

【セッションNo. 2】

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6.0

株式会社ミガロ.
RAD事業部 技術支援課
吉原 泰介

【アジェンダ】

Q1. IBM i を使ったファイルのアップロード/ダウンロード

Q2. トレイアイコンを利用したアプリケーションの常駐

Q3. リモートデバッガで運用PCのデバッグ調査

Q4. Delphi/400 V2010 新機能紹介
- ジェスチャアプリケーション作成 -

■ おまけ資料

・ Delphi/400 V2009V2010用MGRMaskEditコンポーネント

■ Q1. IBM i を使ったファイルのアップロード/ダウンロード



【質問】

宮城県農業信用基金協会 石井 孝彦様 リクエスト

IBM i にPCファイルをアップロードすることはできますか？



【回答】

PCファイルはFTPやP-COMMでバイナリ転送できます。
Delphi/400を使用したアプリケーションでPCファイルを
バイナリ転送することも可能です。

IndyのIdFTPというコンポーネントを使うと
簡単にIBM i とファイルの送受信することができます。

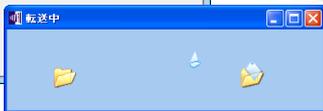
■ Q1. IBM i を使ったファイルのアップロード/ダウンロード

IBM i にファイルをアップロードする



2= 変更 3= コピー 4= 削除
8= 記述の表示 9= 保管

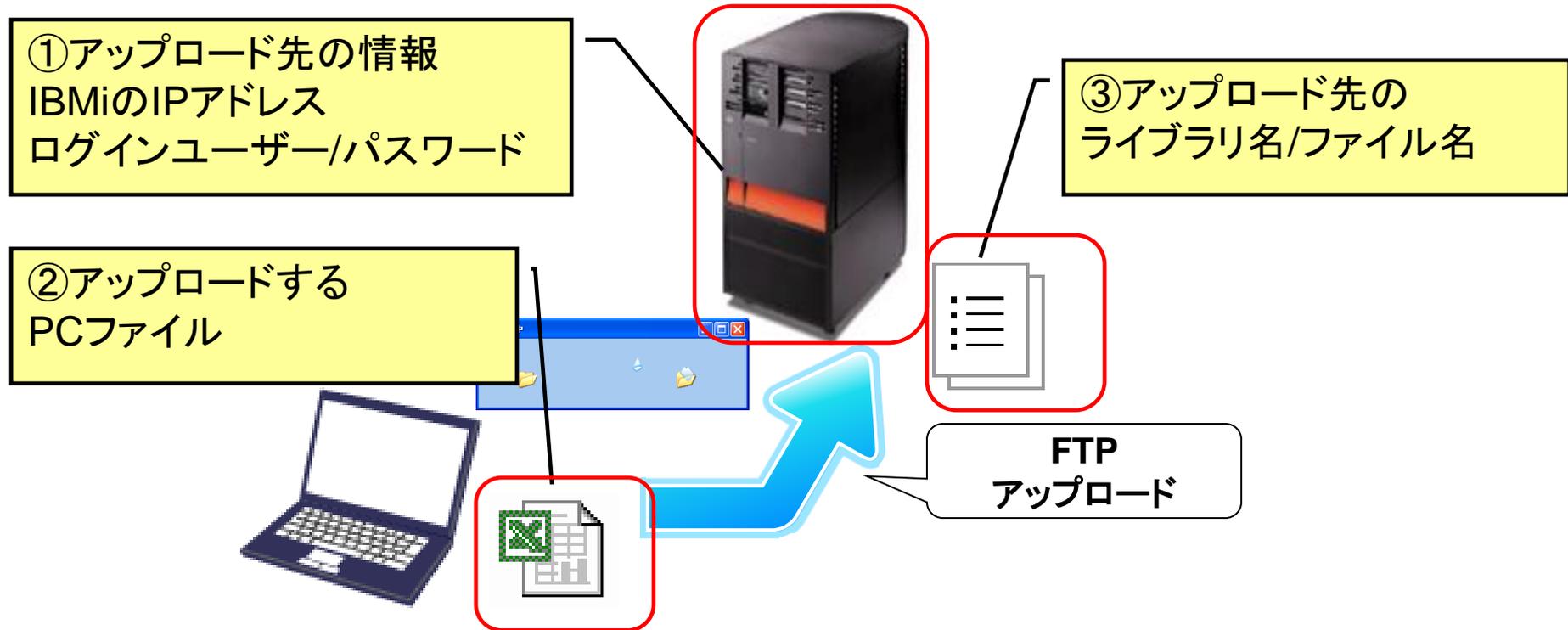
OPT	オブジェクト	タイプ	属性
—	SAMPLEXLS	*FILE	PF-DTA



FTP
アップロード

■ Q1. IBM i を使ったファイルのアップロード/ダウンロード

ファイルをFTPアップロードするために必要な情報



■ Q1. IBM i を使ったファイルのアップロード/ダウンロード

ファイルをFTPアップロードするために必要な情報

①アップロード先の情報
IBMiのIPアドレス
ログインユーザー/パスワード

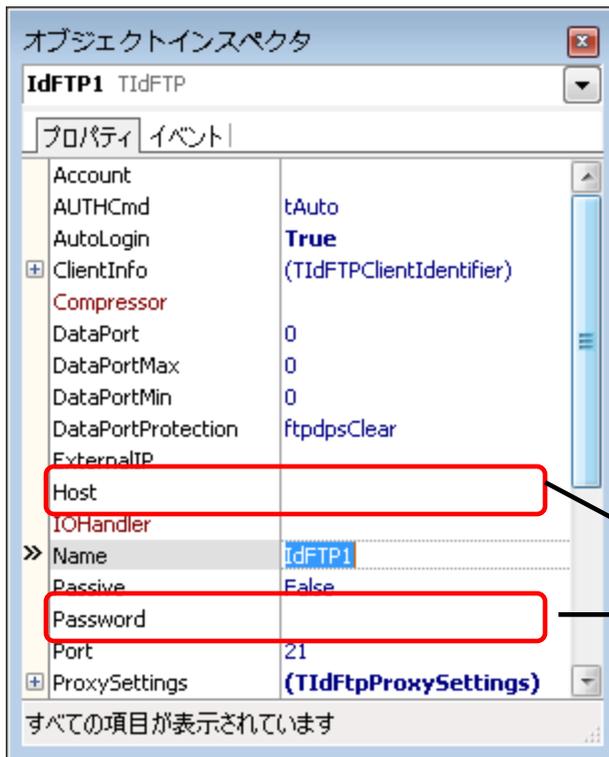
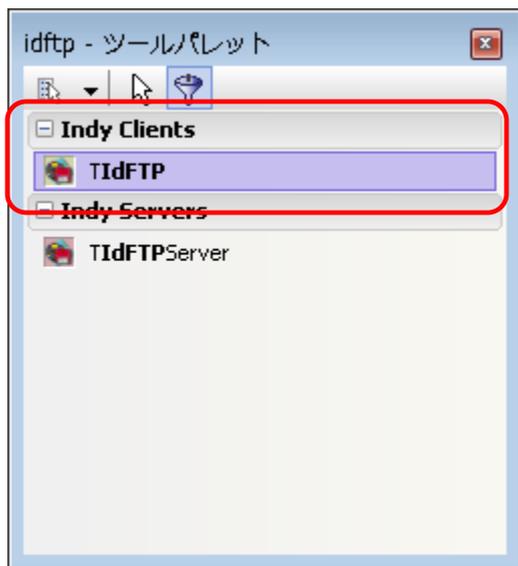
②アップロードする
PCファイル

The screenshot shows the IBM i FTP client interface. A red box highlights the login fields: IBM i (192.168.XX.XX), ユーザー (QSECOFR), and パスワード (*****). Another red box highlights the file selection area: アップロードファイル (C:\demo\プログラム\第6回\Session2\demo\サンプル\Sample.xls) and the save fields: 保存先ライブラリ (D4TECLIB) and 保存ファイル名 (SAMPLEXLS). A third red box highlights the 'アップロード' button. Callout lines connect these elements to the numbered text boxes on the left and bottom right.

③アップロード先の
ライブラリ名/ファイル名

■ Q1. IBM i を使ったファイルのアップロード/ダウンロード

IdFTPコンポーネント



接続情報
ホスト
ユーザー/パスワード

■ Q1. IBM i を使ったファイルのアップロード/ダウンロード

アップロードボタン押下処理

```
procedure TfrmQ1.btnUPLoadClick(Sender: TObject);
begin
  try
    try
      frmWait.show;           //転送画面を表示
      Application.ProcessMessages; //描画調整

      //IdFTPコンポーネントでIBM i へ接続
      IdFTP1.Quit;           //切断
      IdFTP1.Host := edtIBM.Text; //接続先IPを設定
      IdFTP1.Username := edtUSR.Text; //ログインユーザー名設定
      IdFTP1.Password := edtPWD.Text; //ログインパスワード設定

      IdFTP1.Connect; //FTP接続
      //アップロード処理
      //      *PCファイル      *アップロード先ライブラリ/ファイル
      IdFTP1.Put (edtUPFILE.Text, edtUPLIB.Text + '/' + edtUPPF.Text);
```



①アップロード先の情報
で設定した内容

②アップロードファイル
③アップロード先
で設定した内容でPUT

■ Q1. IBM i を使ったファイルのアップロード/ダウンロード

```
//アップロードした情報を管理ファイルに格納
```

```
with DataModule1.Table1 do
```

```
begin
```

```
  Append;
```

```
  FieldByName('UPLDLB').AsString := edtUPLIB.Text;
```

```
  FieldByName('UPLDPF').AsString := edtUPPF.Text;
```

```
  FieldByName('UPFLNM').AsString := ExtractFileName(edtUPFILE.Text);
```

```
  Post;
```

```
  Close;
```

```
  Open;
```

```
end;
```

```
IdFTP1.Quit; //FTP切断
```

```
finally
```

```
  frmWait.close; //転送画面を終了
```

```
end;
```

```
//完了メッセージ
```

```
showmessage('アップロード成功');
```

```
except
```

```
//例外発生時はエラーメッセージ
```

```
showmessage('アップロード失敗');
```

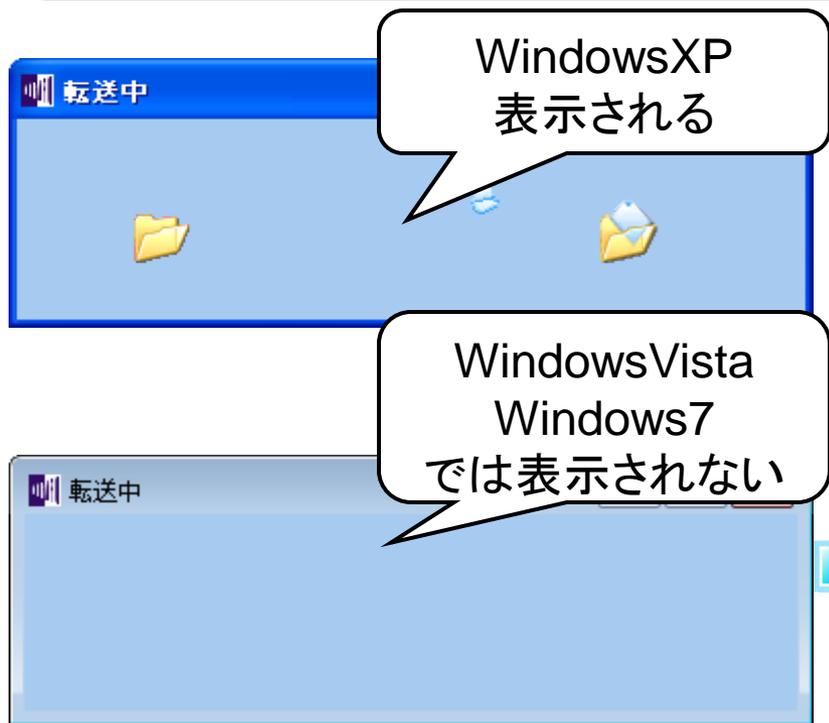
```
end;
```

```
end;
```

ダウンロードする為に
アップロード情報を
アップロード管理ファイル
にレコード追加
※サンプル仕様

■ Q1. IBM i を使ったファイルのアップロード/ダウンロード

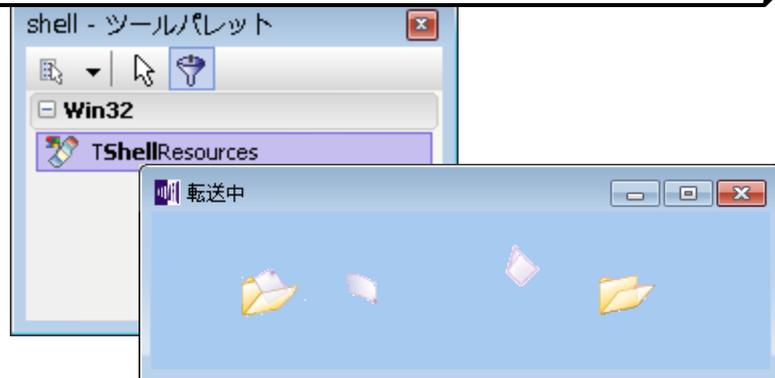
Windows Vista Windows7での
Animateコンポーネントのアニメーションについて



WindowsXP
表示される

WindowsVista
Windows7
では表示されない

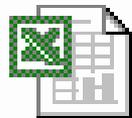
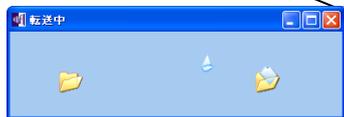
! Windows Vista からは
リソースの管理が変わっており、
**Shell32.dll にアニメーションに必要な
avi が含まれていない**ので表示できません。
Delphi2007以降は
TShellResourceコンポーネントを置く
ことで新しいアニメーションを表示できます。



■ Q1. IBM i を使ったファイルのアップロード/ダウンロード

IBM i のファイルをダウンロードする

FTP
ダウンロード



【Q1】 IBM i を使ってファイルをアップロード/ダウンロード

IBM i | 192.168.XX.XX

ユーザー | QSECOFR

パスワード | *****

アップロード ダウンロード

ライブラリ	ファイル	元ファイル
YOSLIB	SAMPLXLS	Sample.xls
YOSLIB	EXCEL	Sample.xls
YOSLIB	LOG	drwtsn32.log

ライブラリ | YOSLIB

ファイル名 | SAMPLXLS

ダウンロード先 | C:\¥デモプログラム¥第6回¥Session2¥デモサンプル¥Sample.xls

ダウンロード

名前

SAMPLE.xls



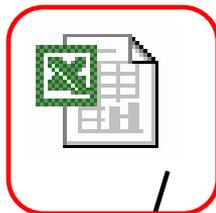
■Q1. IBM i を使ったファイルのアップロード/ダウンロード

ファイルをFTPダウンロードするために必要な情報

②ダウンロード元の
ライブラリ名/ファイル名



①ダウンロード元の情報
IBMiのIPアドレス
ログインユーザー/パスワード



③ダウンロード先の
PCファイル名

■ Q1. IBM i を使ったファイルのアップロード/ダウンロード

IBM i のファイルをダウンロードする

The screenshot shows a web interface for connecting to an IBM i system. It includes fields for IP address, user, and password, a table of files, and a download path field. Three yellow callout boxes provide instructions:

- ①ダウンロード元の情報
IBMiのIPアドレス
ログインユーザー/パスワード
- ②ダウンロード元の
ライブラリ名/ファイル名
- ③ダウンロード先の
PCファイル名

ライブラリ	ファイル	元ファイル
YOSLIB	SAMPLXLS	Sample.xls
YOSLIB	EXCEL	Sample.xls
YOSLIB	LOG	drwtsn32.log

ライブラリ: YOSLIB
ファイル名: SAMPLXLS
ダウンロード先: C:\¥デモプログラム¥第6回¥Session2¥デモサンプル¥Sample.xls

✓ ダウンロード

■ Q1. IBM i を使ったファイルのアップロード/ダウンロード

インターフェースの工夫

【Q1】 IBM i を使ってファイルをアップロード/ダウンロード

IBM i: 192.168.XX.XX
ユーザー: QSECOFR
パスワード: *****

アップロード ダウンロード

ライブラリ	ファイル	元ファイル
YOSLIB	SAMPLXLS	Sample.xls
YOSLIB	CEL	Sample.xls
YOSLIB	G	drwtsn32.log

ライブラリ: YOSLIB
ファイル名: SAMPLXLS

ダウンロード先: C:\¥デモプログラム¥第6回¥Session2¥デモサンプル¥Sample.xls

✓ ダウンロード

名前を付けて保存

OS (C:) の検索

整理 新しいフォルダー

名前 更新日時

名前	更新日時
Bin	2010/01/27 15:...
BootImage	2010/01/07 20:...
co405	2010/05/07 14:...
co407	2010/02/23 14:...
co411	2010/05/11 9:4...
co412	2010/05/10 18:...
CO413	2010/05/13 8:5...
D400自習セミナーセット	2010/05/13 11:...
dell	2010/01/20 13:...
Delphi	2010/03/02 11:...
drivers	2010/01/09 0:5...

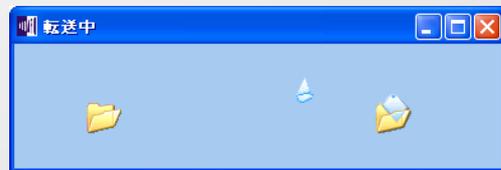
ファイル名 (F): SAMPLE.XLS
ファイルの種類 (T):

保存 (S) キャンセル

■ Q1. IBM i を使ったファイルのアップロード/ダウンロード

ダウンロードロードボタン押下処理

```
procedure TfrmQ1.btnDownloadClick(Sender: TObject);  
var  
  FStream : TFileStream; //ダウンロードファイル  
  Filesize : Integer;    //サイズチェック用  
begin  
  try  
    try  
      frmWait.show;           //転送画面を表示  
      Application.ProcessMessages; //描画調整  
  
      //IdFTPコンポーネントでIBM i へ接続  
      IdFTP1.Quit;           //切断  
      IdFTP1.Host := edtIBM.Text; //接続先IPを設定  
      IdFTP1.Username := edtUSR.Text; //ログインユーザー名設定  
      IdFTP1.Password := edtPWD.Text; //ログインパスワード設定  
      IdFTP1.Connect; //FTP接続  
  
      //ダウンロード先PCファイルを生成  
      FStream := TFileStream.Create(edtDLFILE.Text, fmCreate);
```



①ダウンロード元の情報
で設定した内容
アップロードと同様

③ダウンロード先の
PCファイルを生成

■ Q1. IBM i を使ったファイルのアップロード/ダウンロード

```
//ダウンロード元ファイルのサイズを取得
Filesize := IdFTP1. Size(edtDLLIB. Text + '/' + edtDLPF. Text);
//ファイルサイズが0ならエラー
if Filesize = 0 then
begin
  showmessage('ファイルなし');
  Abort;
end;
//ダウンロード処理
//      *ダウンロード元ファイル      *PCファイル
IdFTP1. Get(edtDLLIB. Text + '/' + edtDLPF. Text, FStream);
finally
  FStream. Free; //ファイル解放
end;
IdFTP1. Quit; //FTP切断
finally
  frmWait. close; //転送画面を終了
end;
showmessage('ダウンロード成功'); //完了メッセージ
except
  showmessage('ダウンロード失敗'); //例外発生時はエラーメッセージ
end;
end;
```

ダウンロード先の
ファイルをチェック

②ダウンロード元で
設定した情報から
ファイルをGET

■Q2. トレイアイコンを利用したアプリケーションの常駐

! 【質問】

Windowsのトレイ通知をアプリケーションで操作できますか？

✉ 【回答】

Delphi/400ではV2006よりTrayIconというトレイ操作専用のコンポーネントが提供されています。
これを利用するとWindowsのトレイ通知やトレイへの常駐といったアプリケーション動作が可能です。

■ Q2. トレイアイコンを利用したアプリケーションの常駐

トレイアイコンとはデスクトップの右下のタスクトレイに格納された常駐しているアプリケーションのアイコンです。

アプリケーションを終了させずに常駐させる使い方

ウィルス対策ソフトなどは
タスクトレイに常駐



必要なメッセージを通知したり、 時間のかかるプログラムをバックグラウンドで処理する使い方

更新の準備ができました。
これらの更新をインストールするには、ここをクリックしてください。

更新のインストール中...
更新のインストール中、作業を続行できます。
進行状況の詳細を表示するには、このアイコンをクリックしてください。

■ Q2. トレイアイコンを利用したアプリケーションの常駐

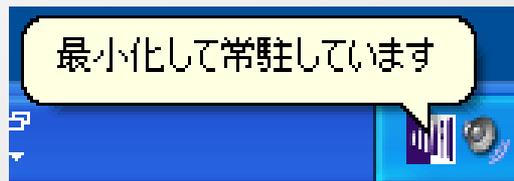
Delphi/400アプリケーションで
アプリケーションを終了させずに常駐させる使い方



■ Q2. トレイアイコンを利用したアプリケーションの常駐

画面のサイズ変更イベントで最小化されたらトレイアイコン化

```
procedure TfrmQ2.FormResize(Sender: TObject);
begin
  //サイズが変更されて最小化されたらトレイアイコンへ
  if Self.WindowState = wsMinimized then
  begin
    Self.Visible := False; //画面非表示
    TrayIcon1.Visible := True; //トレイアイコン表示
    TrayIcon1.BalloonHint := '最小化して常駐しています'; //メッセージ設定
    TrayIcon1.ShowBalloonHint; //メッセージ表示
  end;
end;
```



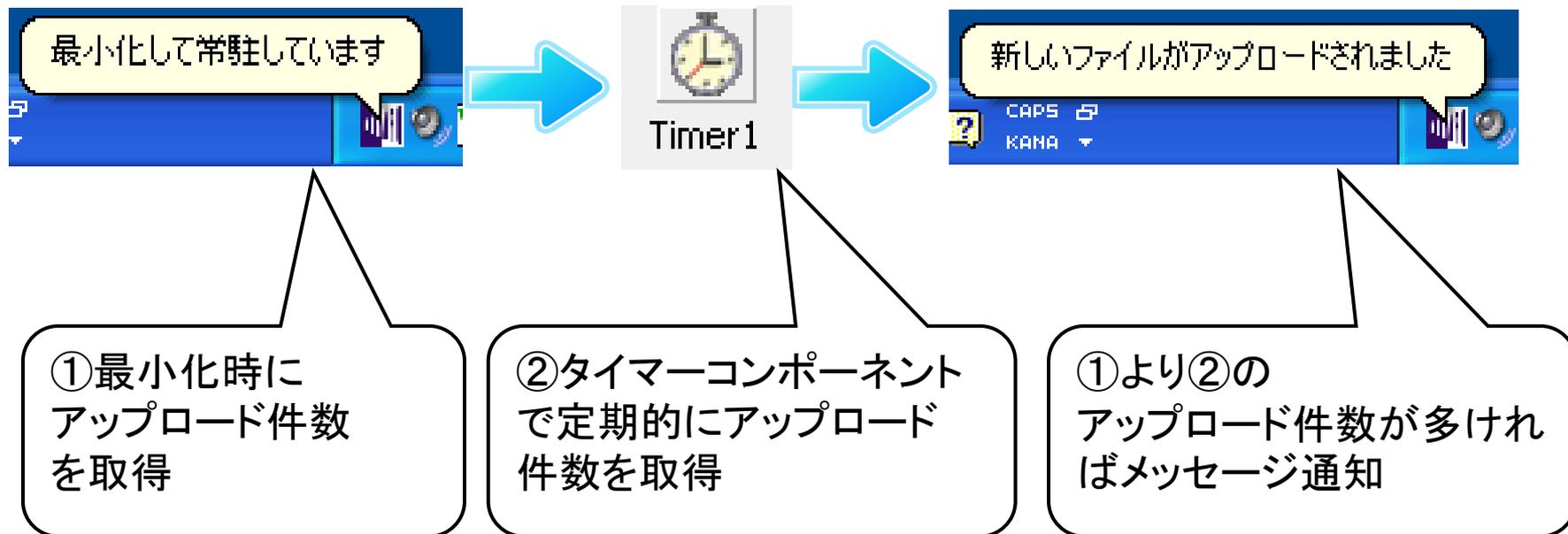
トレイアイコンをクリックされたら元の画面を表示

```
procedure TfrmQ2.TrayIcon1Click(Sender: TObject);
begin
  Self.WindowState := wsNormal; //画面サイズを通常に設定
  Self.Visible := True; //画面表示
  TrayIcon1.Visible := False; //トレイアイコン非表示
end;
```

最小化から
元のサイズに戻して
表示

■ Q2. トレイアイコンを利用したアプリケーションの常駐

Delphi/400アプリケーションで
常駐しながらメッセージを通知する使い方



■ Q2. トレイアイコンを利用したアプリケーションの常駐

画面のサイズ変更イベントで最小化されたらトレイアイコン化

```
procedure TfrmQ2.FormResize(Sender: TObject);
begin
  //サイズが変更されて最小化されたらトレイアイコンへ
  if Self.WindowState = wsMinimized then
  begin
    Self.Visible := False; //画面非表示
    TrayIcon1.Visible := True; //トレイアイコン表示
    TrayIcon1.BalloonHint := '最小化して常駐しています'; //メッセージ設定
    TrayIcon1.ShowBalloonHint; //メッセージ表示
    with DataModule1.Query1 do //アップロード管理ファイルの件数を取得
    begin
      Close;
      Open;
      //グローバルアップロード件数に格納 SELECT COUNT(*) CNT FROM UPLDWK
      PDLFileCount := FieldByName('CNT').AsInteger;
      Close;
    end;
    Timer1.Enabled := True; //アップロードを検知するために常駐タイマーをON
  end;
end;
```

```
//宣言部
private
{ Private 宣言 }
//グローバルアップロード件数 (保持用)
PDLFileCount : Integer;
```

アップロード検知のために最小化時にアップロード件数をグローバル変数に取得

アップロード検知はTimerコンポーネントを利用

■ Q2. トレイアイコンを利用したアプリケーションの常駐

タイマーイベント(アップロード検知処理)

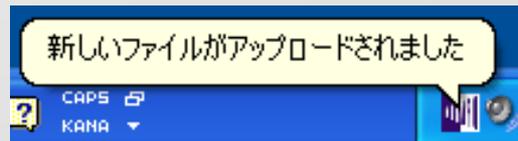


Timer1

```
procedure TfrmQ2.Timer1Timer(Sender: TObject);
```

```
var  
    iCount : Integer; //アップロード件数  
begin  
    with DataModule1 do  
    begin  
        Query1.Close;  
        Query1.Open; //アップロード管理ファイルの件数を取得  
        iCount := Query1.FieldByName('CNT').AsInteger; //ローカルアップロード件数に格納  
        Query1.Close;  
        //グローバルアップロード件数と比較してレコードが増えていればメッセージ表示  
        if PDLFileCount < iCount then  
        begin  
            TrayIcon1.BalloonHint := '新しいファイルがアップロードされました'; //メッセージ設定  
            TrayIcon1.ShowBalloonHint; //メッセージ表示  
            PDLFileCount := iCount; //グローバルアップロード件数更新  
            Table1.Close; //ファイルの読み直し  
            Table1.Open;  
            Table1.Last; //新レコード(最終レコード)へ移動  
        end;  
    end;  
end;  
end;
```

タイマー時間ごとに
アップロード件数が増えてい
ないかチェック



■ Q2. トレイアイコンを利用したアプリケーションの常駐

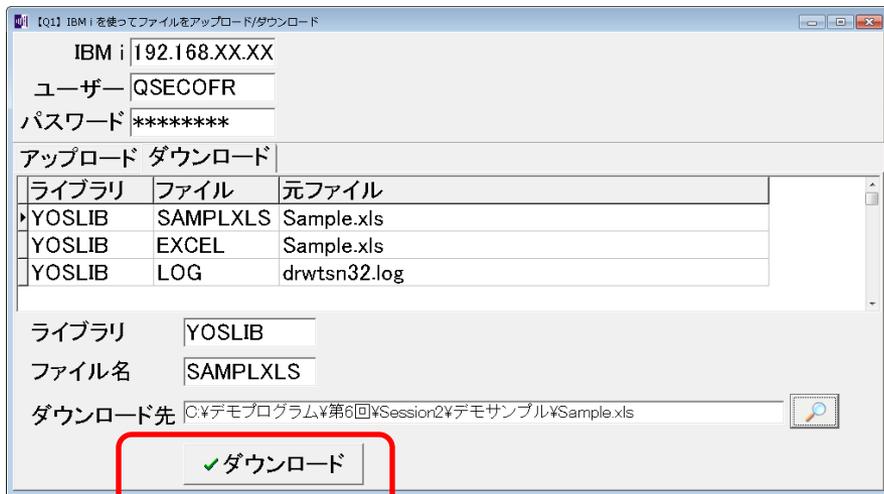
トレイアイコンをクリックされたら元の画面を表示

```
procedure TfrmQ2.TrayIcon1Click(Sender: TObject);  
begin  
  Self.WindowState := wsNormal; //画面サイズを通常に設定  
  Self.Visible := True; //画面表示  
  TrayIcon1.Visible := False; //トレイアイコン非表示  
  Timer1.Enabled := False; //常駐タイマーOFF  
end; n
```

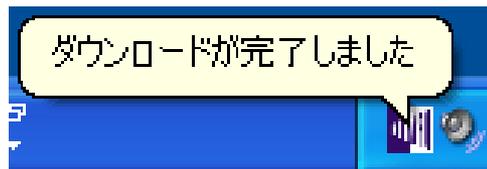
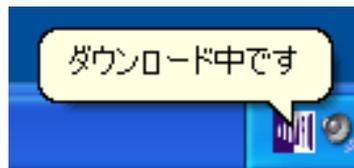
トレイアイコン終了
の際にタイマーも解除

■ Q2. トレイアイコンを利用したアプリケーションの常駐

Delphi/400アプリケーションで
時間のかかるプログラムをバックグラウンドで処理する使い方



ダウンロード処理中は
常駐アプリケーションへ



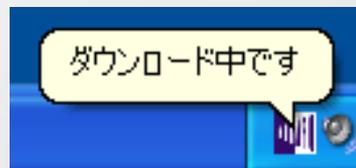
■ Q2. トレイアイコンを利用したアプリケーションの常駐

ダウンロードロードボタン押下処理

```
procedure TfrmQ2.btnDownloadClick(Sender: TObject);
var
  FStream : TFileStream; //ダウンロードファイル
  Filesize : Integer;    //サイズチェック用
begin
  try
    Self.Visible      := False;      //画面を非表示
    TrayIcon1.Visible := True;       //トレイアイコンを表示
    TrayIcon1.BalloonHint := 'ダウンロード中です'; //メッセージ設定
    TrayIcon1.ShowBalloonHint;      //メッセージを表示

    //IdFTPコンポーネントでIBM i へ接続
    IdFTP1.Quit; //切断
    IdFTP1.Host := edtIBM.Text; //接続先IPを設定
    IdFTP1.Username := edtUSR.Text; //ログインユーザー名設定
    IdFTP1.Password := edtPWD.Text; //ログインパスワード設定

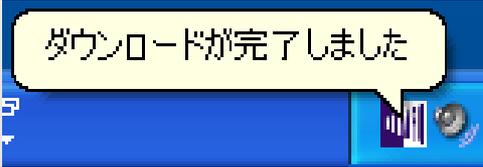
    IdFTP1.Connect; //FTP接続
    //ダウンロード先PCファイルを生成
    FStream := TFileStream.Create(edtDLFILE.Text, fmCreate);
```



待ち受け画面の
代わりにトレイアイコン
でメッセージ表示

■Q2. トレイアイコンを利用したアプリケーションの常駐

```
try
  Filesize := IdFTP1.Size(edtDLLIB.Text + '/' + edtDLPF.Text); //ダウンロード元ファイルのサイズを取得
  if Filesize = 0 then //ファイルサイズが0ならエラー
    begin
      showmessage('ファイルなし');
      Abort;
    end;
  //ダウンロード処理
  //          *ダウンロード元ファイル          *PCファイル
  IdFTP1.Get(edtDLLIB.Text + '/' + edtDLPF.Text, FStream);
finally
  FStream.Free;
end;
IdFTP1.Quit; //FTP切断
//完了メッセージ通知
TrayIcon1.BalloonHint := 'ダウンロードが完了しました';
TrayIcon1.ShowBalloonHint;
except
  //例外発生時はエラーメッセージ通知
  TrayIcon1.BalloonHint := 'ダウンロードが失敗しました';
  TrayIcon1.ShowBalloonHint;
end;
end;
```



待ち受け画面の
代わりにトレイアイコン
でメッセージ表示

■Q3. リモートデバッガで運用PCのデバッグ調査

!【質問】

運用PCでのエラー原因を特定する方法はありませんか？

✉【回答】

特定のPCで発生するエラーについて原因調査する場合調査方法は様々です。

Delphi/400で付属している「RemoteDebugger」というツールを使用すると開発環境から運用PCのアプリケーションをデバッグ実行することが可能です。

■Q3. リモートデバッガで運用PCのデバッグ調査

特定のPCでしか発生しない「あの現象」を調査する



■Q3. リモートデバッガで運用PCのデバッグ調査

①ソース机上チェックで問題点を洗い出す

単純な問題であれば発見できますが、非常に労力がかかります。
また原因がロジックレベルでないと判明できるとも……

②ログやメッセージを埋め込んだEXEを作って調査する

原因のポイントを想定しつつ、ログなどの処理を埋め込んでいきます。
原因を特定していくためには調査用のEXEを作成する労力がかかりますが
多くの場合、非常に有効です。

③Delphi/400の開発環境をインストール/構築してデバッグする

理想ではありますが、インストールやプロジェクト環境構築に非常に
労力がかかります。ライセンスの面でも現実的ではありません。

■Q3. リモートデバッガで運用PCのデバッグ調査

■もうひとつの選択肢■

リモートデバッガを使うと開発環境から
運用PCの端末でデバッグ実行することができます。

開発PC



運用PC



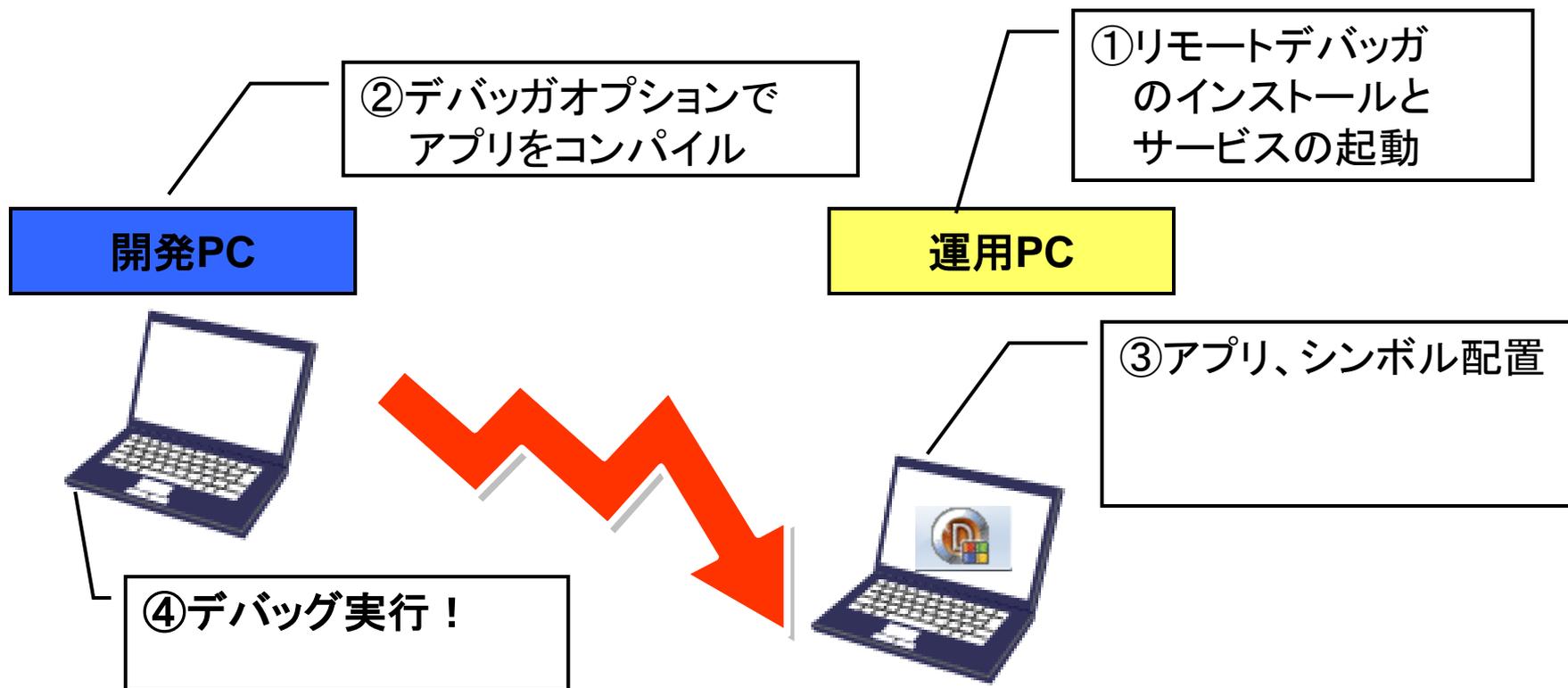
リモートでデバッグ実行調査できる！



Error
あの現象

■Q3. リモートデバッガで運用PCのデバッグ調査

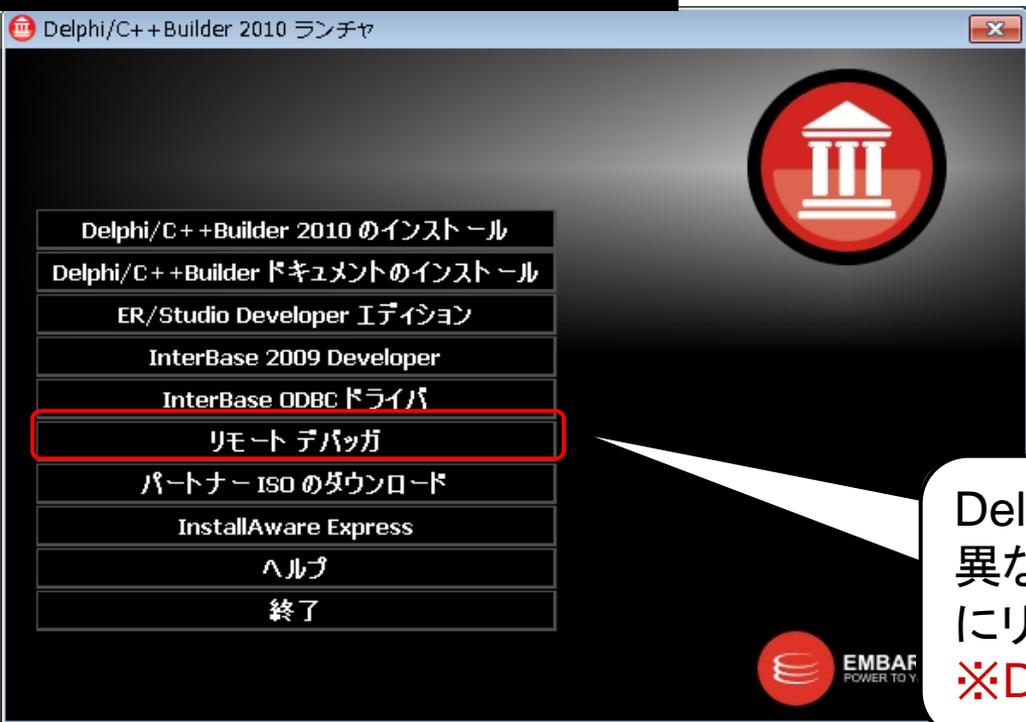
リモートデバッガの使い方



■ Q3. リモートデバッガで運用PCのデバッグ調査

① リモートデバッガのインストールとサービスの起動

V2010 インストーラ



V7 インストーラ



Delphiのバージョンによって名称等異なりますがDelphiのインストーラにリモートデバッガが含まれています。
※Delphi2005だけは付属していません。

■ Q3. リモートデバッガで運用PCのデバッグ調査

① リモートデバッガのインストールとサービスの起動

rmtdbgXXX.exe



“インストールフォルダ¥rmtdbgXXX.exe” -listen
と実行してサービスを起動します。
XXX部分はバージョンによって異なります。

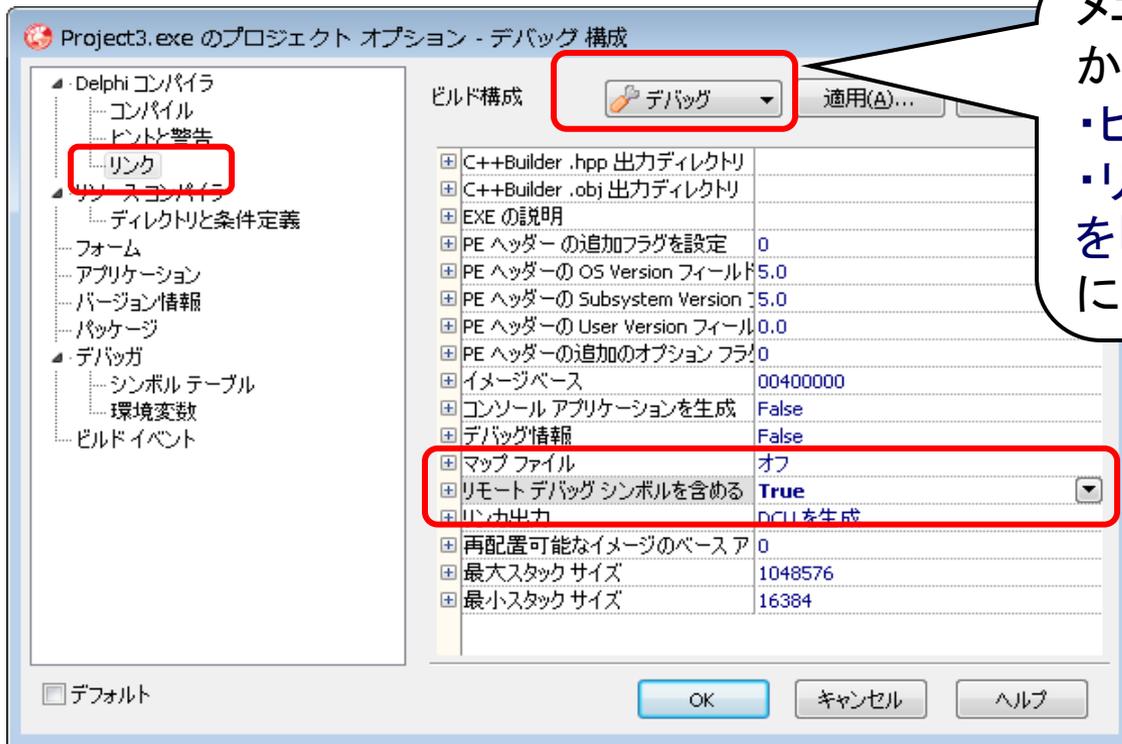


サービスが起動すると
タスクトレイに虫のアイコン
が表示されます。
ウイルスではありません！

■ Q3. リモートデバッガで運用PCのデバッグ調査

② デバッガオプションでアプリをコンパイル

V2009 V2010 資料



メニューの[プロジェクト|オプション]から[リンク]タブで左画面のように
・ビルド構成を「デバッグ」
・リモートデバッグシンボルを含めるを「True」に設定してコンパイルします。

コンパイル生成物

名前

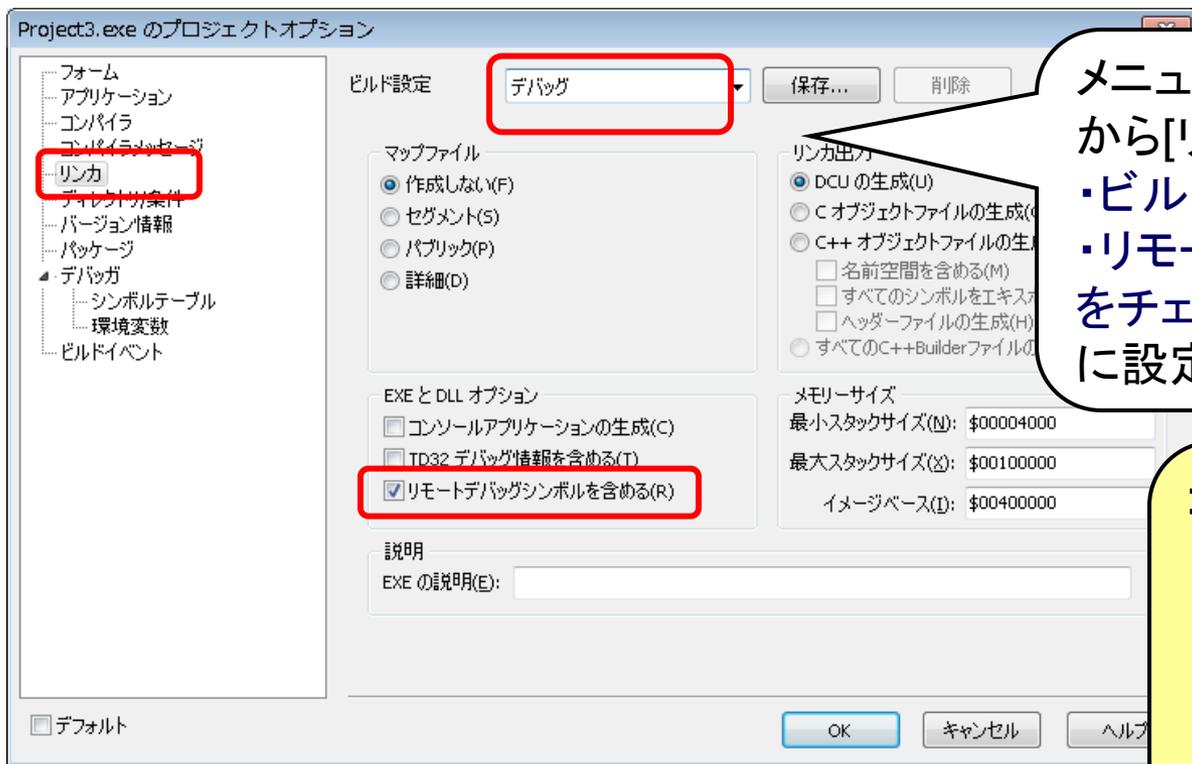
MigaroSample.exe
MigaroSample.rsm

シンボル
ファイル
(.rsm)

■ Q3. リモートデバッガで運用PCのデバッグ調査

② デバッガオプションでアプリをコンパイル

V2006 V2007 資料



メニューの[プロジェクト|オプション]から[リンク]タブで左画面のように・ビルド構成を「デバッグ」・リモートデバッグシンボルを含めるをチェックに設定してコンパイルします。

コンパイル生成物

名前

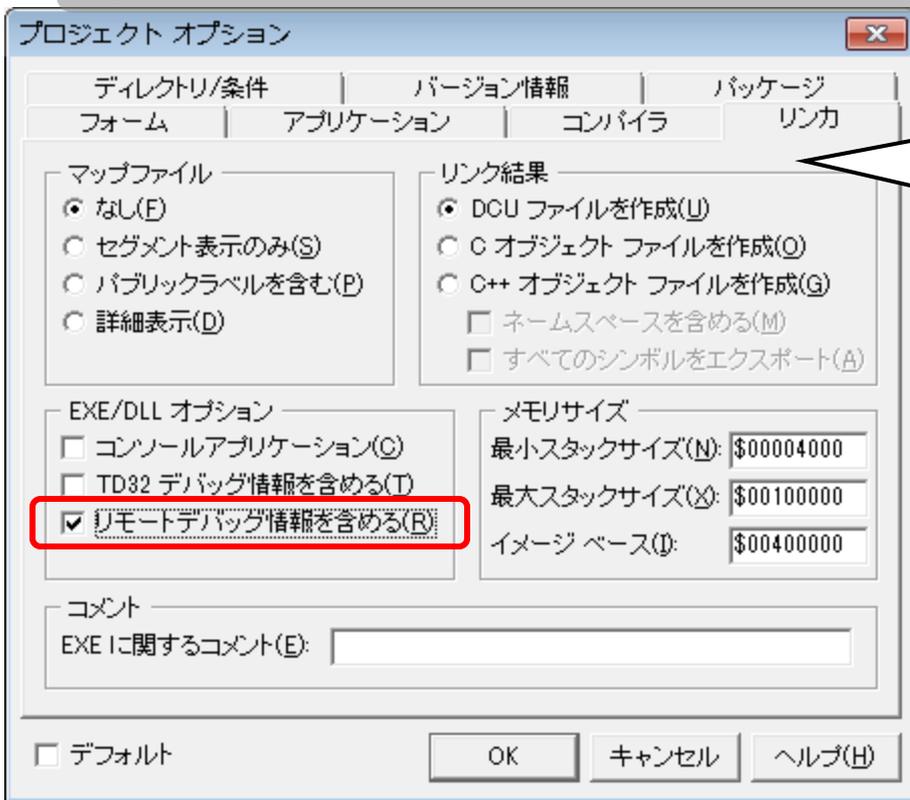
MigaroSample.exe
MigaroSample.rsm

シンボル
ファイル
(.rsm)

■ Q3. リモートデバッガで運用PCのデバッグ調査

② デバッガオプションでアプリをコンパイル

V5 V6 V7 資料



メニューの[プロジェクト|オプション]から[リンク]タブで左画面のように・リモートデバッグ情報を含めるをチェックに設定してコンパイルします。

コンパイル生成物

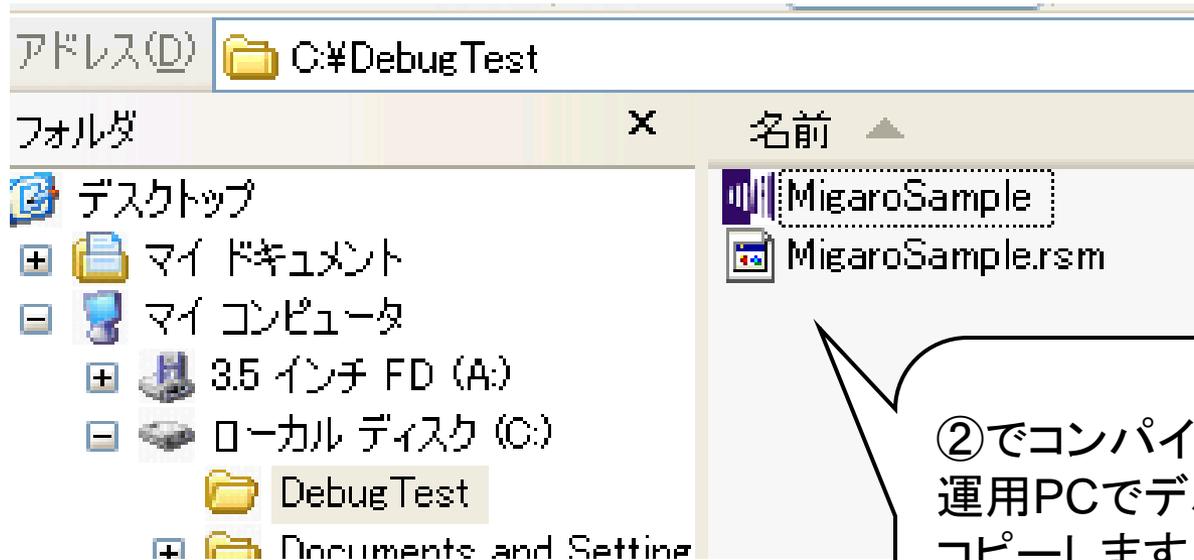
名前

MigaroSample.exe
MigaroSample.rsm

シンボル
ファイル
(.rsm)

■Q3. リモートデバッガで運用PCのデバッグ調査

③アプリ、シンボル配置



②でコンパイルした生成ファイルを運用PCでデバッグしたいフォルダにコピーします。

- ・アプリケーション(.exeや.dll)
- ・シンボルファイル(.rsm)

■ Q3. リモートデバッガで運用PCのデバッグ調査

④ デバッグ実行！

V2006 V2007 V2009 V2010 資料

メニューの[実行|プロセスのロード]から[リモート]タブで下記を設定

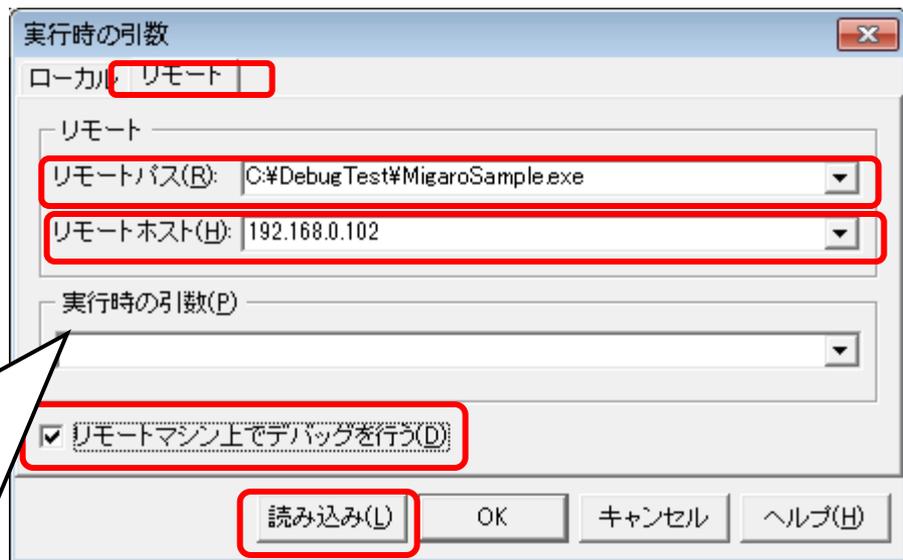
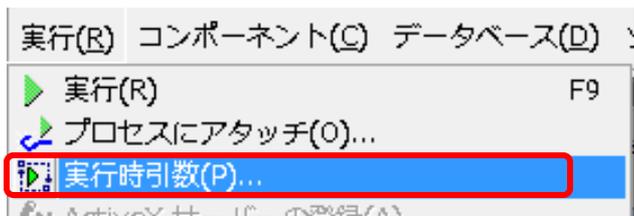
- 【リモートパス】
デバッグ先のPC上のアプリケーションパス
※絶対パスです。
- 【リモートホスト】
デバッグ先のPCのIPまたはホスト名

読み込みボタンで
リモートデバッグを実行

■ Q3. リモートデバッガで運用PCのデバッグ調査

④ デバッグ実行！

V5 V6 V7 資料



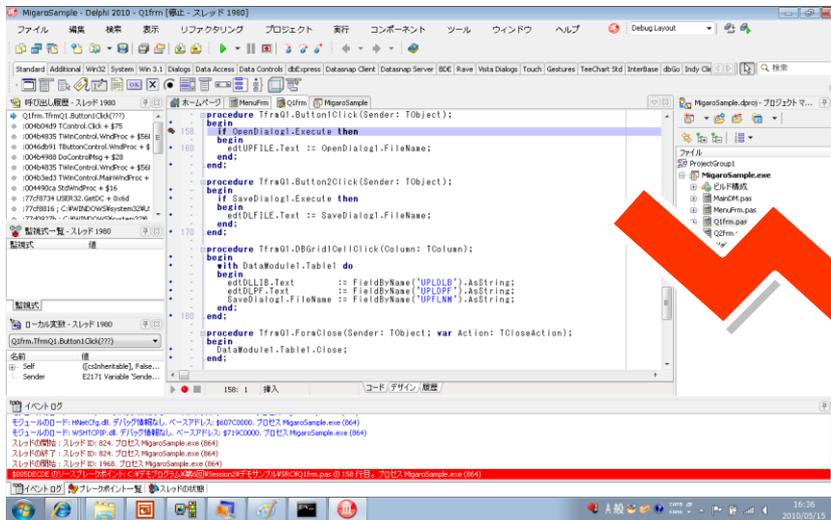
メニューの[実行|実行時引数]
から[リモート]タブで下記を設定
【リモートパス】
デバッグ先のPC上のアプリケーションパス
※絶対パスです。
【リモートホスト】
デバッグ先のPCのIPまたはホスト名
【リモートマシン上でデバッグを行う】
チェック

読み込みボタンで
リモートデバッグを実行

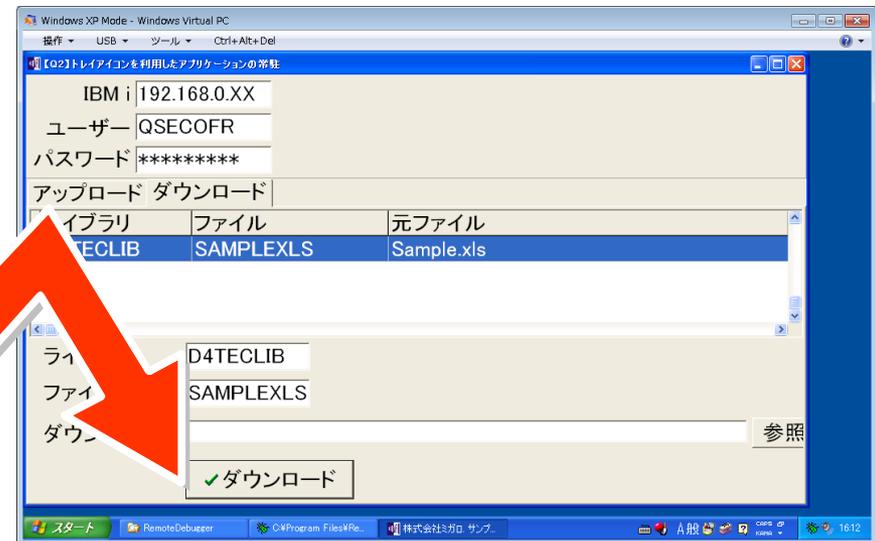
■ Q3. リモートデバッガで運用PCのデバッグ調査

④ デバッグ実行！

開発PC



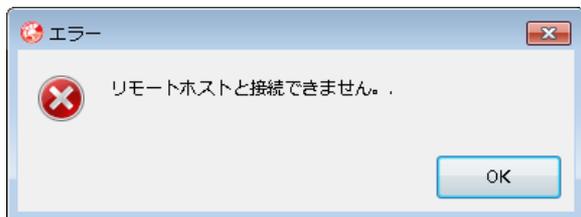
運用PC



運用PCで実行しながら開発環境でデバッグ調査が可能に！

■Q3. リモートデバッガで運用PCのデバッグ調査

補足



運用PCにリモート接続ができない場合は次のような点を確認してください。

- ・サービスが起動しているか
- ・シンボルファイルが配布されているか
- ・Windowsファイアウォールやウィルス対策ソフトで規制されていないか

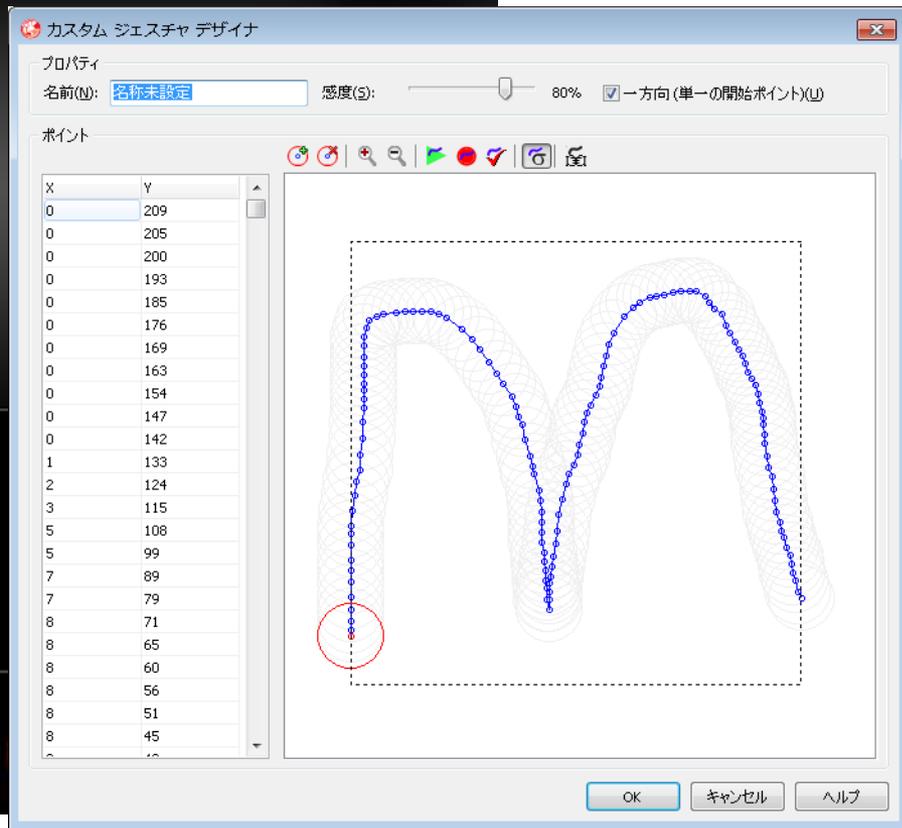
Delphiのアップデートなどがあたっている場合
それに対応したリモートデバッガのPTFなども必要です。

<V2006><http://cc.embarcadero.com/item/25608>

<V2007><http://cc.embarcadero.com/item/24995>

<V2010><http://cc.embarcadero.com/item/27494>

■ Q4 Delphi/400 V2010 新機能紹介 - ジェスチャアプリケーション作成 -



■ Q4 Delphi/400 V2010 新機能紹介 - ジェスチャアプリケーション作成 -

ジェスチャアプリケーションとは

ジェスチャはタッチパネルなどのタッチにおける指の動作の軌跡などを検出してイベントを発生させるものです。
もちろんマウスのフォーカスで同様の動作も可能です。

iPhoneなどでよく見られるアプリケーションが
タッチ操作で行うジェスチャー動作に対応
したアプリケーション。



■ Q4 Delphi/400 V2010 新機能紹介 - ジェスチャアプリケーション作成 -

ジェスチャインターフェース

V2010ではこういった「キーボードレス」のアプリケーションを作成するためのコンポーネントやインターフェースが実装されています。

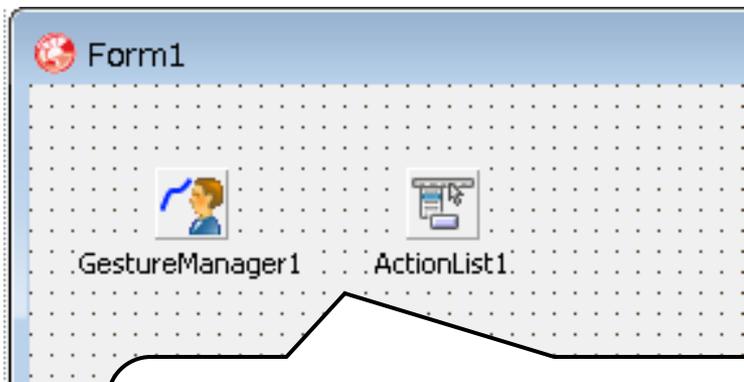
The screenshot displays two windows from the Delphi IDE. The left window, titled 'Form1', shows a grid-based form with a 'GestureManager1' component placed on it. A callout box points to this component with the text 'ジェスチャーを管理するGestureManager'. The right window, titled 'カスタム ジェスチャ デザイナー' (Custom Gesture Designer), shows a gesture design interface. It includes a 'プロパティ' (Properties) section with '名前(例):' and '感度(S):' (80%), and a 'ポイント' (Points) list. The list contains X and Y coordinates for 16 points. A callout box points to the 'TouchKeyboard' component at the bottom of the form with the text 'タッチで動作するTouchKeyboard'. The gesture designer window shows a blue dashed line forming a stylized 'M' shape on a grid.

X	Y
0	209
0	205
0	200
0	193
0	185
0	176
0	169
0	163
0	154
0	147
0	142
1	133
2	124
3	115
5	108
5	99
7	89
7	79
8	71
8	65
8	60
8	56
8	51
8	45

■ Q4 Delphi/400 V2010 新機能紹介

－ ジェスチャアプリケーション作成 －

ジェスチャ動作を実装する①



GestureManagerコンポーネントと
ActionListコンポーネントを配置して
GestureManagerをダブルクリック

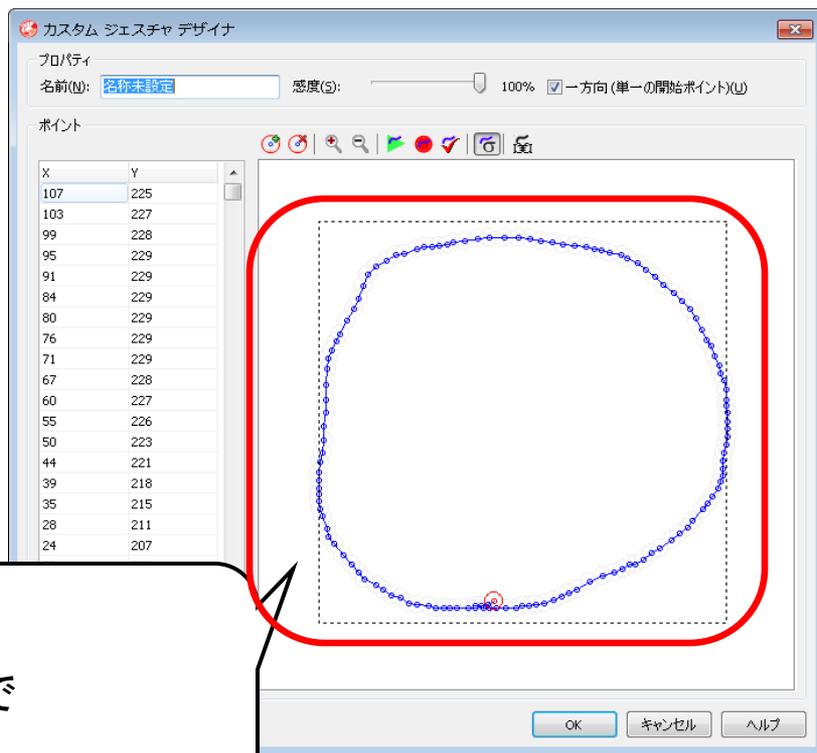
■ Q4 Delphi/400 V2010 新機能紹介

－ ジェスチャアプリケーション作成 －

ジェスチャ動作を実装する②



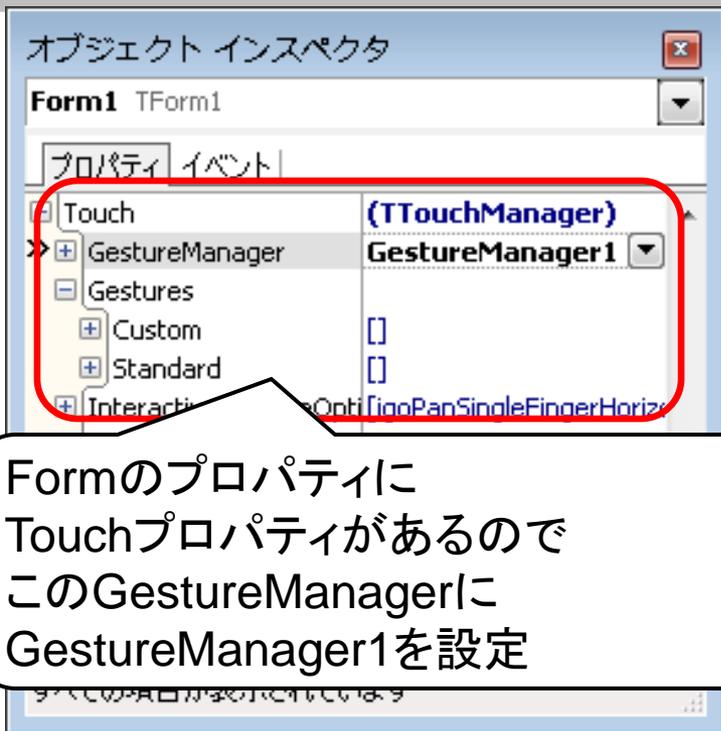
作成ボタンを押すと
「ジェスチャデザイナ」が起動されるので
自由なジェスチャを作成して登録



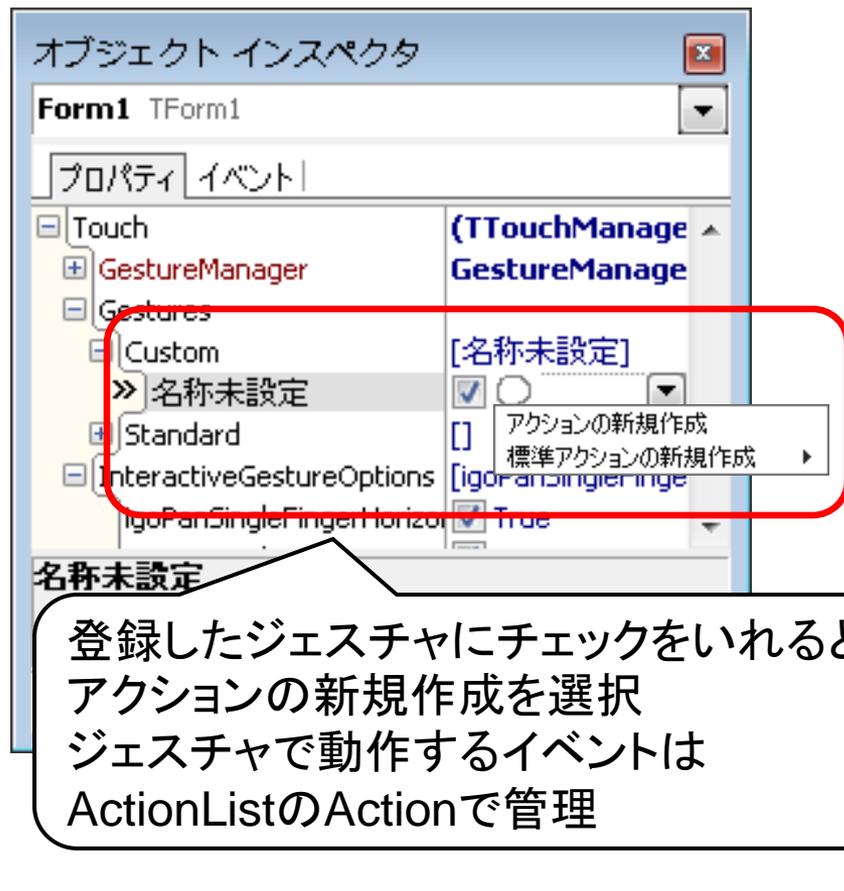
■ Q4 Delphi/400 V2010 新機能紹介

- ジェスチャアプリケーション作成 -

ジェスチャ動作を実装する③



Formのプロパティに
Touchプロパティがあるので
このGestureManagerに
GestureManager1を設定

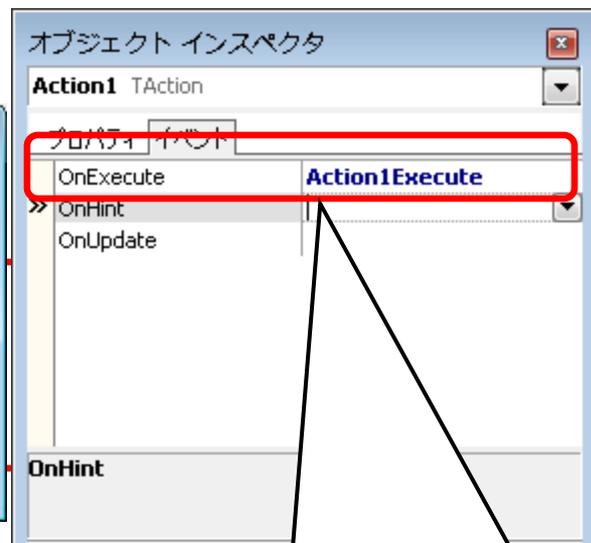
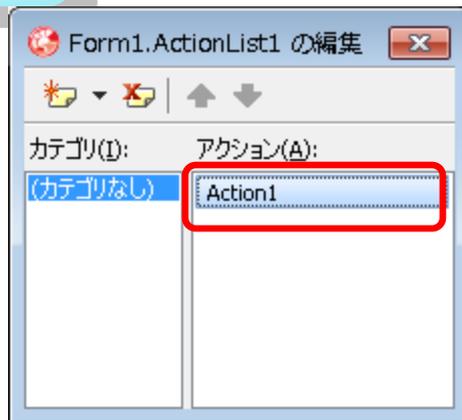
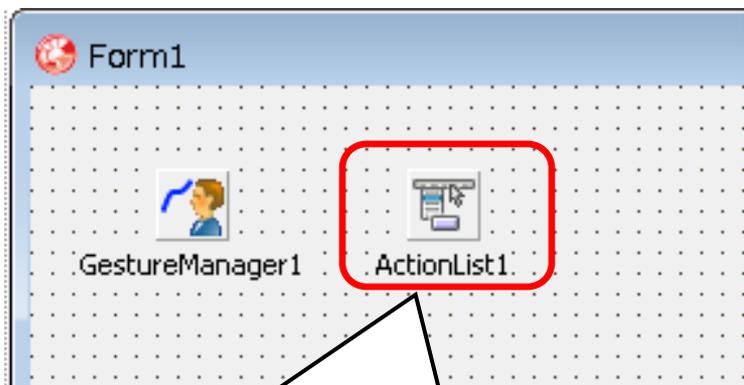


登録したジェスチャにチェックをいれると
アクションの新規作成を選択
ジェスチャで動作するイベントは
ActionListのActionで管理

■ Q4 Delphi/400 V2010 新機能紹介

- ジェスチャアプリケーション作成 -

ジェスチャ動作を実装する④



ActionListをダブルクリックして
新規作成したActionを確認

OnExecuteイベントに記述した
プログラムがジェスチャイベントで
呼び出されます

■ Q4 Delphi/400 V2010 新機能紹介

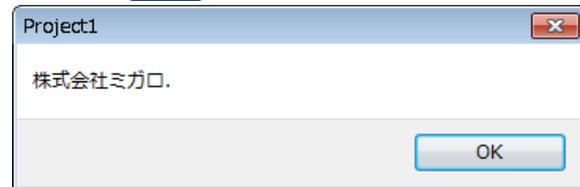
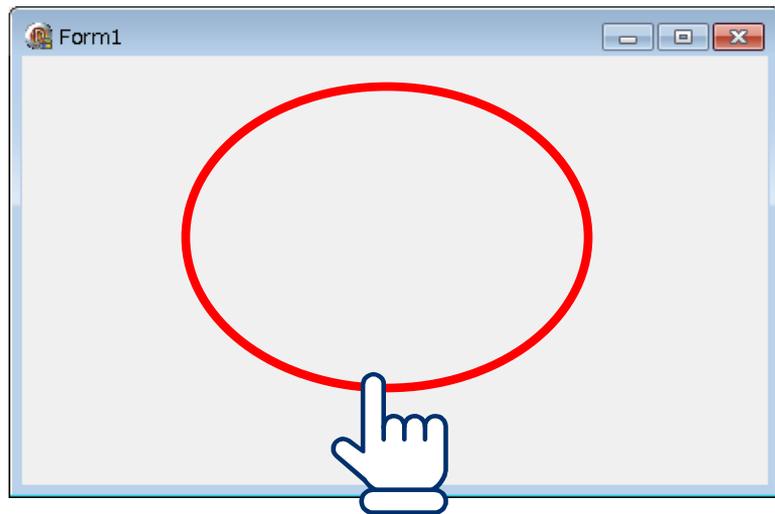
– ジェスチャアプリケーション作成 –

ジェスチャ動作を実装する⑤

```
procedure TForm1.Action1Execute(Sender: TObject);  
begin  
  ShowMessage('株式会社ミガロ。');  
end;
```

プログラムを記述して実行！

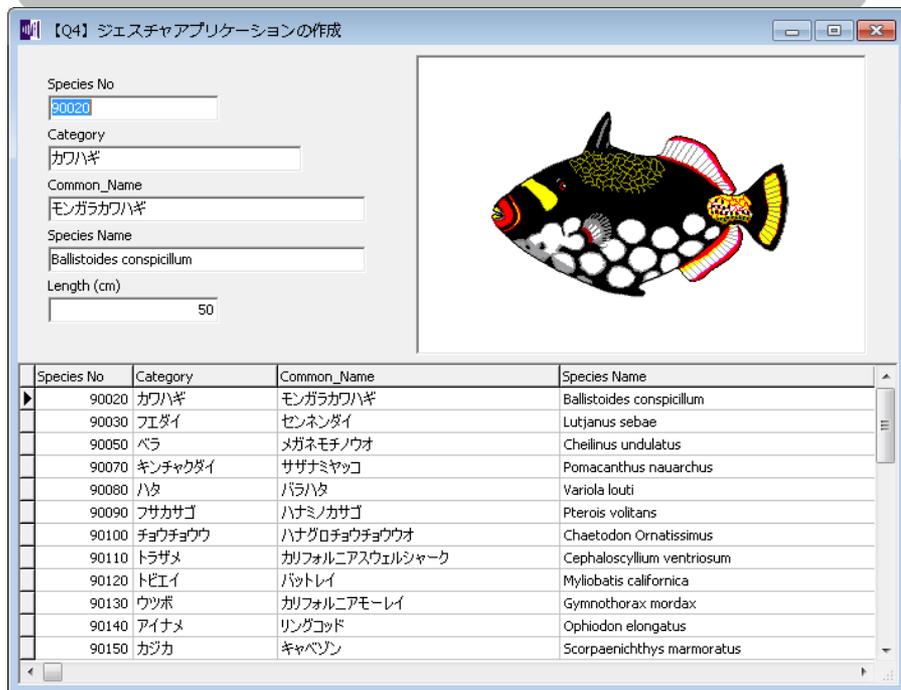
ジェスチャはOnGestureイベントでハンドリングもできますがActionで処理すると作成が容易です。



■ Q4 Delphi/400 V2010 新機能紹介

– ジェスチャアプリケーション作成 –

既存アプリへの応用実装



ジェスチャ設計

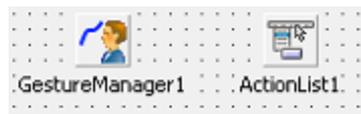
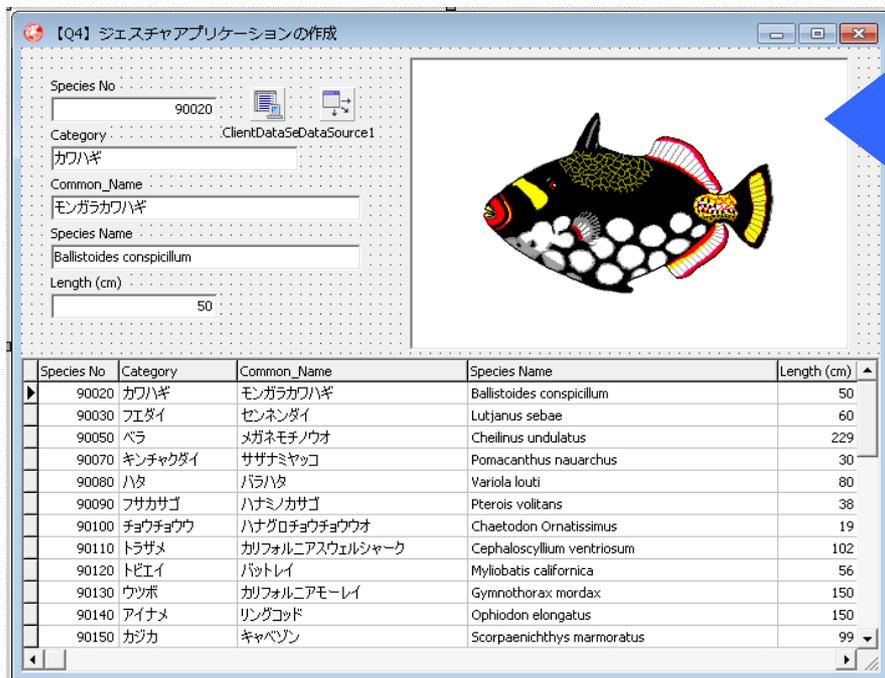
次のデータ

前のデータ

最初のデータ 最後のデータ データ開閉

■ Q4 Delphi/400 V2010 新機能紹介 - ジェスチャアプリケーション作成 -

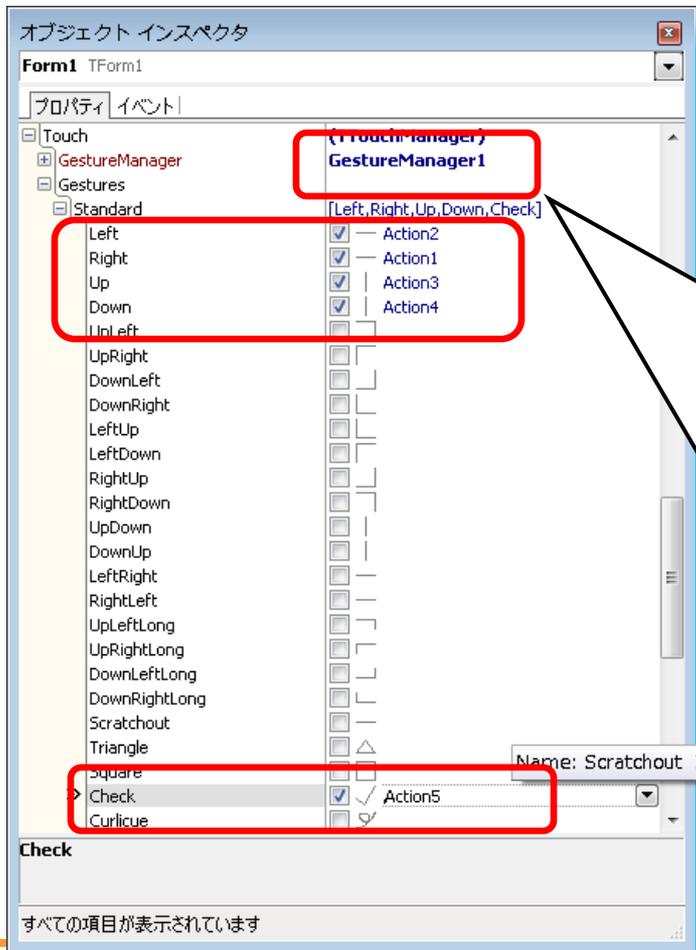
既存アプリへの応用実装①



データを表示している画面に
GestureManagerとActionListを
設定

■ Q4 Delphi/400 V2010 新機能紹介

– ジェスチャアプリケーション作成 –



既存アプリへの応用実装②

FormのGestureManagerにGestureManager1を設定

Standardにある既定のジェスチャにチェックを入れてActionを新規作成

Right : Action1
Left : Action2
Up : Action3
Down : Action4
Check : Action5

■ Q4 Delphi/400 V2010 新機能紹介 - ジェスチャアプリケーション作成 -

既存アプリへの応用実装③



次のデータ

```
procedure TForm1.Action1Execute(Sender: TObject);  
begin  
    //次のデータへ  
    ClientDataSet1.Next;  
end;
```



前のデータ

```
procedure TForm1.Action2Execute(Sender: TObject);  
begin  
    //前のデータへ  
    ClientDataSet1.prior;  
end;
```

■ Q4 Delphi/400 V2010 新機能紹介 - ジェスチャアプリケーション作成 -

既存アプリへの応用実装④

最初のデータ

```
procedure TForm1.Action3Execute(Sender: TObject);  
begin  
    //最初のデータへ  
    ClientDataSet1.First;  
end;
```

最後のデータ

```
procedure TForm1.Action4Execute(Sender: TObject);  
begin  
    //最後のデータへ  
    ClientDataSet1.Last;  
end;
```

■ Q4 Delphi/400 V2010 新機能紹介 - ジェスチャアプリケーション作成 -

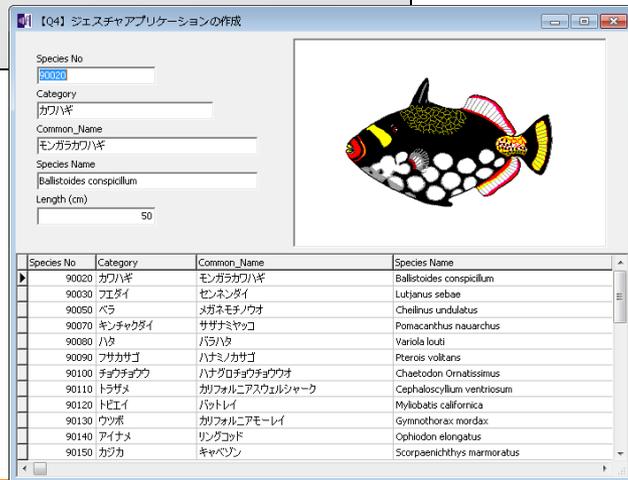
既存アプリへの応用実装⑤



データ開閉

```
procedure TForm1.Action5Execute(Sender: TObject);  
begin  
    //データを閉じて終了  
    ClientDataSet1.Active := not(ClientDataSet1.Active);  
end;
```

完成です



テクニカルサポートからのお知らせ

■テクニカルサポートからのお知らせ①

Delphi/400メンテナンス専用ページでは
2007年2008年のテクニカルセミナーの資料 サンプルPGMを公開しています

MIGARO > メンテナンスプログラム > Delphi/400 V2009 開発版 > Tips&Download>テクニカルセミナー資料 - Microsoft Internet Explorer

アドレス http://www.migaro.co.jp/contents/maintenance/D400DEV/VC/techseminar/index.html

Delphi/400 Version 2009 開発版

メンテナンスプログラム専用ページ

ダウンロード 修正・変更履歴 サンプルプログラム Tips & Download CD販売申込み

日頃Delphi/400をはじめとする弊社製品をご活用いただき、御社のために、弊社で培った開発ノウハウをご紹介させていただいている『ミガロ、テクニカルサポート』。こちらより過去のセッション資料をご覧いただき、ダウンロードしていただけます。

第1回 (2007年)	Delphi/400開発ノウハウお教えします ～現場で培った開発手法一挙公開～	※
サンプルプログラムダウンロード	Delphi/400でこんなに簡単！ GUI開発デモセッション ～既存RPG活用術～	※
	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	※
第2回 (2008年)	開発の幅を拓きましょう！ Delphi/400で始めるWEBアプリ入門	※
サンプルプログラムダウンロード	きっと役立つ！ ミガロ開発TIPS&効率化ツールのご紹介	※
	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	※
第3回 (2008年)	コンポーネント有効活用で開発効率向上！ ～コンポーネント活用テクニックのご紹介～	※
サンプルプログラムダウンロード	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 3.5	※

http://www.migaro.co.jp/contents/maintenance/common/techseminar/01/Mi...

アドレス http://www.migaro.co.jp/contents/maintenance/common/techseminar/01/Migar

Delphi/400 Technical Seminar

【セッションNo.2】

Delphi/400開発ノウハウお教えします
～現場で培った開発手法一挙公開～

完了

ファイルのダウンロード

このファイルを開くか、または保存しますか？

名前: MigaroTechPr02.tch
種類: LZH ファイル, 5.19 MB
発信元: www.migaro.co.jp

開く(O) 保存(S) キャンセル

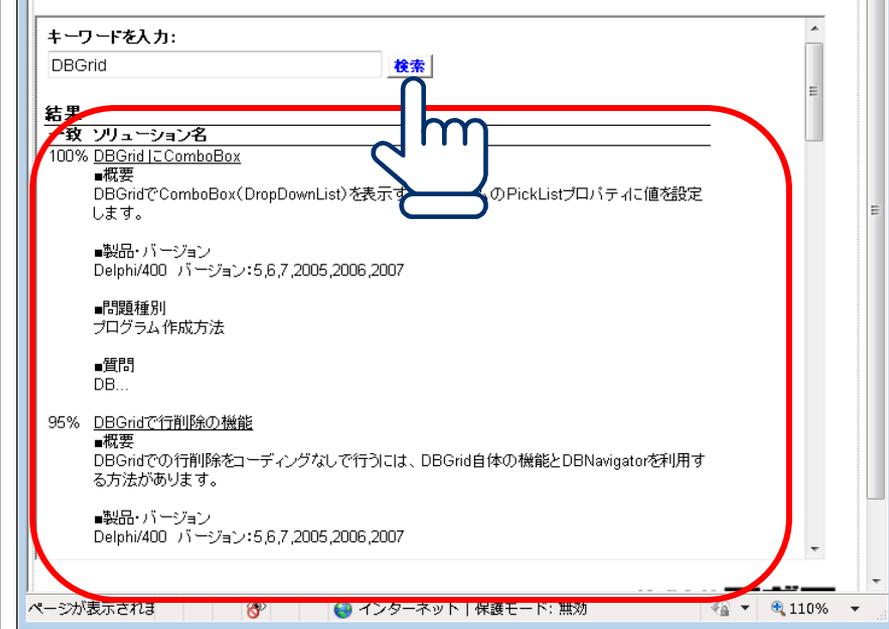
この種類のファイルであれば常に警告する(W)

インターネットのファイルは役に立ちますが、ファイルによってはコンピュータに問題を起すものもあります。発信元が信頼できない場合は、このファイルを開いたり保存したりしないでください。危険性の説明

■テクニカルサポートからのお知らせ②

Delphi/400メンテナンス専用ページの「お問い合わせ検索」のソリューションも是非ご活用ください！

1400件以上公開 しています

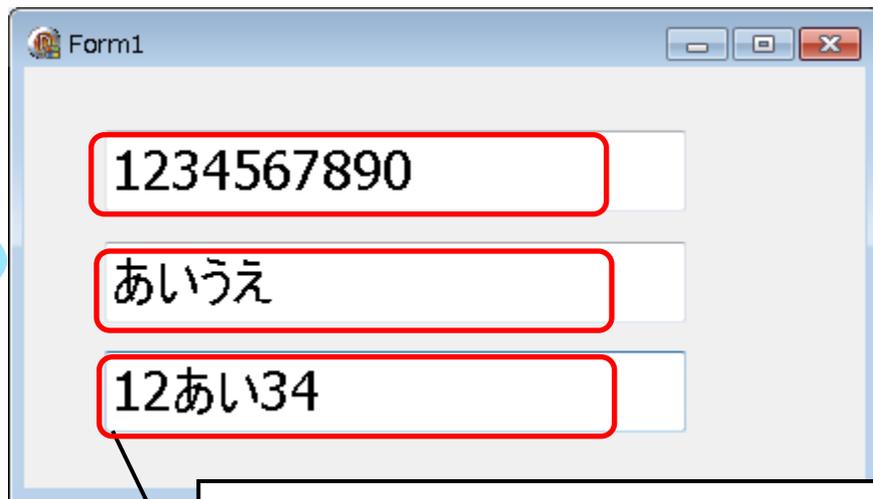
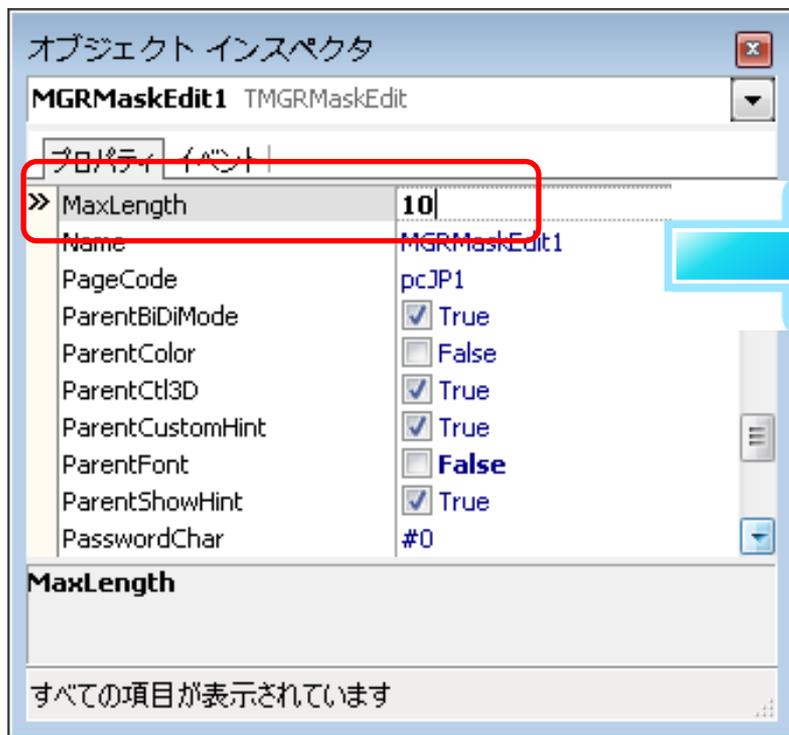


おまけ資料

■おまけ資料

Delphi/400 V2009V2010用MGRMaskEditコンポーネント

※ソースを含めV2009～のシフト文字制御の参考のために
お使いください。



シフトイン / シフトアウトを
考慮した桁数制御などが
可能なEditです。

■おまけ資料

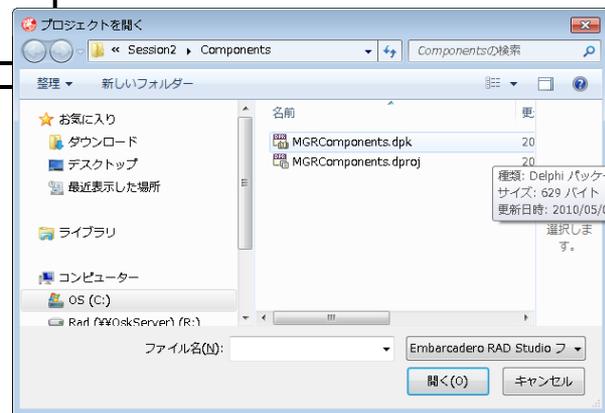
Delphi/400 V2009V2010用MGRMaskEditコンポーネント

導入手順

- ① MECSUtilsユニットをダウンロードして
サンプルのComponentsフォルダにコピー
【MECSUtilsユニット】

<http://cc.embarcadero.com/item/26061>

- ② [ファイル | プロジェクトを開く]より
ComponentsフォルダのMGRComponents.dpk
を開きます

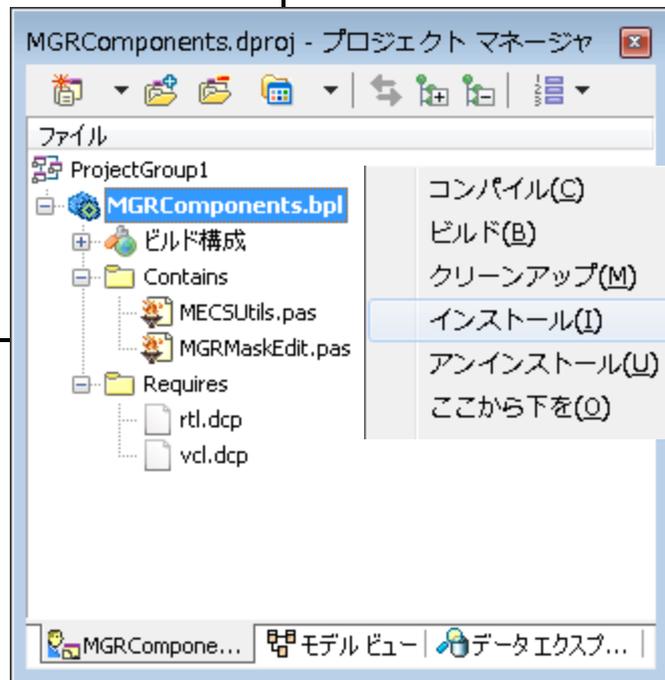


■おまけ資料

Delphi/400 V2009V2010用MGRMaskEditコンポーネント

③ プロジェクトマネージャで
MGRComponents.bplを右クリックして

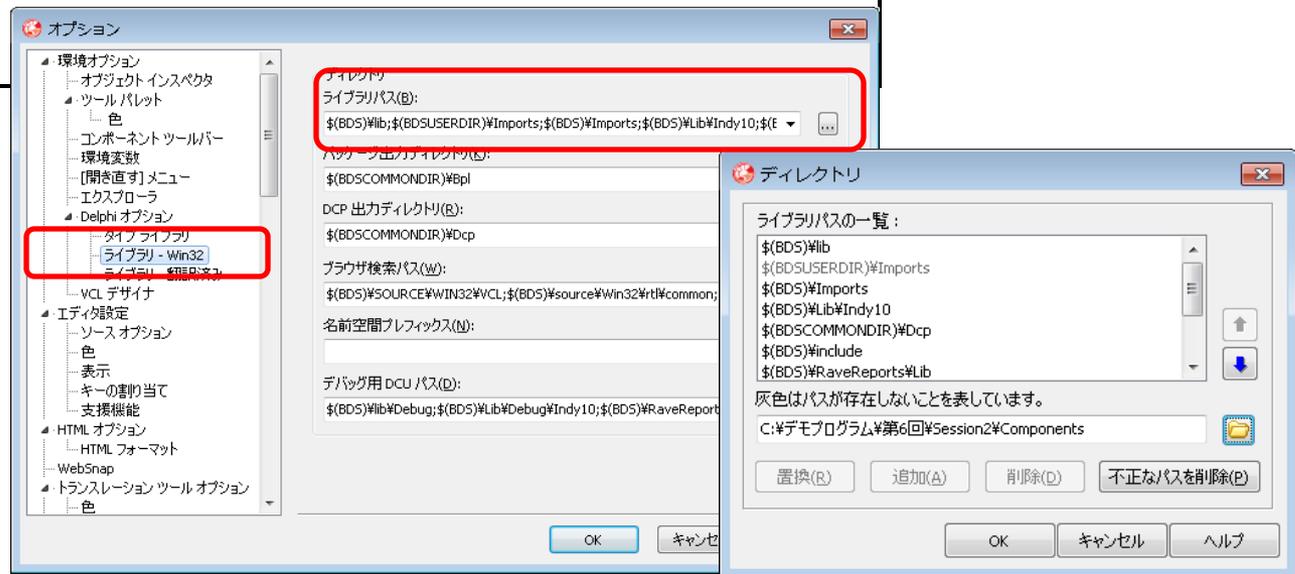
- ・コンパイル
 - ・インストール
- を行います。



■おまけ資料

Delphi/400 V2009V2010用MGRMaskEditコンポーネント

④ [ツール|オプション]より
「Delphiオプション|ライブラリ-Win32」を選択
ライブラリパスのボタンをクリックして
ライブラリパスの一覧にComponentsフォルダ
を追加します。



導入完了

■おまけ資料

Delphi/400 V2009V2010用MGRMaskEditコンポーネント

Unicodeでの主な対応点 function GetEbcDicLength部

```
for i := 1 to Length(s) do
begin
  if MecsisFullWidth(s, i, IsFarEastCharSet(AFont)) then
  begin
    if InDBCS = False then
    begin
      InDBCS := True; //全角文字
      Result := Result + 3;
    end
    else
    begin
      Result := Result + 1;
    end;
  end
  else
  begin
    InDBCS := False; //ASCII 文字もしくは半角カタカナ
  end;
end;
```

文字桁のチェックをバイトでの判断から
MECSUtilsの
MecsisFullWidth判断
に変更

■おまけ資料

Delphi/400 V2009V2010用MGRMaskEditコンポーネント

Unicodeでの主な対応点 procedure TMGRMaskEdit.Change部

```
case FCharaSet of
csSBCSOnly: //半角文字限定の場合、全角文字を消去
begin
i := 1;
sText := Text;
while i <= Length(sText) do
begin
if MecsisFullWidth(sText, i, IsFarEastCharSet(Font.Charset)) then
begin
Delete(sText, i, 1);
end
else
begin
Inc(i);
end;
end;
if Text <> sText then Text := sText;
end;
```

文字桁のチェックをバイトでの判断から
MECSUtilsの
MecsisFullWidth判断
に変更

■おまけ資料

Delphi/400 V2009V2010用MGRMaskEditコンポーネント

Unicodeでの主な対応点 procedure TMGRMaskEdit.KeyPress部

```
procedure TMGRMaskEdit.KeyPress(var Key: Char);  
begin  
  inherited;  
  
  if FEnterNext and (Key = Chr(VK_RETURN)) then Key := #0;  
  
  if (FPageCode = pcJP1) and (FCharaSet <> csDBCSOnly) then  
    if not (MecsIsFullWidth(Key, 1, IsFarEastCharSet(Font.Charset))) then  
      begin  
        Key := UpCase(Key);  
      end;  
end;
```

文字桁のチェックをバイトでの判断から
MECSUtilsの
MecsIsFullWidth判断
に変更