中嶋 祥子

株式会社ミガロ.

RAD事業部 技術支援課

Delphi/400:グラフ活用リファレンス

Delphi/400 には、データをグラフ化するためのコンポーネント「TDBChart」がある。
その設定と活用について、基本的な操作をベースに説明する。

●はじめに
●TDBChartの基本操作
●TDBChartの応用
●TChart
●補足
●注意点
●まとめ

略歴
1968 年 02 月 23 日生
1990 年奈良女子大学家政学部卒
2002 年株式会社ミガロ . 入社
2002 年 11 月 RAD 事業部配属

現在の仕事内容
お客様からの Delphi/400 に関する技術的な質問や問い合わせに対応している。また、メルマガ「Migaro News」やホームページの Tips など、開発に役立つ情報も担当。

1.はじめに

アプリケーションでは多様なデータを処理するが、その際、処理や操作だけでなく、得られるデータをどう活用するかも重要なポイントとなってくる。照会画面などでデータを並べただけでは、データの特徴を直感的につかむことも難しいし、情報を活かしきることもできない。

それに対してグラフであれば、一目で傾向や流れ、パターンを把握することが可能である。

Delphi/400 には、データをグラフ化するためのコンポーネントとして「TDBChart」がある。よく使用される TDBGrid と同様に、簡単な設定で利用できる。その方法について、基本的な操作をベースにいくつかのポイントを説明する。また、「TChart」についても併せて紹介する。

2.TDBChartの基本操作

実際の操作手順を説明していこう。

フォームに TDBChart を配置する。なにも設定していない状態が図 1 である。【図 1】

TDBChartの設定

TDBChart をダブルクリックすると、設定画面が表示される。ここで、実際のグラフである TSeries を追加・設定していく。

Add ボタンより表示されたダイアログから種類を選択すると、TSeries が追加され、図 2 の状態になる。ここでは、棒グラフである「Bar」を選択する。【図 2】

追加された Series1 をクリックすると、TSeries の設定画面が表示される。【図 3】

データとの関連付けは、図 4 のように、Data Source タブ内で行う。【図 4】

プルダウンリストより DataSet を選択し、次に DataSet：で使用するデータセットを指定する。このプルダウンリストには、そのフォームから参照できるデータセットが表示される。

次に、X 軸のラベル、X 軸のフィールド、Y 軸のフィールドを選択する。

X 軸のフィールドは横軸の値になるフィールドで、それとは別のフィールドをラベルに指定できる。例えば、X 軸のフィールドに店舗コードを指定すると、店舗コード順にグラフが作成される。そのときに、ラベルを店舗コードでなく店舗名としておくと、店舗コード順に並び、ラベルを店舗名とすることができる。

今回のデータでは、X 軸・ラベルとも年月のフィールドとする。最後に、設定が終了すれば、設定画面を閉じる。

以上が、データをグラフとして表示する設定である。

TDBChartの例

実際に TDBChart に表示するには、データセットを Open する。

グラフの各座標が、データから（年月、売上金額）＝（201001,11834588）、（201002,18078025）…と設定され、それに応じて描画や設定が行われる。

図1

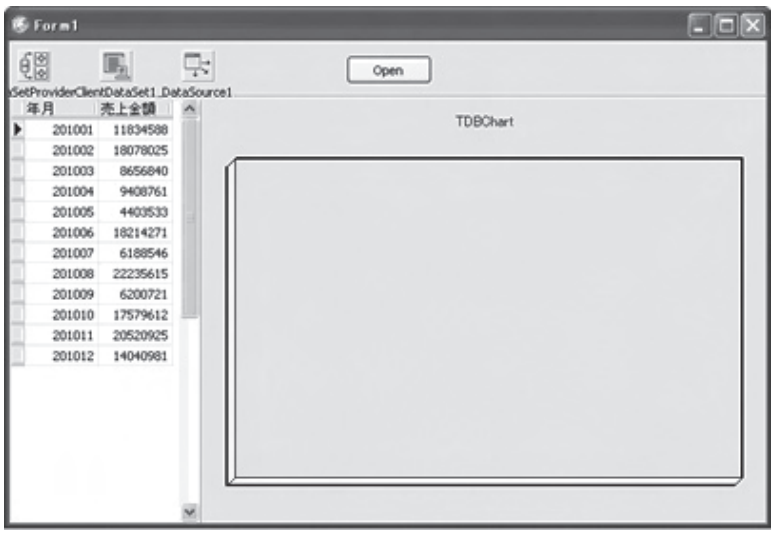
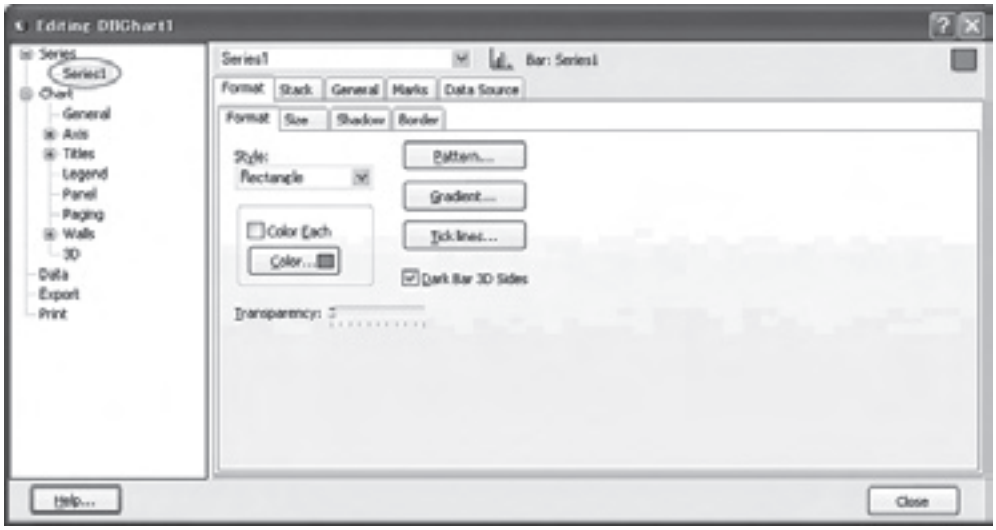

Form1
Open
TDBChart
年月
売上金額
201001 11834588
201002 18078025
201003 8656840
201004 9408761
201005 4403533
201006 18214271
201007 6188546
201008 22235615
201009 6200721
201010 17579612
201011 20520925
201012 14040981

図2


種類を選択
追加されたSeries

図3


Editing DBChart1
Series1
Chart
General
Axis
Titles
Legend
Panel
Paging
Walls
3D
Data
Export
Print
Format
Stack
General
Marks
Data Source
Shadow
Border
Style:
Rectangle
Pattern...
Gradient...
Tick lines...
Color Each
Color...
Dark Bar 3D Sides
Transparency:
Help...
Close

また、設計画面でデータセットをOpenしても、図5のようにグラフが表示されるので、確認しながら操作することができ、非常に便利である。【図5】

グラフの外観

このように、簡単な設定でコーディングなしでもデータをグラフ化できる。さらに外観を変える方法もいくつか説明する。設定と反映結果は、図6のようになる。【図6】

- **タイトル**
  Chartの Titlesから設定する。
- **グラフの色**
  Seriesの Colorを変更する。
- **マーカー**
  マーカーとは、グラフから伸びた軸の値を示すボックスである。表示値を、X軸値かY軸値かを指定できる。基本操作としてまず、表示/非表示の切り替えを行う。
- **凡例**
  グラフの右横にある、凡例の表示/非表示の切り替えを行う。

X軸のラベル

ラベルの操作はChartの Axis（軸）に対して行うが、Axisは6種類ある。X軸はここでは下の軸になるため、BottomAxisである。

現在は図6の通り、X軸のラベルである年月が2か月おきに表示されている。これはラベルの間隔が、描画されるラベル幅とTDBChartの幅によって自動的に調整されるためである。【図6】

TDBChartの幅を大きくしたり、ラベルのフォントサイズを小さくするなどで対応できることもあるが、設定によりすべてのラベルを表示するように変更できる。BottomAxisの Show all labelsにチェックを入れると、すべてのラベルを表示される。

だが、今回はラベルの文字列長がその幅よりも長いため、すべて表示されてしまうと、図7のように文字が重なってしまう。【図7】

すべてのラベルが表示され、文字も重ならないようにするには、Alternateにチェックを入れてラベルをずらして2段に表示する方法か（図8）、またはラベルの Angleで回転角度を270度に指定することで、縦書きに見せる方法がある（図9）。これも設定画面のみで行える。【図8】【図9】

このように、簡単な基本設定と数か所の設定変更だけで、見栄えのよいグラフが作成できる。

3. TDBChartの応用

TDBChartでは、複数のグラフを表示することもできる。その例として、次に支店ごとの月別売上データを用意し、設定を行う方法を説明していこう。

今回のグラフは、前述のように売上年月別売上金額を表示し、支店ごとに比較できるようにする。また基本操作と同様に、グラフは売上年月をX軸に、売上金額をY軸とする。

まず支店の数のTSeriesを追加し、基本操作と同様にData Sourceでデータセットを設定する。TSeriesごとに、対応する支店別の売上データを関連付ける。

データ以外に、X軸のラベル表示なども基本操作で述べた方法で変更した状態が図10である。【図10】

マーカー

基本操作ではマーカーは非表示にした。今回はマーカーの操作を説明するために、表示したままにする。現在、マーカーにはX軸の値が表示されているが、これをY軸の売上金額に変更する。

Styleの選択値が、デフォルトのX valueだと、X軸の値が表示される。これをValueとすると、図11のようにY軸の値つまり売上金額に変更され、またデフォルトでカンマ区切りの編集表示がなされている。【図11】

X軸のラベル

次にX軸のラベルを見ると、データ値そのままの“201001”“201002”…となっている。

ラベルはフォーマット設定ができるので、見やすくするために“0000年00月”と年月のフォーマットを指定すると、図12のように反映される。【図12】

凡例

ここで凡例を見ると、TSeries名が“Series1”“Series2”“Series3”と作成した時の名前のままで表示されている。

変更は各TSeriesに対して行うので、設定画面でSeriesを選び、一覧からSeries1を選択後、Titleボタンをクリックし、凡例に表示したい文字列を入力する。

それぞれのグラフが各支店に対応しているので、支店名に変更しておく。

設定と反映結果は図13となる。【図13】

コードによる記述

TDBChartや TSeriesに対する操作を、設定画面から行う方法で説明してきたが、実はすべてプロパティの値として設定されている。そのため、通常のコンポーネントと同様に、オブジェクトインスペクタから設定することもできる。

図14は、TDBChartの BottomAxisのプロパティの一部である。【図14】

通常のコンポーネントのプロパティと同じであるため、設計画面だけでなく、コードによってプロパティを制御することが可能である。

マーカー切替のコード例

実際にどのようなコードになるのかを、マーカーの表示を切り替えるコードを例として説明する。

プログラムの動作として、各TSeriesに対応するチェックボックスを用意し、ここにチェックが入っていればそのTSeriesのマーカーを表示することとする。

チェックボックスの操作時に表示と非表示を切り替えるため、チェックボックスの OnClickイベントでソース1のように記述する。【ソース1】

これは、マーカーの Visibleプロパティに、チェックボックスがチェックされればTrue、されなければFalseをセットしている。

TSeriesのマーカーは、初期設定では表示状態になっている。起動時に非表示としたい場合、前述したように設計画面からでも行えるが、今回はFormの OnCreateイベントで行うことにする。また、このとき個々の TSeriesを指定する方法とは別に、TDBChartにある TSeries

図4

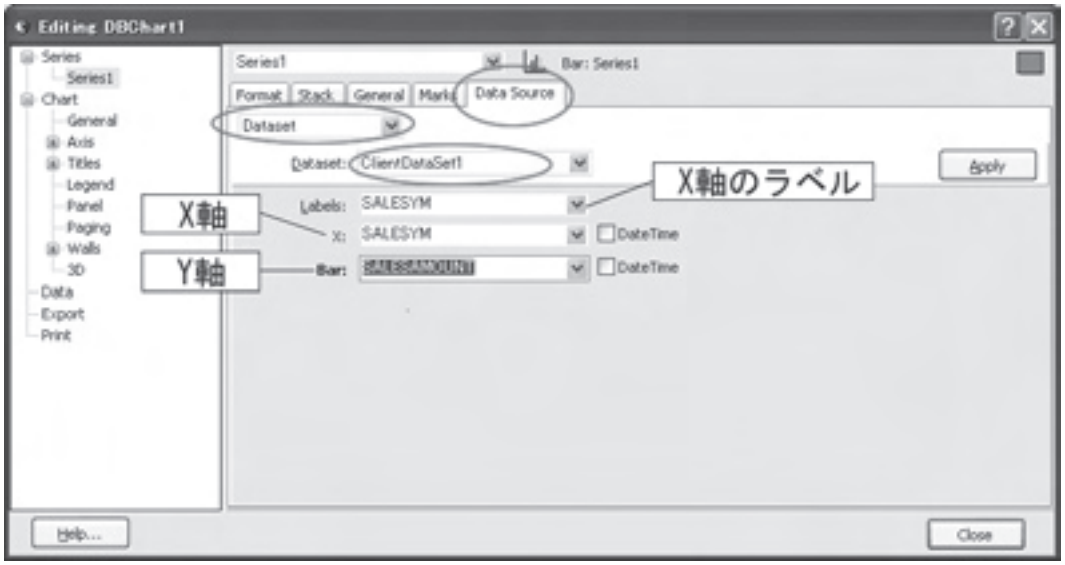
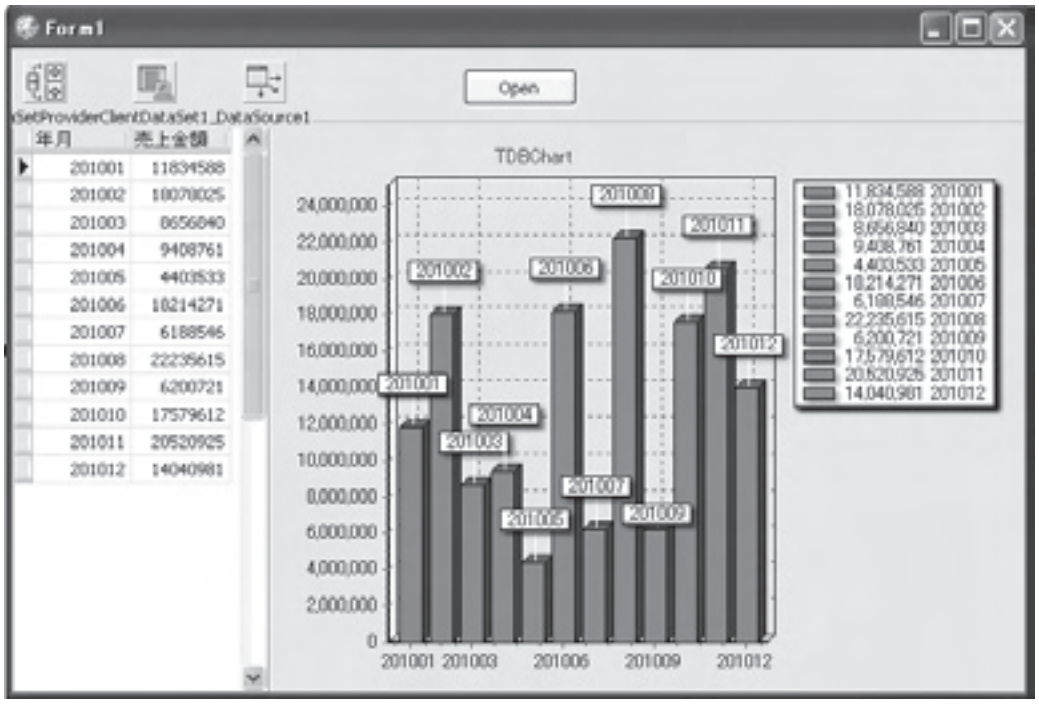

Editing DBChart1
Series
Series1
Chart
General
Axis
Titles
Legend
Panel
Paging
Walls
3D
Data
Export
Print
Series1
Bar: Series1
Format
Stack
General
Marks
Data Source
Dataset
Dataset: ClientDataSet1
Apply
X軸のラベル
X軸
Y軸
Labels: SALESYM
X: SALESYM
DateTime
Bar:
DateTime
Help...
Close

図5


Form1
Open
SetProviderClientDataSet1_DataSource1
年月
売上金額
201001 11834588
201002 18078025
201003 8656840
201004 9408761
201005 4403533
201006 18214271
201007 6188546
201008 22235615
201009 6200721
201010 17579612
201011 20520925
201012 14040981
TDBChart
24,000,000
22,000,000
20,000,000
18,000,000
16,000,000
14,000,000
12,000,000
10,000,000
8,000,000
6,000,000
4,000,000
2,000,000
0
201001 201003 201006 201009 201012
11,834,588 201001
18,078,025 201002
8,656,840 201003
9,408,761 201004
4,403,533 201005
18,214,271 201006
6,188,546 201007
22,235,615 201008
6,200,721 201009
17,579,612 201010
20,520,925 201011
14,040,981 201012

に順にアクセスしてプロパティを設定することもできるので、後者の方法を説明する。

最初の TSeries は DBChart1.Series[0]、次に Series[1] となる。また TDBChart にある TSeries の数は SeriesCount である。これらを使ってループさせることで、順に TSeries にアクセスし、そのプロパティを操作することができる。【ソース 2】

実行すると、起動時にはマーカーは表示されていない。そしてチェックボックスを操作すると、それに対応して、マーカーの表示・非表示が切り替えられていることが図 15 のように確認できる。【図 15】

4.TChart

TDBChart は、グラフの対象となるデータセットとフィールドを設定するだけで、自動でグラフが作成される。内部的にデータを解析して行われるため、コーディングを一切行わずに作成でき、非常に便利なコンポーネントである。だが、データから自動で行われるために、想定通りのグラフとならないことがある。

そのようなケースには、「TChart」を使用するとよい。これは、TDBChart と異なり、データ連動はしない。TChart では、グラフのポイント（座標）はすべてコードで記述するので、任意の位置にグラフを描画することができ、TDBChart で対応できないときには有効である。

さて、その TChart の設定は、TDBChart と基本的に同じで、TSeries を追加する。

初期状態では、ダミーのデータがセットされている。ダミーのデータを削除するか、FormCreate 時にソース 3 のコードでクリアすることで、なにもセットされていない図 16 のように初期状態となる。【図 16】【ソース 3】

グラフの座標設定

グラフの座標を設定する方法には、いくつかの関数がある。

① AddY（ラベルなし）

ソース 4 のように、Y 座標の値のみを関数に引き渡す。すると、X 座標値は 0 から自動で連番となることが図 17 でわかる。【図 17】【ソース 4】

② AddY（ラベルあり）

ソース 5 のように、Y 座標の値を関数に引き渡し、また同じ AddY から X 座標のラベルを指定することもできる。結果は図 18 となり、値でなく指定した文字が表示されている。【図 18】【ソース 5】

③ AddXY

ソース 6 のように、X 座標と Y 座標を指定すると、X 座標の位置を指定できる。これにより、図 19 のように、連続していないグラフが作成できる。【図 19】【ソース 6】

また座標の追加以外、TChart の場合も TDBChart と同じく、追加した座標から自動で外観の設定が行われる。このため、タイトルやラベルの表示等を変更したい箇所については、TDBChart のようにプロパティから設定することができる。

TChartの使用例

ここで、TChart を使用する方がよい場合を説明する。

例えば、座標が（得意先コード、売上金額）であり、データ値が（1001, 100）（1002, 200）（2001, 300）（2002, 400）となるデータがあったとする。

TDBChart では、図 20 のように“1001”と“1002”は連番となるが、“1002”と“2001”の間は連番とならず、間が空いてしまう。また X 座標の差が、X 軸全体の幅に対して小さすぎるため、“1001”と“1002”、“2001”と“2002”でグラフが重なってしまっている。【図 20】

これは、X 軸の値が、データ値である得意先コードの“1001”“1002”“2001”“2002”であり、値が連続していないためである。TDBChart では、あくまでデータ値からグラフの座標が決定されるのであって、レコード順で連番には設定されない。

これに対して TChart では、座標の位置を指定できるので、ファイルを読み込んで順にグラフ表示するときに、ソース 7 のように AddY を使って Y 値とラベルを指定する。

その実行結果が図 21 である。得意先コードが X 軸にラベル表示されながら、レコード順に連続してグラフが作成されている。【図 21】【ソース 7】

5.補足

今回の説明ではすべて棒グラフを使用した。棒グラフ以外にも、多くの種類の形状が用意されている。

TSeries を追加する場合に指定できるが、グラフを作成した後からでも図 22 のように、Change ボタンから種類を変更できる。図 23 は変更前（Bar ＝棒グラフ）と変更後（Line ＝折れ線グラフ）の画面である。【図 22】【図 23】

また立体表示されているが、平面表示にすることもできる。これも、設定画面からでも行えるし、また、TDBChart のプロパティ View3D の True/False で切り替えることも可能である。切り替えた結果は図 24 となる。【図 24】

6.注意点

グラフであるため、TDBChart の TSeries で X 軸と Y 軸に指定するフィールドは、数値型あるいは日付型時刻型でなければならない。

もし文字型フィールドであれば、SQL 文の INT でキャストしたり、データセットの計算項目で数値型に変換したり、連番をセットするなどの対応が必要になる。

TChart であれば、ソースコード内から設定するので、StrToInt 関数などを使用して値を変換する方法などがある。

7.まとめ

TDBChart と TChart の大まかな内容を説明したが、非常に簡単な設定だけでグラフ表示できることが理解いただけたかと思う。

プロパティの数が非常に多く、慣れないうちは戸惑うかもしれないが、プロパティ構成は各軸やタイトル、凡例などの要素ごとに階層になっているので、どの要素かわかれば操作しやすい。

またプロパティが多くあることは、逆に考えると、コーディングなしでもグラフを多彩に変更することが可能ということになる。さらにコードを併用することで、利用の幅が広がることが期待できる。

M

図6

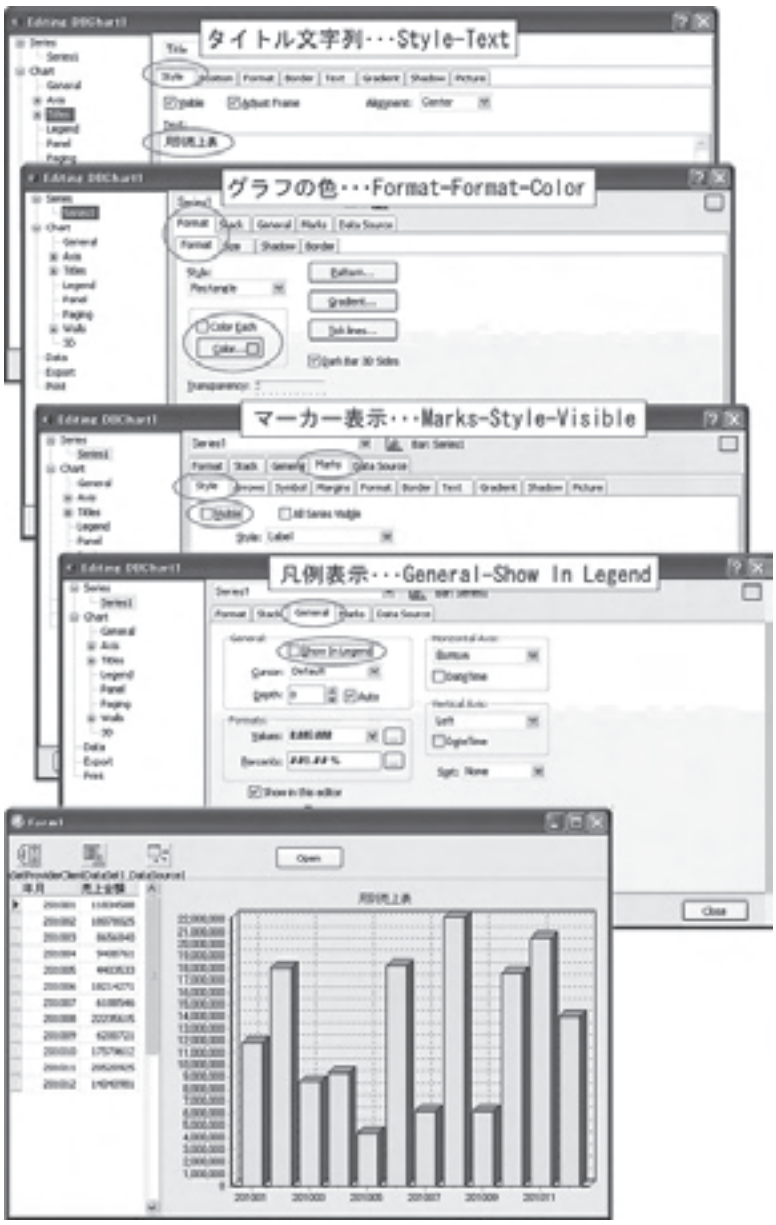

タイトル文字列･･･Style-Text
グラフの色･･･Format-Format-Color
マーカー表示･･･Marks-Style-Visible
凡例表示･･･General-Show In Legend

図7


ラベル表示･･･Scales-Increment-Show all labels

図8

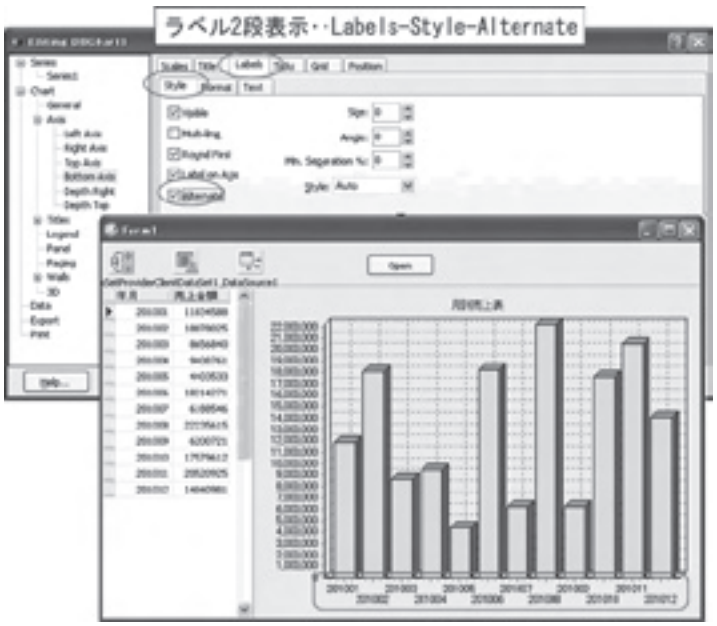


図9

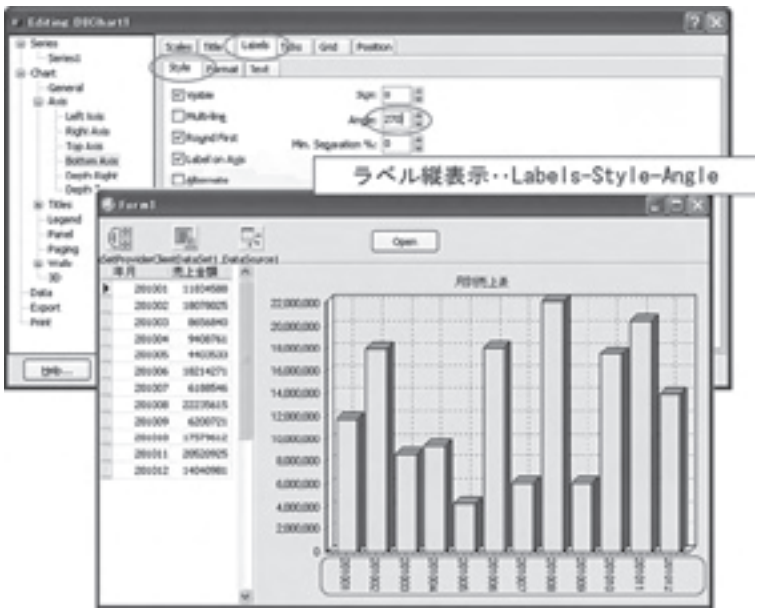


図10

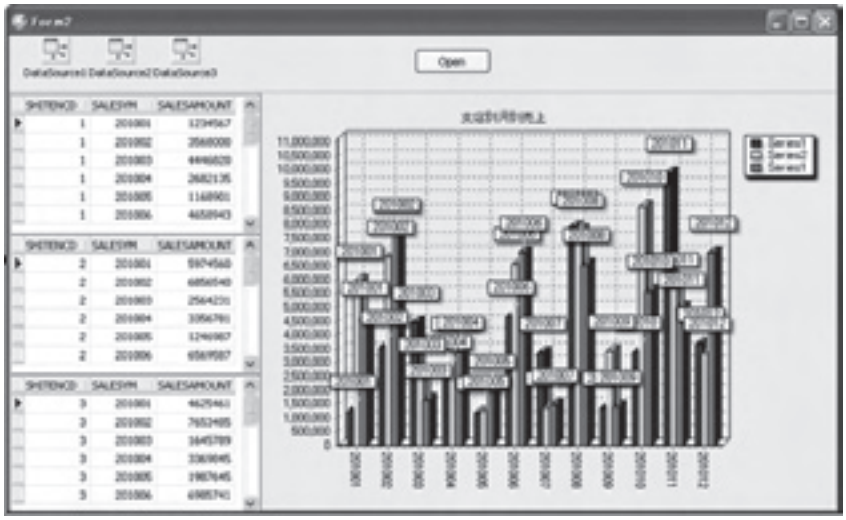


図11

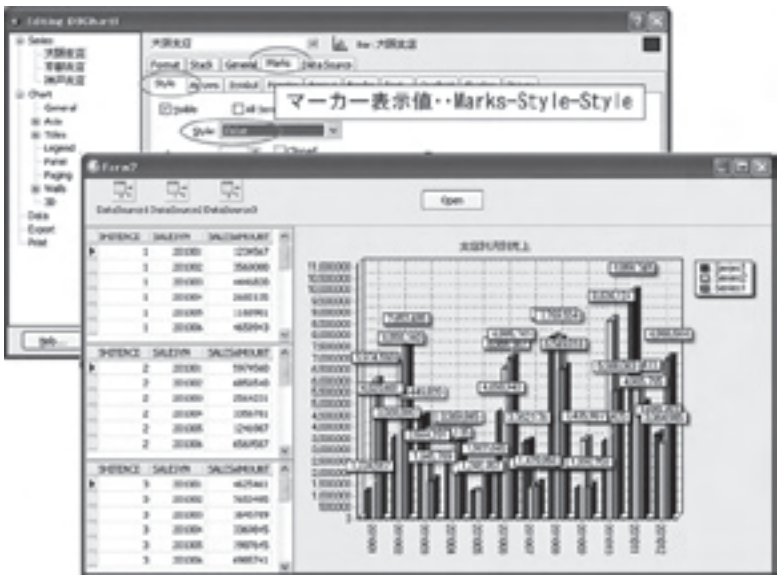


図12

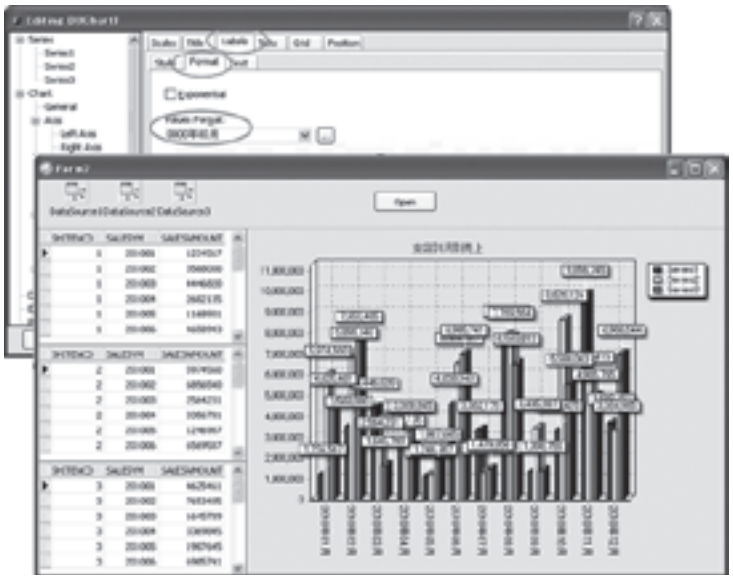


図13

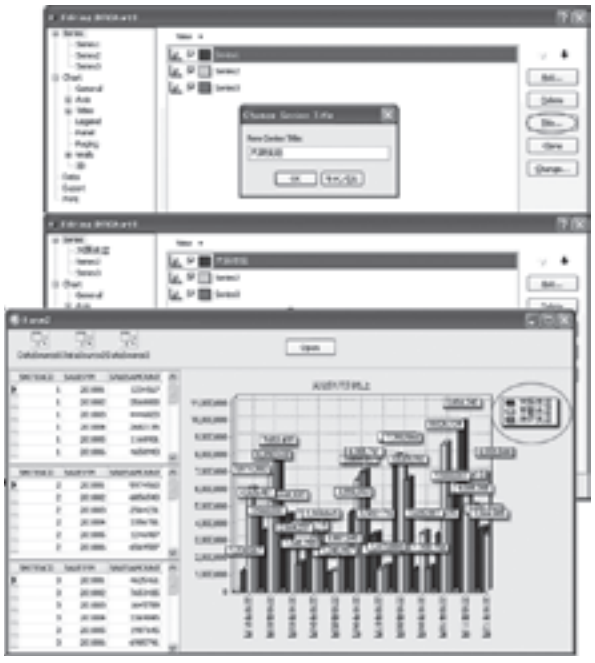


図14


オブジェクト インスペクタ
DBChart1 TDBChart
プロパティ
イベント
BottomAxis
(TChartAxis)
Automatic
False
AutomaticMaximum
False
AutomaticMinimum
True
Axis
(TChartAxisP
AxisValuesFormat
0000年00月
DateTimeFormat
EndPosition
100
ExactDateTime
True
Grid
(TAxisGridPe
GridCentered
False
TeeChart Standard v8.05.50522 Win32
© 1995-2009 by Steema Software
About TeeChart... Edit Chart...
AxisValuesFormat
すべての項目が表示されています

図15

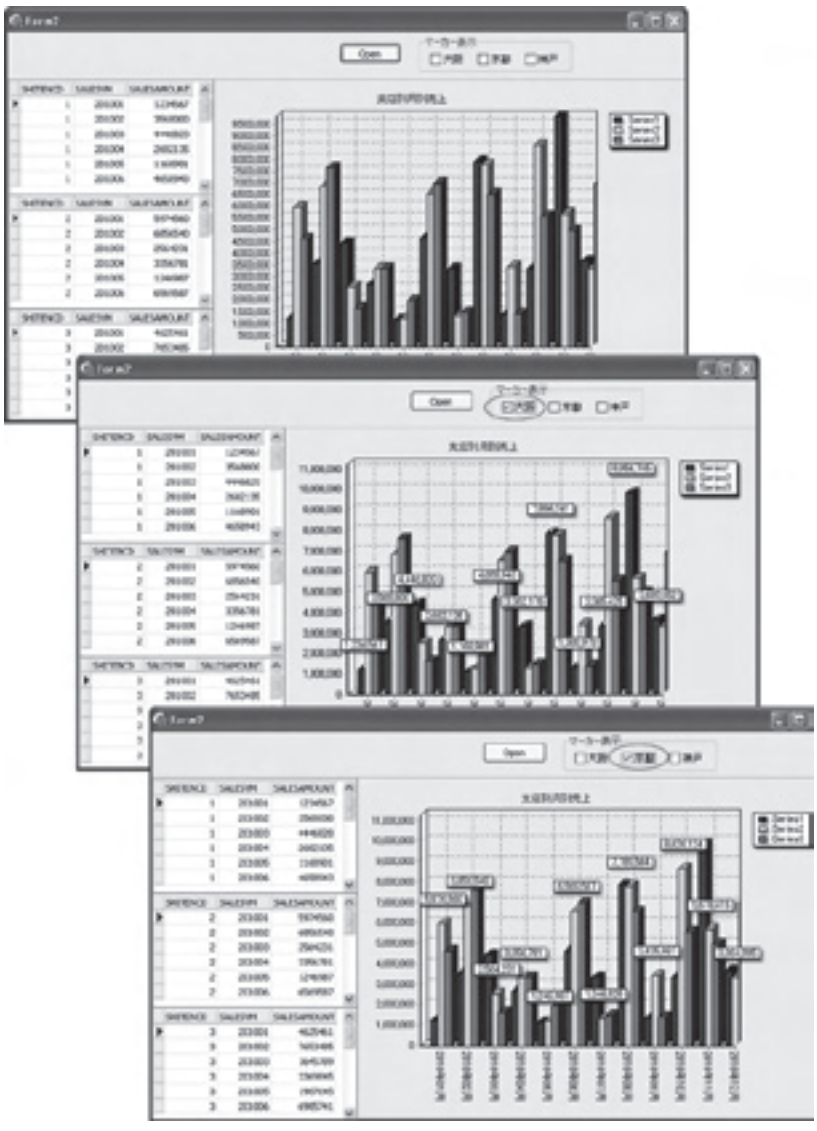


図16


Form3
実行

図17

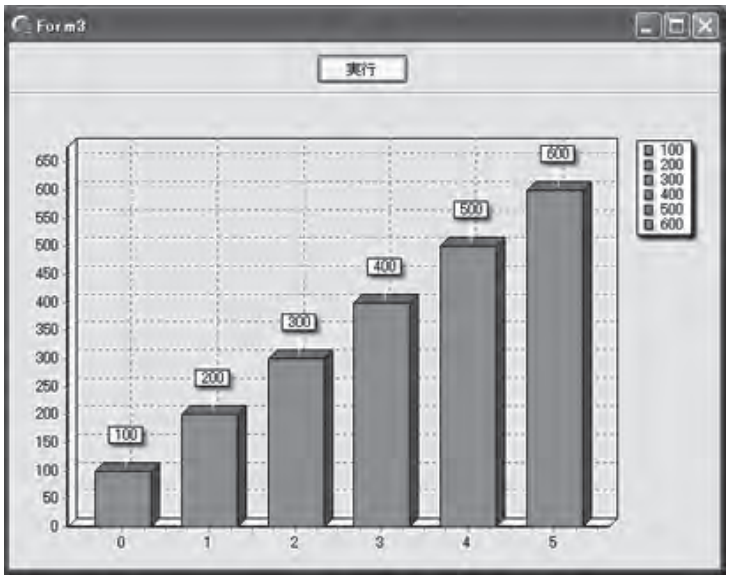

Form3
実行
650
600
550
500
450
400
350
300
250
200
150
100
50
0
0
1
2
3
4
5
100
200
300
400
500
600
100 200 300 400 500 600

図18

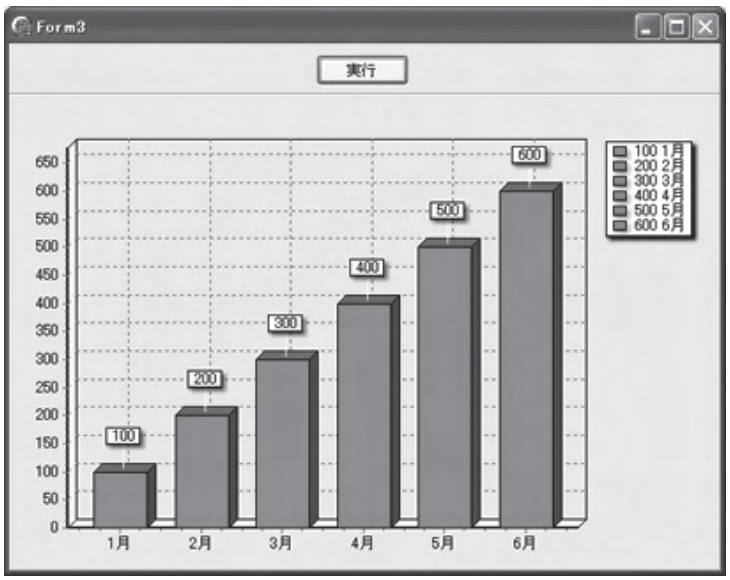

Form3
実行
650
600
550
500
450
400
350
300
250
200
150
100
50
0
1月
2月
3月
4月
5月
6月
100
200
300
400
500
600
100 1月
200 2月
300 3月
400 4月
500 5月
600 6月

図19

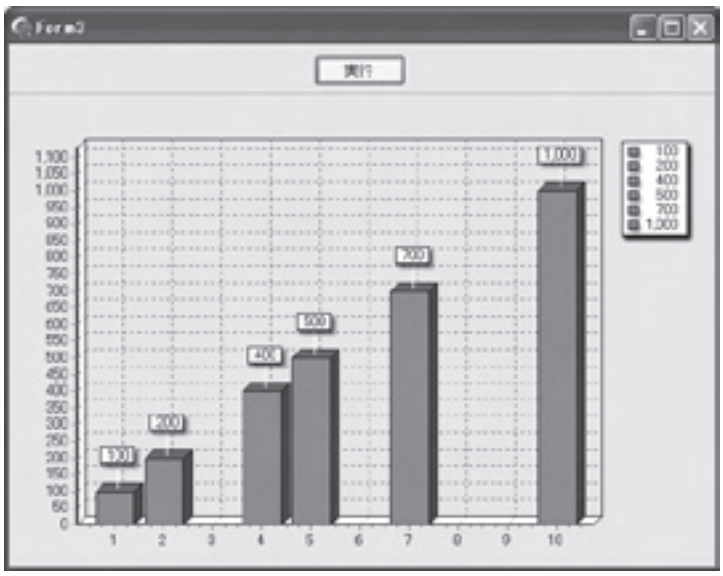

Form3
実行
100
200
400
500
700
1,000

図20

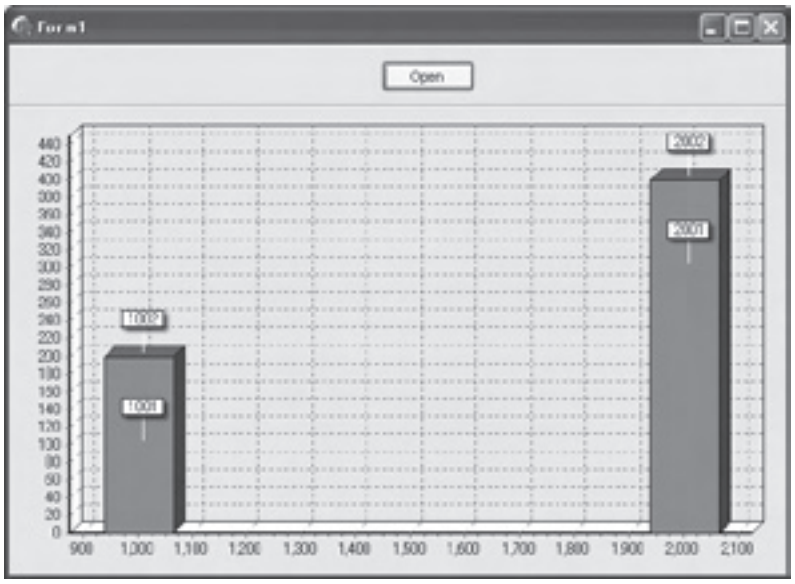

Form1
Open
1001
1002
2001
2002

図21

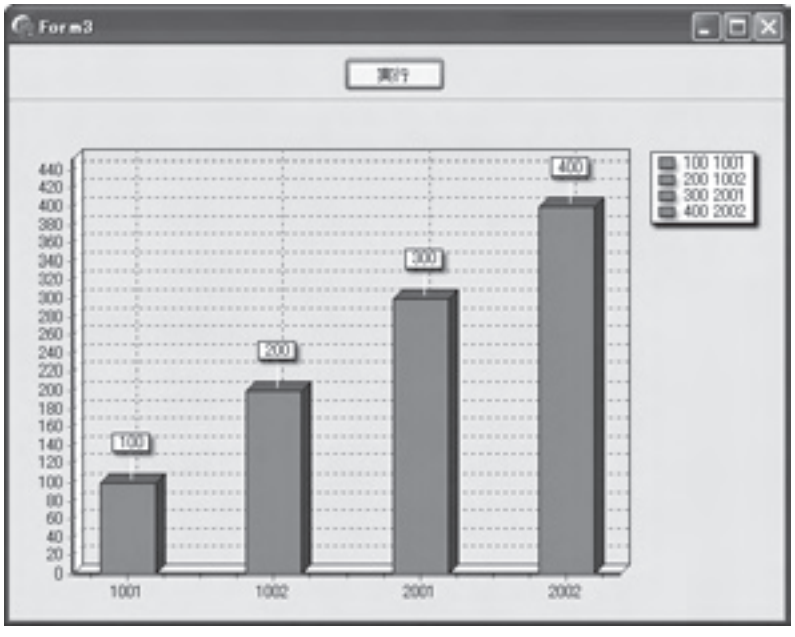

Form3
実行
100 1001
200 1002
300 2001
400 2002
1001
1002
2001
2002

図22

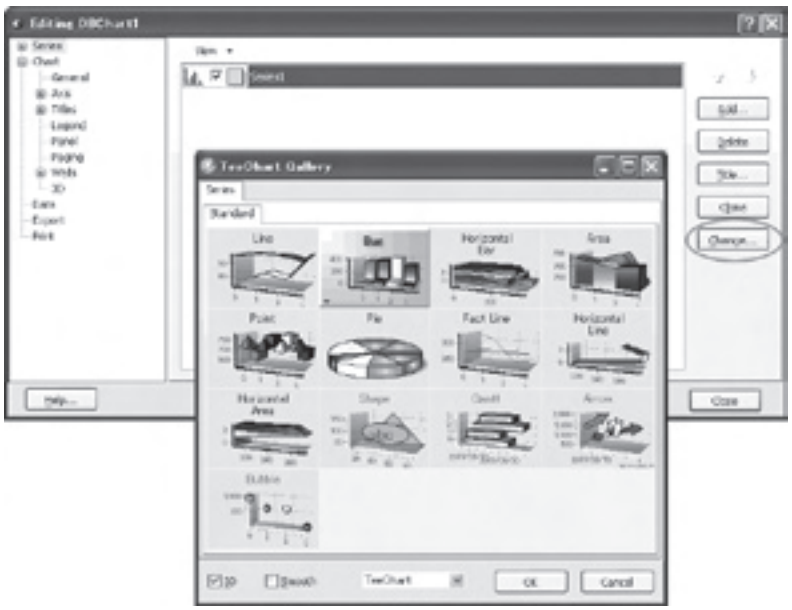


図23

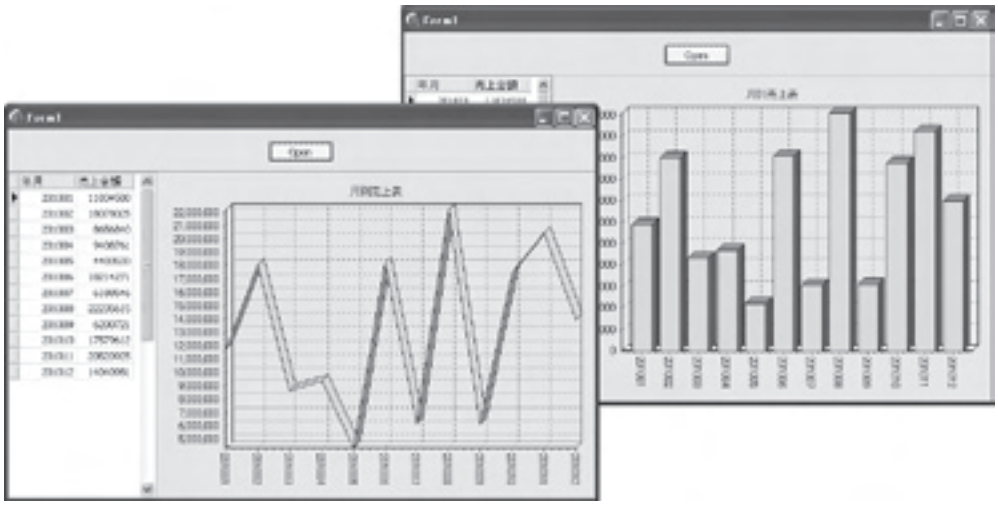
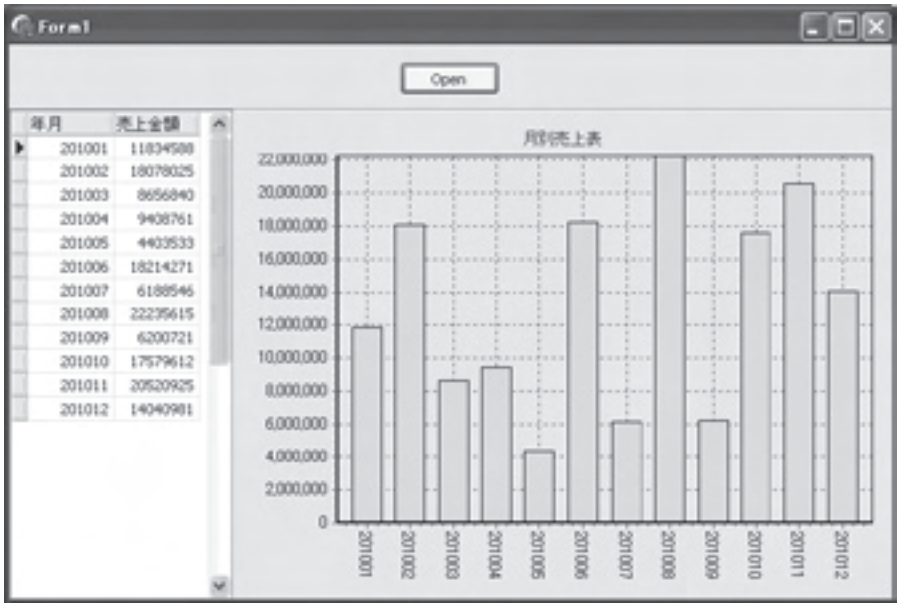


図24


Form1
Open
年月
売上金額
201001 11834588
201002 18078025
201003 8656840
201004 9408761
201005 4403533
201006 18214271
201007 6188546
201008 22235615
201009 6200721
201010 17579612
201011 20520925
201012 14040981
月別売上表
22,000,000
20,000,000
18,000,000
16,000,000
14,000,000
12,000,000
10,000,000
8,000,000
6,000,000
4,000,000
2,000,000
0
201001
201002
201003
201004
201005
201006
201007
201008
201009
201010
201011
201012

ソース1

```
procedure TForm2.ChbOsakaClick(Sender: TObject);
begin
  Series1.Marks.Visible := ChbOsaka.Checked;
end;
```

ソース2

```
procedure TForm2.FormCreate(Sender: TObject);
var
  i : integer;
begin
  for i := 0 to DBChart1.SeriesCount - 1 do
    DBChart1.Series[i].Marks.Visible := False;
end;
```

ソース3

```
procedure TForm3.FormCreate(Sender: TObject);
begin
  Series1.Clear;
end;
```

ソース4

```
procedure TForm3.Button1Click(Sender: TObject);
begin
  Series1.AddY(100);
  Series1.AddY(200);
  Series1.AddY(300);
  Series1.AddY(400);
  Series1.AddY(500);
  Series1.AddY(600);
end;
```

ソース5

```
procedure TForm3.Button1Click(Sender: TObject);
begin
  Series1.AddY(100,'1月');
  Series1.AddY(200,'2月');
  Series1.AddY(300,'3月');
  Series1.AddY(400,'4月');
  Series1.AddY(500,'5月');
  Series1.AddY(600,'6月');

end;
```

ソース6

```
procedure TForm3.Button1Click(Sender: TObject);
begin
  Series1.AddXY( 1, 100);
  Series1.AddXY( 2, 200);
  Series1.AddXY( 4, 400);
  Series1.AddXY( 5, 500);
  Series1.AddXY( 7, 700);
  Series1.AddXY(10,1000);

end;
```

ソース7

```
procedure TForm3.Button1Click(Sender: TObject);
begin
  with ClientDataSet1  do
  begin
    Open;
    while not eof do
    begin
      Series1.AddY(FieldByName('KINGAKU').AsInteger,
                   IntToStr(FieldByName('TOKUCD').AsInteger));
      Next;
    end;
  end;
end;
```