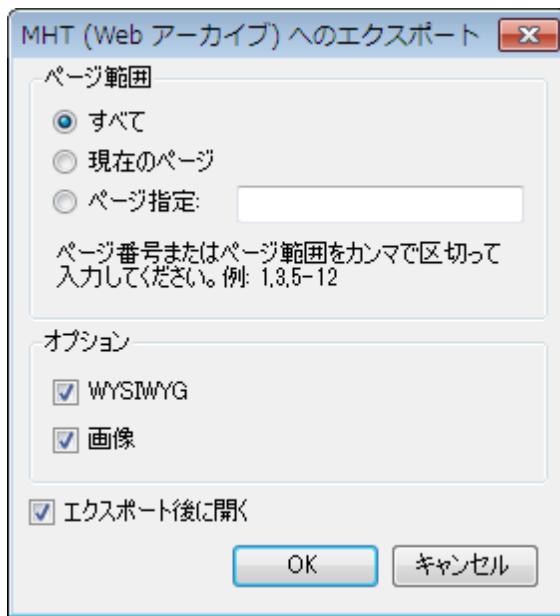
エクスポートの結果ファイルの外観やサイズは、レポートテンプレートによって異なります（[レポートデザインにおける推奨事項](#)を参照）。

## MHT (Web アーカイブ)へのエクスポート

MHT (MIME HTML の省略) は Web ページのアーカイブ形式です。これを使用すると、通常は外部リンク(イメージ、Flash アニメーション、Java アプレット、音声ファイルなど)で表されるリソースを HTML コードと共に 1 つのファイルにまとめることが出来ます。MHT ファイルの内容は、HTML 電子メールメッセージのようエンコードされます。これには MIME タイプ "multipart/related" が使用されます。

### エクスポート方法 表形式

MHT へエクスポートする場合、パラメーターを設定するためのダイアログウィンドウが提供されます。



以下のエクスポートパラメーターがあります。

- 「WYSIWYG」- エクスポートの結果は、可能な限りレポートと近い状態になります。
- 「画像」- 画像のエクスポートを有効にします。

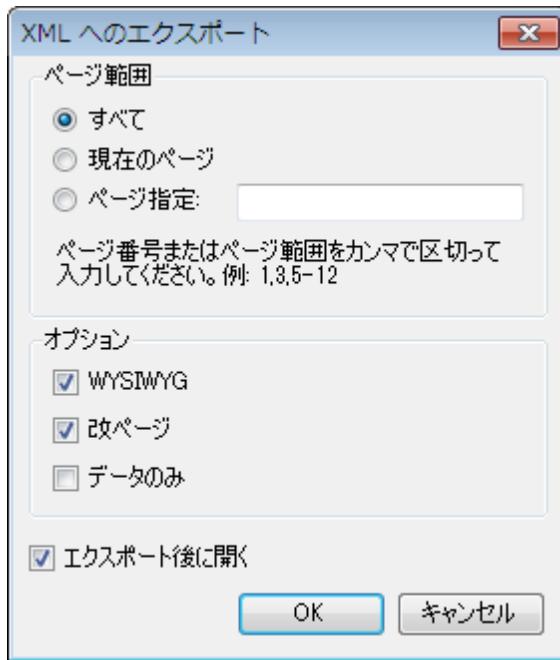
エクスポートの結果ファイルの外観やサイズは、レポートテンプレートによって異なります ([レポートデザインにおける推奨事項](#)を参照)。

## Excel (XML)へのエクスポート

Excel は、電子ワークシートを使って作業するためのアプリケーションです。これは Microsoft Office に含まれています。

### エクスポート方法 表形式

XML へエクスポートする場合は、出力ファイルのパラメーターを設定できるダイアログウィンドウがあります。



以下のエクスポートパラメーターがあります。

- 【WYSIWYG】- エクスポートの結果は、可能な限りレポートと近い状態になります。このオプションを無効にすると、結果ファイルで行数および列数が削減されます。
- 【改ページ】- 結果ファイルで改ページを有効にします。

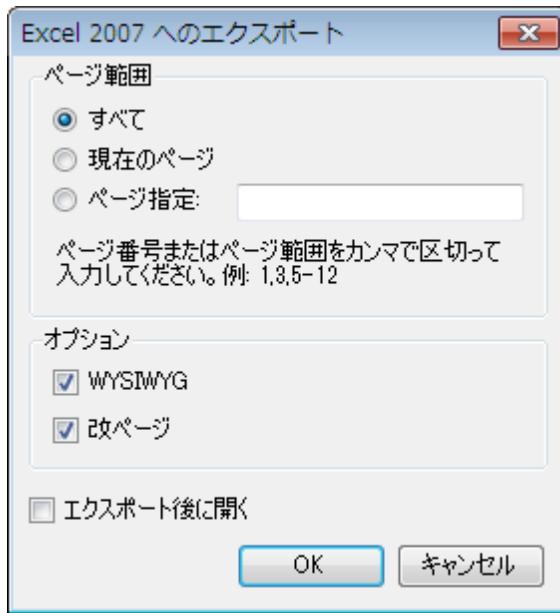
エクスポートの結果ファイルの外観やサイズは、レポートテンプレートによって異なります（[レポートデザインにおける推奨事項](#)を参照）。

## Excel 2007 へのエクスポート

Excel 2007 は、電子ワークシートを使って作業するためのアプリケーションです。これは Microsoft Office 2007 に含まれています。

### エクスポート方法 表形式

Excel にエクスポートする場合は、出力ファイルのパラメーターを設定できるダイアログ ウィンドウがあります。



以下のエクスポートパラメーターがあります。

- 「WYSIWYG」- エクスポートの結果は、可能な限りレポートと近い状態になります。このオプションを無効にすると、結果ファイルで行数および列数が削減されます。
- 「改ページ」- 結果ファイルで改ページを有効にします。

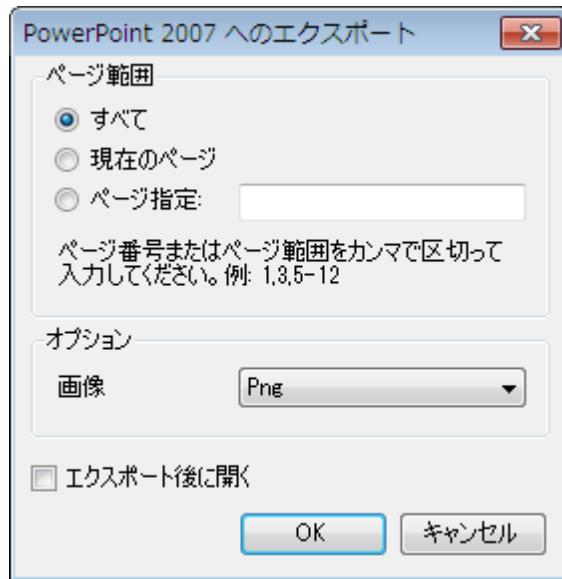
エクスポートの結果ファイルの外観やサイズは、レポートテンプレートによって異なります（[レポートデザインにおける推奨事項を参照](#)）。

## PowerPoint 2007 へのエクスポート

PowerPoint 2007 は、電子プレゼンテーションを使って作業するためのアプリケーションです。これは Microsoft Office 2007 に含まれています。

エクスポート方法はレイヤー方式です。

PowerPoint にエクスポートする場合は、出力ファイルのパラメーターを設定できるダイアログ ウィンドウが提供されます。



以下のエクスポートパラメーターがあります。

- ・ [画像]- 結果ファイルにおける画像の形式を選択します。

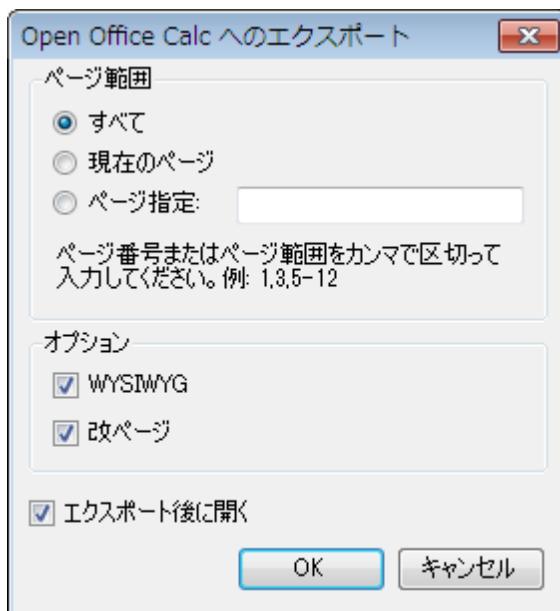
## OpenOffice Calc へのエクスポート

OpenDocument Format (ODF、OASIS Open Document Format for Office Application) は、OASIS によって設計された形式で、OpenOffice で使用される XML に基づいています。

FastReport は表 (ods ファイル)へのエクスポートをサポートします。そのファイルはOpenOffice で開くことができます。

### エクスポート方法 表形式

OpenOffice Calc にエクスポートする場合は、出力ファイルのパラメーターを設定できるダイアログウインドウが提供されます。



以下のエクスポートパラメーターがあります。

- 「WYSIWYG」- エクスポートの結果は、可能な限りレポートと近い状態になります。このオプションを無効にすると、結果ファイルで行数および列数が削減されます。
- 「改ページ」- 結果ファイルで改ページを有効にします。

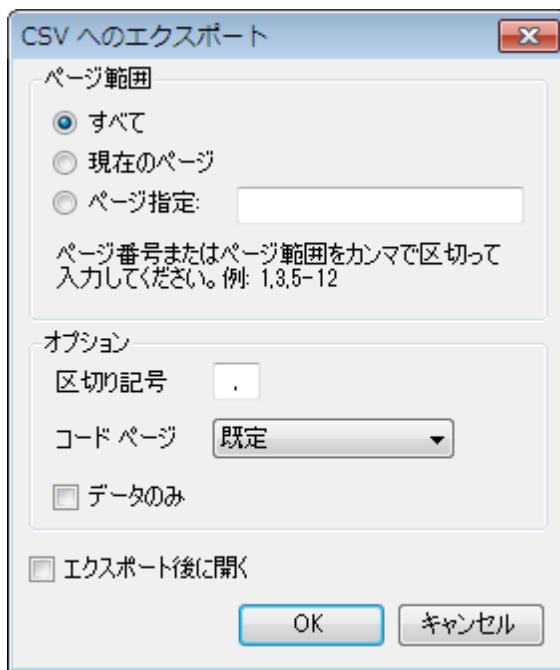
エクスポートの結果ファイルの外観やサイズは、レポートテンプレートによって異なります（[レポートデザインにおける推奨事項](#)を参照）。

## CSVへのエクスポート

CSVファイルは、リスト形式の表で構造化されたデータのデジタル保存に使用されます。CSVファイルの各行は表における1行に対応します。行内でフィールドはカンマで区切られます。各フィールドは表における1つの列に対応します。CSVファイルは多くの場合、2つの異なるコンピューター プログラム間（データベース プログラムとスプレッドシート プログラム間など）で表形式のデータを交換するために使用されます。

### エクスポート方法 表形式

CSVにエクスポートする場合は、出力ファイルのパラメーターを設定できるダイアログ ウィンドウが提供されます。



以下のエクスポートパラメーターがあります。

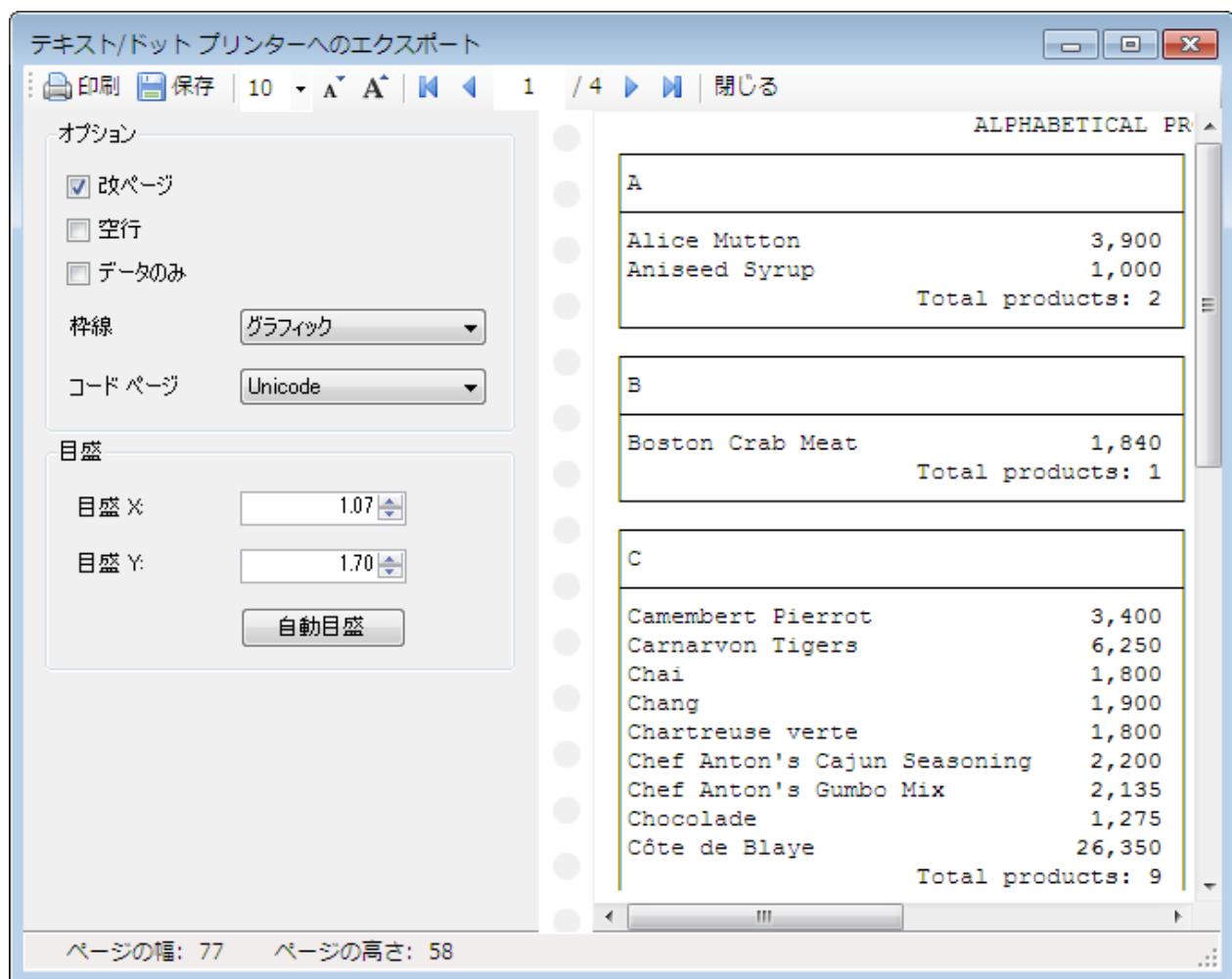
- 【区切り記号】- フィールドの区切りの記号です。
- 【コードページ】- 結果ファイルのテキストをエンコードするために使用するコードページです。"既定" の場合は Windows の既定のコードページが使用されます。Excel のバージョンによっては Unicode コードページがサポートされないので注意してください。
- 【データのみ】- このオプションを有効にすると データバンドに置かれているオブジェクトのみをエクスポートします。

## TXTへのエクスポート

TXTは、あらゆるテキストエディターで開くことができる、またはドットマトリックスプリンターに印刷できる標準テキストファイルです。

### エクスポート方法 表形式

TXTにエクスポートする場合は、出力ファイルのパラメーターを設定できるダイアログウィンドウが提供されます。



以下のエクスポートパラメーターがあります。

- 【改ページ】- 結果ファイルで改ページを有効にします。
- 【空行】- 結果ファイルで空行を有効にします。
- 【データのみ】- このオプションを有効にすると、データバンドに置かれているオブジェクトのみをエクスポートします。
- 【枠線】- オブジェクトの罫線の種類です。罫線をエクスポートしない場合は"なし"を選択します。
- 【コードページ】- 結果ファイルのテキストをエンコードするために使用するコードページです。
- 【目盛 X】- 横の目盛りです。
- 【目盛 Y】- 縦の目盛りです。
- 【自動目盛】- 目盛 X および目盛 Y を自動的に計算してデータが次落しないようにします。

## イメージへのエクスポート

FastReport では、情報を以下のグラフィック形式にエクスポートすることができます。

- BMP
- PNG
- JPG

- GIF
- TIFF
- Windows メタファイル (EMF、WMF )

#### エクスポート方法 描画

イメージ ファイルへエクスポートする場合、パラメーターを設定するためのダイアログ ウィンドウが提供されます。



以下のエクスポートパラメーターがあります。

- [解像度]- グラフィカルイメージの解像度です。表示には96dpi、印刷には300dpi を使用してください。TIFF 形式にエクスポートする場合は、縦と横の解像度の値を別々に設定することができます。
- [JPEG 画質]- JPG ファイルの圧縮レベルです。このオプションはJPEG 形式にエクスポートする場合にのみ使用されます。
- [ページごとにファイルを分ける]- このオプションを有効にすると、レポートはページ単位で別々のファイルへエクスポートされます。エクスポート先のファイル名には、選択したページに基づく番号が付加されます。
- [マルチフレーム TIFF]- このオプションはマルチフレームのTIFF ファイルを作成します。このオプションはTIFF 形式にエクスポートする場合にのみ使用されます。
- [モノクロ TIFF]- このオプションはモノクロのTIFF ファイルを作成します。このオプションはTIFF 形式にエクスポートする場合にのみ使用されます。

複数のページを1 つのファイルにエクスポートする場合は ([ページごとにファイルを分ける]オプションが無効の場合)、大量のCPU/メモリリソース

ースが使用されます。

## レポートデザインにおける推奨事項

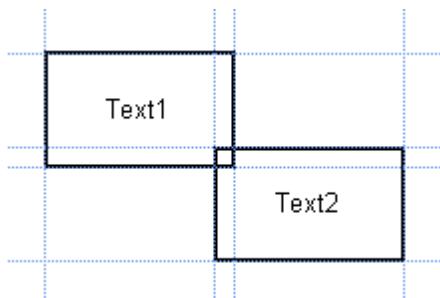
このセクションでは、別のデータ形式へのエクスポート向けにレポートをデザインする場合の特別な要件について説明します。

FastReport では、レポートの作成時にオブジェクトを操作する方法が数多くあります。この利点を活かし、あらゆるレポートの迅速な開発、さらにその印刷を行うことができます。印刷されるドキュメントは画面上の表示と同じになります。これが FastReport レポートジェネレーターの主な使用目的といえます。

このように開発が自由である一方、FastReport ドキュメントを別のデータ形式へエクスポートする際には複雑になるもうマイナス面があります。エクスポート先の形式によっては情報の表し方に独自の制限や要件があり、予想以上に複雑になることもあります。HTML、XLS または RTF などの形式は、テーブルデータプレゼンテーションを使用します。これらの形式では、テーブルのマーキング時にレイヤーにおけるセルの交差や配置を行うことができません。

一般に、エクスポートフィルターはこれらの要件を考慮に入れます。これは、オブジェクトの交差を考慮し、それらを最適に配置する特別なアルゴリズムによって実行されます。オブジェクトが交差しているとき、エクスポート結果として出力されるテーブルには新たな列と行が出現します。これはエクスポート結果と元のレポートの表示を極力同じにするために必要なことです。レポートのデザインで多くのオブジェクトを交差させると、エクスポート結果のテーブルで列および行が増加し、ファイルのサイズや複雑さに影響します。

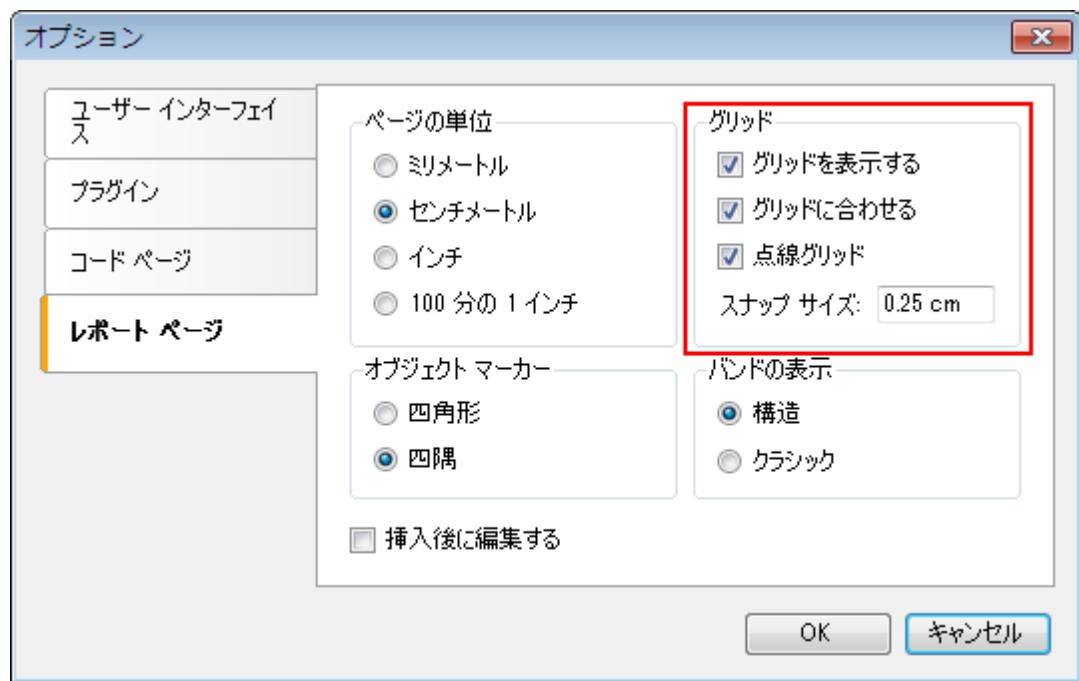
元のレポートが的確にデザインされているかどうかによって、エクスポートの品質は大きく異なります。次の例を見てみましょう。



同じレポート上で、互いに一部が交差する2つのオブジェクトがあります。レポート作成におけるレコード数(行数)は150でした。これを RTF 形式へエクスポートする場合は450行が作成されます(オブジェクト1につき150行、交差に対して150行)。オブジェクトの交差を解消すれば、エクスポート結果の表は300行のみになります。大きなレポート、多数のオブジェクトであれば、この差が極めて顕著になります。もちろん、これは出力ファイルのサイズにも影響します。

レポートで表を作成する場合は、隣接するセルの罫線に注意してください。セルは交差せず、レイヤーに配置されていることが重要です。エクスポートフィルターアルゴリズムによってセルは切り離されますが、予期しない結果が生じます(希望どおりには表示されません)。オブジェクトは、縦および横に並べて置かれるように配置してください。ガイドラインはこれを実行するために役立ちます。

グリッドの配置はセルが重なる場合にも役立ちます。デザイナーのオプションでグリッドの配置を有効にします。グリッドの配置を簡略化するためには、グリッドのピッチを広げることができます。グリッドのピッチと配置の設定は、[表示]メニューの[オプション]を選択して開く[オプション]ダイアログにあります。



テキストに枠線を付ける場合は、線、四角形オブジェクトなどの単独グラフィックオブジェクトではなくテキストオブジェクトの墨線の使用をお勧めします。透過のテキストオブジェクトの下で背景オブジェクトを使用しないようにしてください。

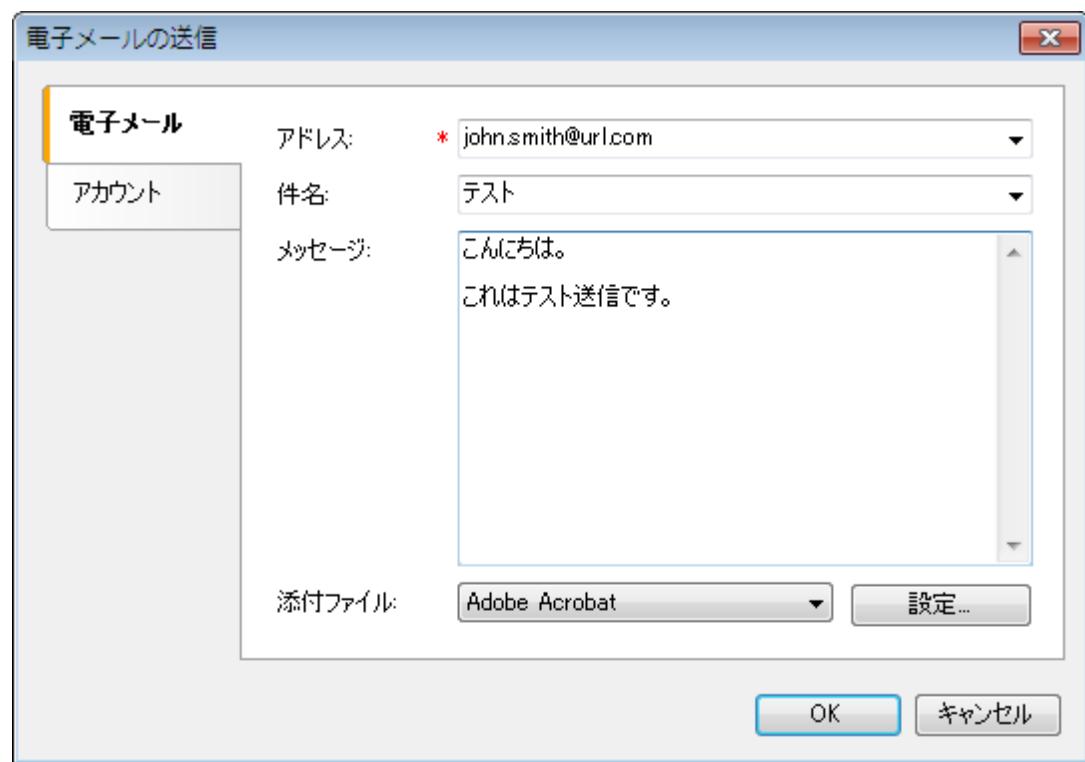
上記のシンプルな規則を適用すれば、表ベースの形式へのエクスポート後も最適に表示されるレポートを作成することができます。

## レポートを電子メールで送信する

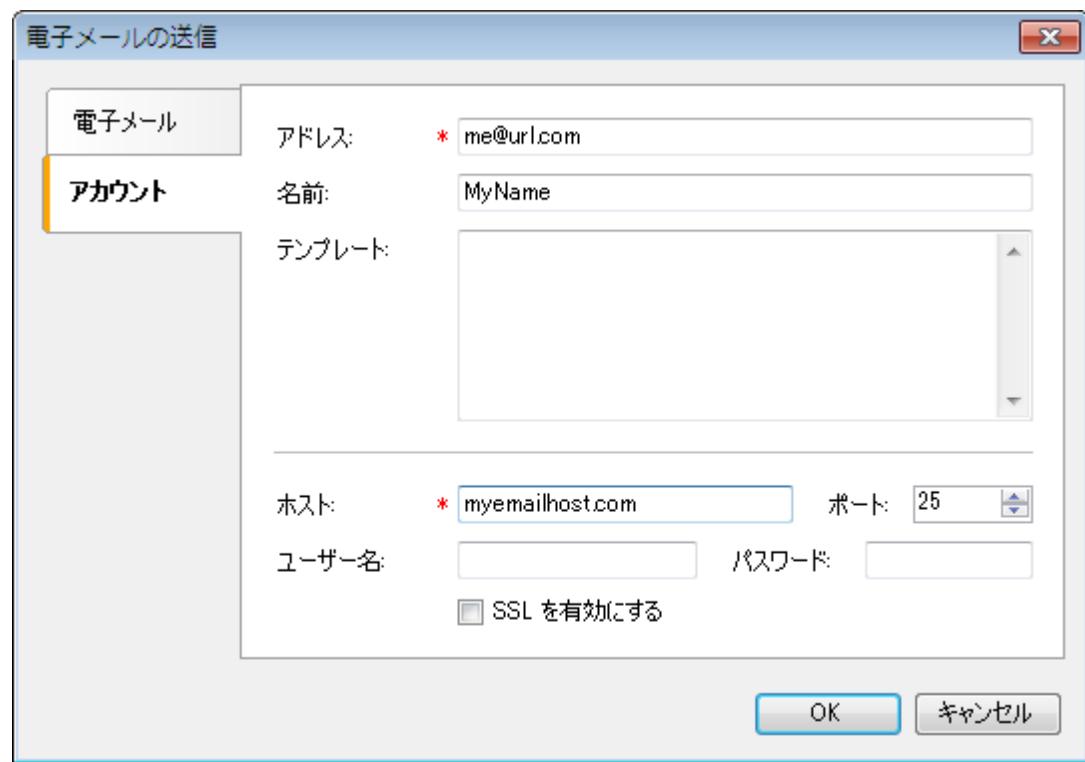
FastReport では、実行したレポートを電子メールで送信することができます。これは次の 2 つのモードで実行することができます。

モード	説明
SMTP	これは既定のモードです。電子メールを送信するためにプログラムを追加する必要はありません。
MAPI	このモードはプログラムで設定します。これを Config.EmailSettings.UseMAPI プロパティに True を設定するか、あるいは EnvironmentSettings コンポーネントを使用する場合はその EnvironmentSettings.EmailSettings.UseMAPI プロパティに True を設定します。  電子メールを送信するため、FastReport は Outlook Express など既定の電子メールクライアントを使用します。このクライアントでは MAPI プロトコルがサポートされている必要があります。

電子メールを送信するためには、受信者の電子メールアドレスを指定する必要があります。件名と電子メール本文も指定する必要がありますが、これは任意です。[電子メールの送信] ダイアログの下部 (添付ファイル) では、メッセージに添付するレポートの形式を選択します。



SMTP モードを使用する場合は、アカウントを設定する必要があります。この設定は一度だけ行う必要があります。一度設定すれば、FastReport によってそのパラメーターが構成ファイルに保存されます。このパラメーターは [アカウント] タブにあります。必須入力フィールドには赤色のアスタリスクが付いています。



ただし、ホストサーバーで認証が必要な場合は、ユーザー名とパスワードをそれぞれ [ユーザー名] および [パスワード] フィールドに入力する必要があります。