

【A5】 Delphi/C++Builderテクニカルセッション

「知って得する！現役ヘルプデスクが答える Delphiテクニカルエッセンス 2.0」

株式会社ミガロ
RAD事業部 技術支援課 顧客サポート
吉原 泰介

アジェンダ

- ミガロについて
- 問合せ分析
- よくある問合せQ&A

- 【Q1】DBGridで簡単にソートを行いたい！
- 【Q2】VB-Report Ver3.0で効率よく出力したい！
- 【Q3】TreeViewでメニューを制御したい！
- 【Q4】一覧明細にローカル画像を表示したい！
- 【Q5】StringGridで文字列を縦表示したい！

- サポートより

ミガロについて

株式会社ミガロ - A 100% System i Company -

会社情報

所在地: 本社 大阪市浪速区湊町2-1-57
 難波サンケイビル13F
 拠点 東京都港区麻布台1-4-3
 エグゼクティブタワー麻布台11F

事業内容

IBM System i 向けのソフトウェア・ツール販売および技術サポート

- | | |
|----------------------------|-----------|
| 【開発ツール】 | 【災害対策ソフト】 |
| • Delphi/400 | • *noMAX |
| • JACI400 | |
| 【アプリケーション・ライフサイクル・マネージメント】 | |
| • MKS | |

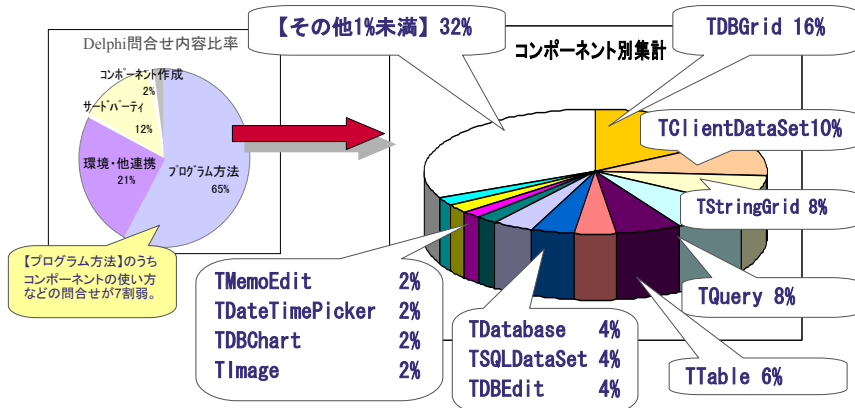
Delphi/400

DelphiをSystem iに完全対応させたミドルウェア

国内約570社、全世界約5,000社の導入実績

問合せ分析(コンポーネント)

☐ サポートには月100件弱の問合せや個別技術サポート



【Q1】DBGridで簡単にソートを行いたい！

OnTitleClick処理

```
procedure Tfrm1.DBGrid1.TitleClick(Column: TColumn);
var
  sFieldNM :String; // 退避フィールド名
begin
  with DBGrid1 do
  begin
    // 明細非表示時は処理無効
    if DataSource.DataSet.Active = False then Exit;

    // カラムのフィールド名を退避
    sFieldNM := Column.FieldName;

    with (DataSource.DataSet as TClientDataSet) do
    begin
      // インデックスフィールド作成
      if IndexDefs.Count = 0 then
        IndexDefs.Add('alndex', '', []);
```

```
//----- 降順へ -----
// 表題設定
Column.Title.Caption :=
StringReplace(Column.Title.Caption, '▲', '', [rfReplaceAll]);
Column.Title.Caption := Column.Title.Caption + '▼';

// 降順フィールドの設定
if IndexDefs[0].DescFields = '' then
  IndexDefs[0].DescFields := sFieldNM
else
  IndexDefs[0].DescFields := IndexDefs[0].DescFields + ',' + sFieldNM;

// インデックスフィールドのオプション初期化
IndexDefs[0].Options := [];
end
else if AnsiPos('▼', Column.Title.Caption) <> 0 then
begin
  //----- 設定解除 -----
  // 表題設定
  Column.Title.Caption :=
StringReplace(Column.Title.Caption, '▼', '', [rfReplaceAll]);
```

②【昇順状態の場合】
クリックされたField名を
降順のためDescFieldsに追加

③【降順状態の場合】
クリックされたField名を
両方のリストから削除

【Q1】DBGridで簡単にソートを行いたい！

```
// 並び順の設定解除 (降順フィールド)
if AnsiPos(sFieldNM + ',', IndexDefs[0].DescFields) <> 0 then
  sFieldNM := sFieldNM + ',';
else if AnsiPos(',') + sFieldNM, IndexDefs[0].DescFields) <> 0 then
  sFieldNM := ',' + sFieldNM;

IndexDefs[0].DescFields := StringReplace(IndexDefs[0].DescFields,
  sFieldNM, '', [rfReplaceAll]);

// カラムのフィールド名を再取得
sFieldNM := Column.FieldName;

// 並び順の設定解除 (昇順フィールド)
if AnsiPos(sFieldNM + ',', IndexDefs[0].Fields) <> 0 then
  sFieldNM := sFieldNM + ',';
else if AnsiPos(',') + sFieldNM, IndexDefs[0].Fields) <> 0 then
  sFieldNM := ',' + sFieldNM;

IndexDefs[0].Fields :=
StringReplace(IndexDefs[0].Fields, sFieldNM, '', [rfReplaceAll]);
// 並び替え完全解除の場合インデックス名クリア
if IndexDefs[0].Fields = '' then IndexName := '';

// インデックスフィールドのオプション初期化
IndexDefs[0].Options := [];
end
```

```
else
begin
  //----- 昇順へ -----
  // 表題設定
  Column.Title.Caption := Column.Title.Caption + '▲';

  // 昇順フィールドの設定
  if IndexDefs[0].Fields = '' then
    IndexDefs[0].Fields := sFieldNM
  else
    IndexDefs[0].Fields := IndexDefs[0].Fields + ',' + sFieldNM;

  // インデックスフィールドのオプション初期化
  IndexDefs[0].Options := [];
  // インデックス名設定
  if IndexName = '' then IndexName := 'alndex';
end;

// データを開いたままソートを適用させるため
// Indexを削除して再設定
DeleteIndex('alndex');
IndexName := 'alndex';
First;
end;
end;
end;
```

①【解除状態の場合】
クリックされたField名を
昇順のためFieldsに追加

【Q1】DBGridで簡単にソートを行いたい！ - C++Builder版

OnTitleClick処理

```
void __fastcall TForm1::DBGrid1TitleClick(TColumn *Column)
{
    AnsiString sFieldNM; // 退避フィールド名

    // 明細非表示時は処理無効
    if(DBGrid1->DataSource->DataSet->Active == false)
        return;

    // カラムのフィールド名を退避
    sFieldNM = Column->FieldName;

    TClientDataSet* cds =
        (TClientDataSet*)(DBGrid1->DataSource->DataSet);
    // インデックスフィールド作成
    if(cds->IndexDefs->Count == 0)
        cds->IndexDefs->Add("alndex", "", TIndexOptions());
}
```

```
//----- 降順 -----
// 明細の表題/並び替えの制御 //
//----- 降順 -----
if(AnsiPos("▲", Column->Title->Caption) != 0) {
    // 表題設定
    Column->Title->Caption = StringReplace(Column->Title->Caption, "▲", "",
        TReplaceFlags() << rfReplaceAll);
    Column->Title->Caption = Column->Title->Caption + "▼";

    // 降順フィールドの設定
    if(cds->IndexDefs->Items[0]->DescFields == "")
        cds->IndexDefs->Items[0]->DescFields = sFieldNM;
    else
        cds->IndexDefs->Items[0]->DescFields =
            cds->IndexDefs->Items[0]->DescFields + "," + sFieldNM;

    // インデックスフィールドのオプション初期化
    cds->IndexDefs->Items[0]->Options = TIndexOptions();
}
else if(AnsiPos("▼", Column->Title->Caption) != 0) {
    // ----- 設定解除 -----
    // 表題設定
    Column->Title->Caption =
        StringReplace(Column->Title->Caption,
            "▼", "", TReplaceFlags() << rfReplaceAll);
}
```

②【昇順状態の場合】
クリックされたField名を
降順のためDescFieldsに追加

③【降順状態の場合】
クリックされたField名を
両方のリストから削除

【Q1】DBGridで簡単にソートを行いたい！ - C++Builder版

```
// 並び順の設定解除 (降順フィールド)
if(AnsiPos(sFieldNM + ",", cds->IndexDefs->Items[0]->DescFields) != 0)
    sFieldNM = sFieldNM + ",";
else if(AnsiPos(", " + sFieldNM, cds->IndexDefs->Items[0]->DescFields) != 0)
    sFieldNM = ", " + sFieldNM;

cds->IndexDefs->Items[0]->DescFields =
    StringReplace(cds->IndexDefs->Items[0]->DescFields, sFieldNM, "",
        TReplaceFlags() << rfReplaceAll);

// カラムのフィールド名を再取得
sFieldNM = Column->FieldName;

// 並び順の設定解除 (昇順フィールド)
if(AnsiPos(sFieldNM + ",",
    AnsiString(cds->IndexDefs->Items[0]->Fields)) != 0)
    sFieldNM = sFieldNM + ",";
else if(AnsiPos(", " + sFieldNM,
    AnsiString(cds->IndexDefs->Items[0]->Fields)) != 0)
    sFieldNM = ", " + sFieldNM;

cds->IndexDefs->Items[0]->Fields =
    StringReplace(cds->IndexDefs->Items[0]->Fields, sFieldNM, "",
        TReplaceFlags() << rfReplaceAll);

// 並び替え完全解除の場合インデックス名クリア
if(cds->IndexDefs->Items[0]->Fields == "") cds->IndexName = "";

// インデックスフィールドのオプション初期化
cds->IndexDefs->Items[0]->Options = TIndexOptions();
}
```

```
else {
    // ----- 昇順 -----
    // 表題設定
    Column->Title->Caption = Column->Title->Caption + "▲";

    // 昇順フィールドの設定
    if(cds->IndexDefs->Items[0]->Fields == "")
        cds->IndexDefs->Items[0]->Fields = sFieldNM;
    else
        cds->IndexDefs->Items[0]->Fields =
            cds->IndexDefs->Items[0]->Fields + "," + sFieldNM;

    // インデックスフィールドのオプション初期化
    cds->IndexDefs->Items[0]->Options = TIndexOptions();

    // インデックス名設定
    if(cds->IndexName == "") cds->IndexName = "alndex";
}
// データを開いたままソートを適用させるため
// Indexを削除して再設定
cds->DeleteIndex("alndex");
cds->IndexName = "alndex";
cds->First();
}
```

①【解除状態の場合】
クリックされたField名を
昇順のためFieldsに追加

【Q2】VB-Report Ver3.0で効率よく出力したい！



【質問】

アドバンスソフトウェア株式会社 <http://www.adv.co.jp/>

VB-Report Ver3.0を使用して帳票・Excelの出力をしています。パフォーマンスを見直すにはどうしたら良いでしょうか？



【回答】

VB-Report Ver3.0の出力パフォーマンスにはフォーマットとなるExcel自体が大きく影響します。セル数や属性設定などを見直すことでパフォーマンスを向上させることができます。

【Q2】VB-Report Ver3.0で効率よく出力したい！

フォーマットExcelと扱いの考慮点

- ①Cellの属性や設定はフォーマット側で事前に設定する。
- ②フォーマット内の情報を少なくする。
Cell数、書式、属性、結合、罫線、網掛等
- ③可能であれば固定(A1参照や座標)形式で指定する。
変数指定の場合、変数の位置検索が行われるため、通常の固定指定よりもパフォーマンスが落ちます。

【Q2】VB-Report Ver3.0で効率よく出力したい！

フォーマットExcelと扱いの考慮点

- ④位置指定で変数を使用する場合
先頭(或いは前処理)行から小さい行、小さい列の順に
処理を行う。

変数位置検索

**A1	**B1	**C1	**D1	**E1
**A2	**B2	**OZ	**D2	**E2
**A3	**B3	**C3	**D3	**E3

【Q3】TreeViewでメニューを制御したい！

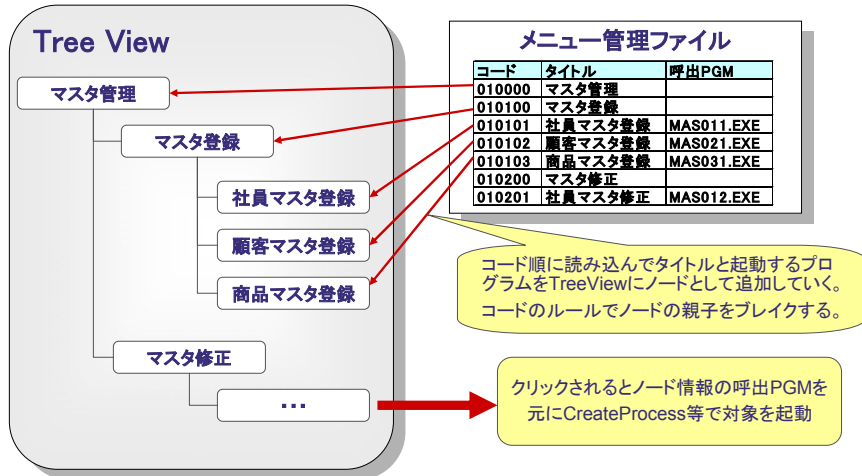
!【質問】

メニュー内容をDB上のファイルで動的に構成できない
でしょうか？

✉【回答】

メニューの作成手法はいろいろありますが、
TreeViewを使用すると見やすいメニューを動的制御する
ことができます。

【Q3】TreeViewでメニューを制御したい！



【Q3】TreeViewでメニューを制御したい！

```

~~~~~
type
  PTVRec = ^TTVRec;
  TTVRec = record
    EXENM: string;
  end;
~~~~~

procedure TfrmQ3.LoadMenu;
var
  TopNode: TTreeNode; //追加されたトップノード
  ChdNode1: TTreeNode; //追加された子ノード
  ChdNode2: TTreeNode; //追加された孫ノード
  TVRecPtr: PTVRec; //ノード情報
  BrkCd1: string; //大区分
  BrkCd2: string; //中区分
  Title: string; //タイトル
  MenuCD: string; //メニューコード
begin
  //既存のノードを削除
  TreeView1.Items.Clear;

```

ノード制御処理

```

//初期設定
TopNode := nil;
BrkCd1 := "";
BrkCd2 := "";

//ファイルより追加
with Datamodule1.CdsMenu do
begin
  First;
  while not eof do
  begin
    //データをセット
    New(TVRecPtr);

```

ノード情報整理

```

//データレコードからノード情報を取得
MenuCD := 'メニューコード';
Title := 'メニュー上の名称';
TVRecPtr^.EXENM := 'メニューからの呼び出すPGM';

```


【Q3】TreeViewでメニューを制御したい！

```
// トップノード
if Copy(MenuCD,1,2) <> BrkCd1 then
begin
  TopNode := TreeView1.Items.AddChildObject(nil,Title,TVRecPtr);
end
else
begin
if Copy(MenuCD,3,2) <> BrkCd2 then
//子ノード
begin
//ノード追加
ChdNode1 := TreeView1.Items.AddChildObject(TopNode,Title,TVRecPtr);
end
else
//孫ノード
begin
//中分類'00'のとき、親ノードがトップノード
if Copy(MenuCD,3,2) = '00' then
ChdNode1 := TopNode;

ChdNode2 := TreeView1.Items.AddChildObject(ChdNode1,Title,TVRecPtr);
end;
end;
```

1階層ノード追加

2階層ノード追加

3階層ノード追加

```
//大中分類回避
BrkCd1 := Copy(MenuCD,1,2);
BrkCd2 := Copy(MenuCD,3,2);

//次データへ
Next;
end;
First;
end;

//全ノードを展開
TreeView1.FullExpand;

end;
```

ブレイク用にコード回避

TreeViewのImagesプロパティでTImageListを設定して表示アイコンを持つことができます。
事前にアイコンを登録したり、実際の実行ファイルからExtractIconを使用して動的にアイコンを取得・設定することも可能です。

【Q3】TreeViewでメニューを制御したい！

TreeViewの他の使用例

TreeViewは他にも体系的なデータの一覧検索照会画面などで利用することにより、ユーザーが見た目で使い易い画面にすることができます。



【Q3】TreeViewでメニューを制御したい！ - C++Builder版

```
~~~~~
struct TTVRec {
    AnsiString EXENM;
};
typedef TTVRec* PTVRec;
~~~~~
```

ノード制御処理

```
void __fastcall TfrmQ3::LoadMenu()
{
    TTreeNode* TopNode; //追加されたトップノード
    TTreeNode* ChdNode1; //追加された子ノード
    TTreeNode* ChdNode2; //追加された子ノード
    PTVRec TVRecPtr; //ノード情報
    AnsiString BrkCd1; //大区分
    AnsiString BrkCd2; //中区分
    AnsiString Title; //タイトル
    AnsiString MenuCD; //メニューコード
    TClientDataSet* CdsMenu;

    //既存のノードを削除
    TreeView1->Items->Clear();
}
```

```
//初期設定
TopNode = NULL;
BrkCd1 = "";
BrkCd2 = "";

//ファイルより追加
CdsMenu = Datamodule1->CdsMenu;
CdsMenu->First();
while( ! CdsMenu->Eof() ) {
    //データをセット
    TVRecPtr = new TTVRec();

    //データレコードからノード情報を取得
    MenuCD = "メニューコード";
    Title = "メニュー上の名称";
    TVRecPtr->EXENM = "メニューから呼び出すPGM";
}
```

ノード情報整理

【Q3】TreeViewでメニューを制御したい！ - C++Builder版

```
//トップノード
if( MidStr(MenuCD,1,2) != BrkCd1 ) {
    TopNode = TreeView1->Items->AddChildObject(NULL,Title,TVRecPtr);
}
else {
    if( MidStr(MenuCD,3,2) != BrkCd2 ) {
        //子ノード
        //ノード追加
        ChdNode1 = TreeView1->Items->AddChildObject(TopNode,Title,TVRecPtr);
    }
    else {
        //孫ノード
        //中分類'00'のとき、親ノードがトップノード
        if( MidStr(MenuCD,3,2) == "00" )
            ChdNode1 = TopNode;
        ChdNode2 = TreeView1->Items->AddChildObject(ChdNode1,Title,TVRecPtr);
    }
}
```

1階層ノード追加

2階層ノード追加

3階層ノード追加

ブレイク用にコード遷遷

```
//大中分類遷遷
BrkCd1 = MidStr(MenuCD,1,2);
BrkCd2 = MidStr(MenuCD,3,2);

//次データへ
CdsMenu->Next();
CdsMenu->First();

//全ノードを展開
TreeView1->FullExpand();
}
```

TreeViewのImagesプロパティでTImageListを設定して表示アイコンを持つことができます。
事前にアイコンと登録したり、実際の実行ファイルからExtractIconを使用して動的にアイコンを取得・設定することも可能です。

【Q4】一覧明細にローカル画像を表示したい！

!【質問】

ファイルサーバやローカルにある画像ファイルを一覧検索の明細に表示できますか？

✉【回答】

DBのテーブル上にない画像を表示する場合、TImageを使用します。DBCtrlGrid上で一覧明細として表示したい場合は、OnPaintPanelイベントを利用することで実現が可能です。
またScrollBoxを使用しても同様の内容は作成できます。

【Q4】一覧明細にローカル画像を表示したい！

OnPaintPanel処理

```
procedure Tfrm1.DBCtrlGrid1PaintPanel(DBCtrlGrid: TDBCtrlGrid; Index: Integer);
begin
  with DBCtrlGrid.DataSource.DataSet do
  begin
    if not(Active) then Exit;
    //画像出力
    try
      //画像を読み込み(bmp画像)
      Image1.Picture.LoadFromFile('フィールド内容で画像を指定');
    except
      try
        Image1.Picture.LoadFromFile('対象画像ファイルがない場合の画像');
      except
        end;
      end;
    end;
  end;
end;
```

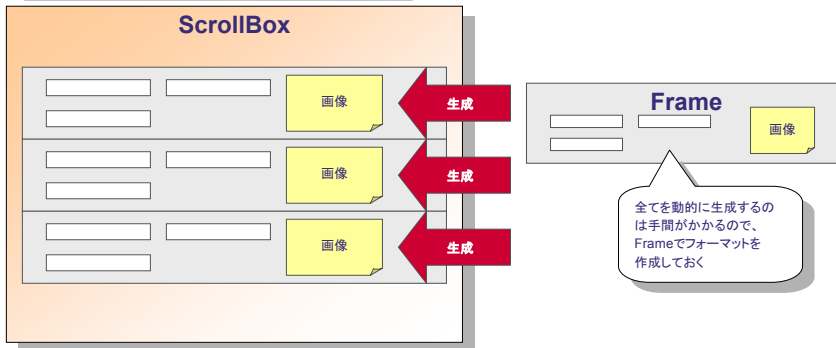


DBEditでDB上のデータと連動

描画処理でDB以外から随時画像を跳込

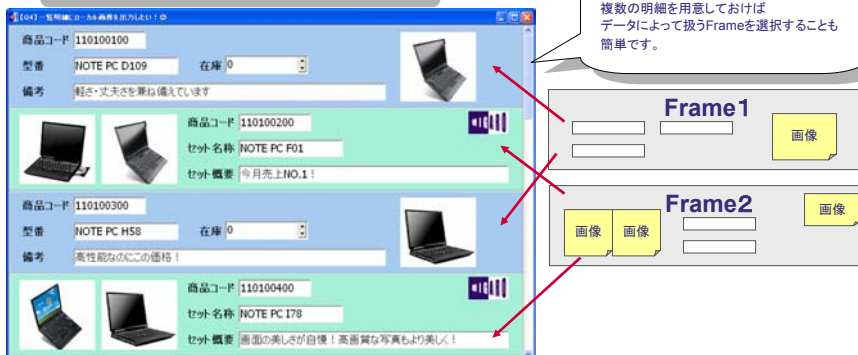
【Q4】一覧明細にローカル画像を表示したい！

ScrollBarで作成する場合



【Q4】一覧明細にローカル画像を表示したい！

ScrollBarで作成する場合



【Q5】StringGridで文字列を縦表示したい！

!【質問】

StringGridで表示される文字列を縦表示したいのですが、可能でしょうか？

✉【回答】

OnDrawCellイベントで対象のフォントを回転させてTextOutやDrawtextで出力させることで縦描写を行うことができます。

【Q5】StringGridで文字列を縦表示したい！

フォントの回転

回転したフォントを作成し、CanvasのFontに割り当てることで表示を調整します。

SingleByte文字とDoubleByte文字では縦書き用のフォントを使用すると回転による実際の向きが異なるので注意

	45度	90度	270度
Single-Byte ABCDEFGHIJ	ABCDEFGHIJ	ABCDEFGHIJ	ABCDEFGHIJ
Double-Byte あいうえお	あいうえお	あいうえお	あいうえお

MS ゴシック @MS ゴシック MS ゴシック @MS ゴシック MS ゴシック @MS ゴシック

【Q5】StringGridで文字列を縦表示したい！

OnDrawCell処理

```
procedure TfrmQ5.StringGrid1DrawCell(Sender: TObject;
  ACol, ARow: Integer; Rect: TRect; State: TGridDrawState);
var
  LFont: TLogFont;
  NFont, OFont: HFont;
begin
  with StringGrid1 do
  begin
    Canvas.Font.Size := 20;
    //Fontを取得し、回転させます。
    GetObject(Canvas.Font.Handle, SizeOf(LFont), @LFont);
    with LFont do
    begin
      フォントの回転調整
      IfEscapement := 900; //通常：0、左：900、右：2700
      IfCharSet := DEFAULT_CHARSET;
      IfOutPrecision := OUT_DEFAULT_PRECIS;
      IfClipPrecision := CLIP_DEFAULT_PRECIS;
      IfQuality := DEFAULT_QUALITY;
      IfPitchAndFamily := DEFAULT_PITCH;
      IfFaceName := '@MS ゴシック';
    end;
  end;
end;
```

```
//描画領域が持っているFontと入れ替えます。
NFont := CreateFontIndirect(LFont);
OFont := SelectObject(Canvas.Handle, NFont);
//描画領域の内容をFillRectでクリアします。
Canvas.FillRect(Rect);
//Fontが変更された状態で描画領域へ文字列を書き込みます。
Canvas.TextOut(CellRect(ACol, ARow).Left + 9,
  CellRect(ACol, ARow).top + 85,
  Cells[ACol, ARow]);
//入れ替えたFontを元に戻し、新たに設定したFont情報を
//開放します。
NFont := SelectObject(Canvas.Handle, OFont);
DeleteObject(NFont);
end;
```

縦書用フォント指定の場合、
DoubleByte文字の回転方向が
SingleByte文字の回転と向きが
若干異なります。

DrawTextで出力する場合は
DT_NOCLIPを指定する必要があります。

【Q5】StringGridで文字列を縦表示したい！ - C++Builder版

OnDrawCell処理

```
void __fastcall TfrmQ5::StringGrid1DrawCell(TObject *Sender,
  int ACol, int ARow,
  TRect &Rect, TGridDrawState State)
{
  TLogFont LFont;
  HFONT NFont, OFont;

  StringGrid1->Canvas->Font->Size = 20;
  //Fontを取得し、回転させます。
  GetObject(
    StringGrid1->Canvas->Font->Handle, sizeof(LFont), &LFont);
  LFont.IfEscapement = 900; //通常：0、左：900、右：2700
  LFont.IfCharSet = DEFAULT_CHARSET;
  LFont.IfOutPrecision = OUT_DEFAULT_PRECIS;
  LFont.IfClipPrecision = CLIP_DEFAULT_PRECIS;
  LFont.IfQuality = DEFAULT_QUALITY;
  LFont.IfPitchAndFamily = DEFAULT_PITCH;
  StrCopy(LFont.IfFaceName, "@MS ゴシック");
}
```

```
//描画領域が持っているFontと入れ替えます。
NFont = CreateFontIndirect(&LFont);
OFont = SelectObject(StringGrid1->Canvas->Handle, NFont);
//描画領域の内容をFillRectでクリアします。
StringGrid1->Canvas->FillRect(Rect);
//Fontが変更された状態で描画領域へ文字列を書き込みます。
StringGrid1->Canvas->TextOut(
  StringGrid1->CellRect(ACol, ARow).Left + 9,
  StringGrid1->CellRect(ACol, ARow).top + 85,
  StringGrid1->Cells[ACol][ARow]);
//入れ替えたFontを元に戻し、新たに設定したFont情報を解放します。
NFont = SelectObject(StringGrid1->Canvas->Handle, OFont);
DeleteObject(NFont);
}
```

フォントの回転調整

縦書用フォント指定の場合、
DoubleByte文字の回転方向が
SingleByte文字の回転と向きが
若干異なります。

DrawTextで出力する場合は
DT_NOCLIPを指定する必要があります。

サポートより

- 弊社ではIBM System i向けのソフトウェア・ツール販売および受託開発をしておりますが、Delphiという共通分野の情報は積極的にフィードバックしていきたいと考えています。

