【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

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

【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

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q1. IBM i を使ったファイルのアップロード/ダウンロード

【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？

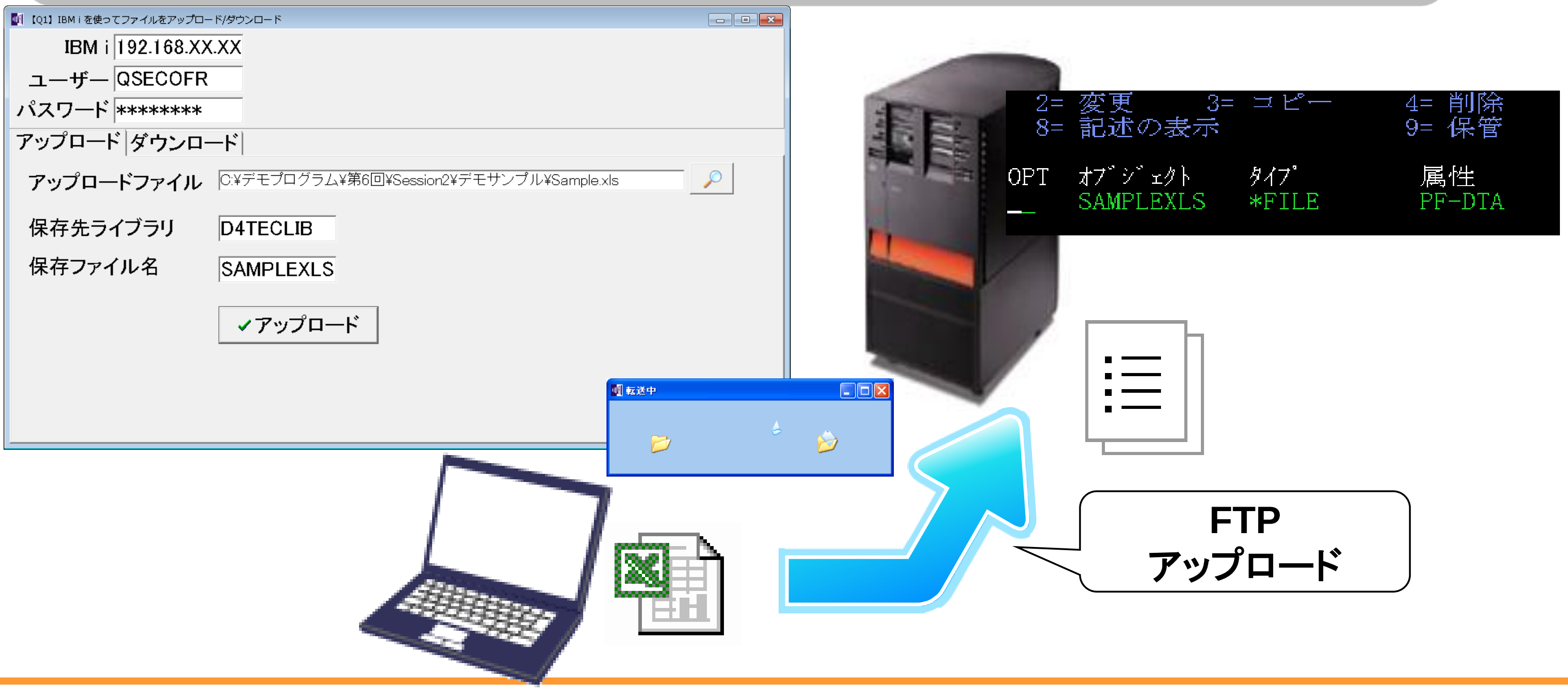
【回答】

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。

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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



本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

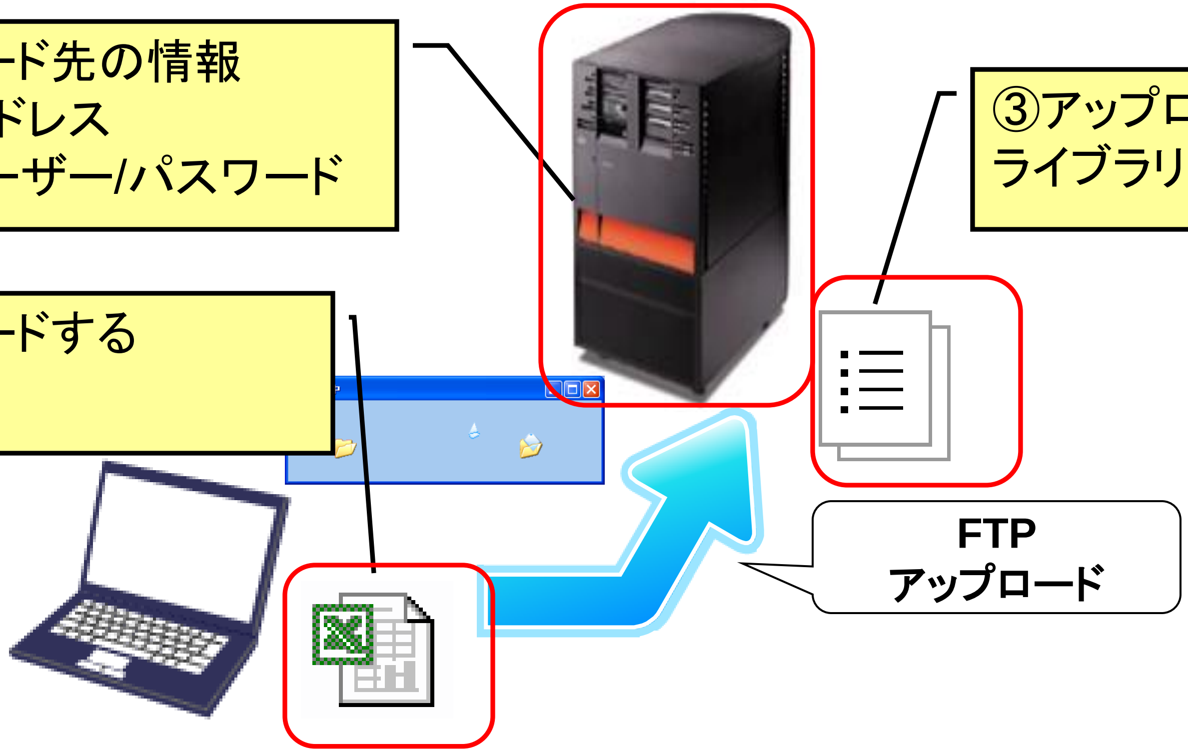
■Q1. IBM i を使ったファイルのアップロード/ダウンロード

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

③アップロード先の
ライブラリ名/ファイル名



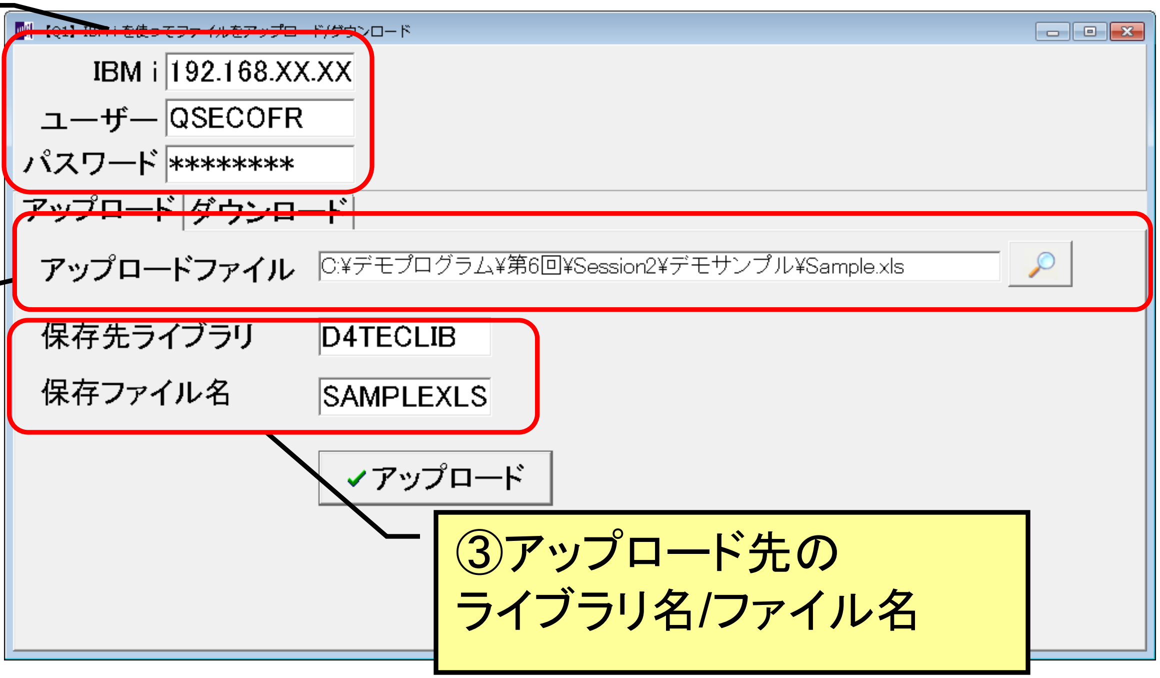
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q1. IBM i を使ったファイルのアップロード/ダウンロード

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

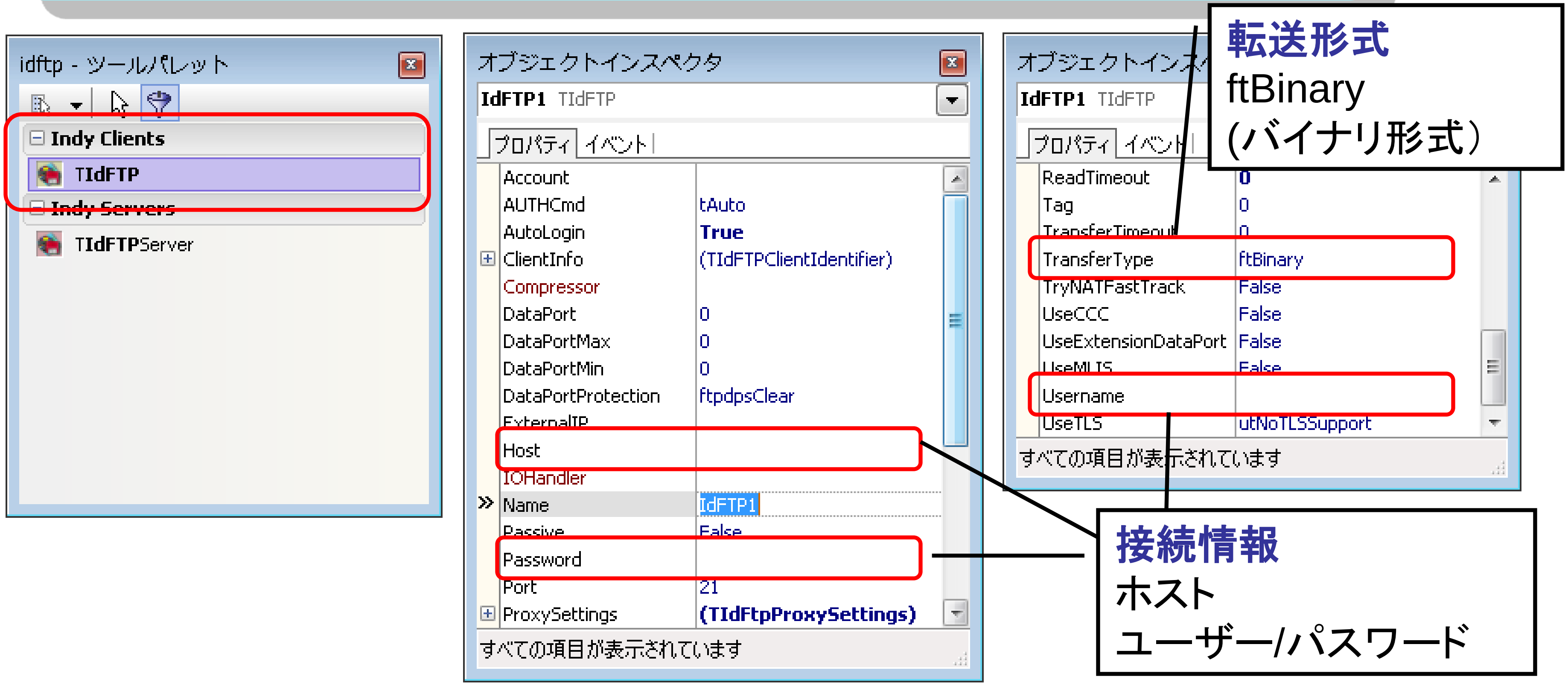


③アップロード先の
ライブラリ名/ファイル名

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q1. IBM i を使ったファイルのアップロード/ダウンロード

IdFTPコンポーネント



本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q1. IBM i を使ったファイルのアップロード/ダウンロード

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

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q1. IBM i を使ったファイルのアップロード/ダウンロード

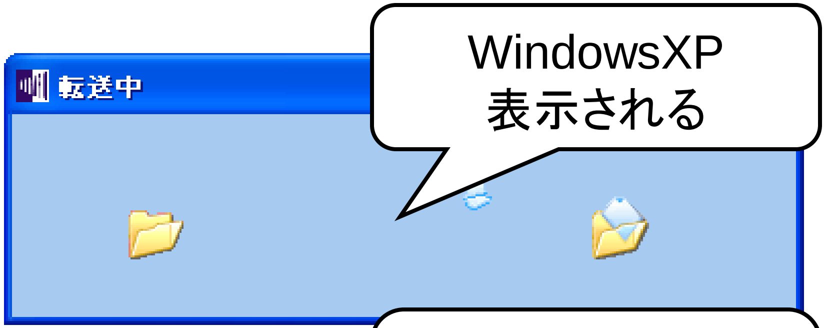
```
      //アップロードした情報を管理ファイルに格納
      with DataModule1.Table1 do
      begin
        Append;
        FieldByName('UPLDLB').AsString :=  edtUPLIB.Text;
        FieldByName('UPLDPF').AsString :=  edtUPPF.Text;
        FieldByName('UPFLNM').AsString :=  ExtractFileName(edtUPFILE.Text);
        Post;
        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;
```

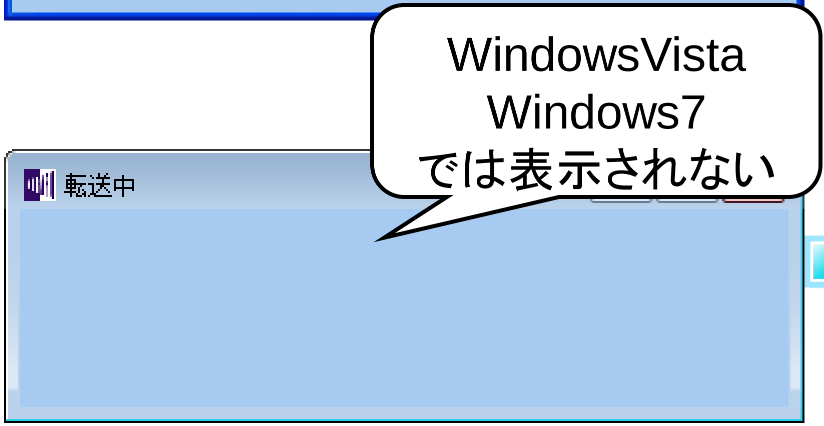
ダウンロードする為に
アップロード情報を
アップロード管理ファイル
にレコード追加
※サンプル仕様

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

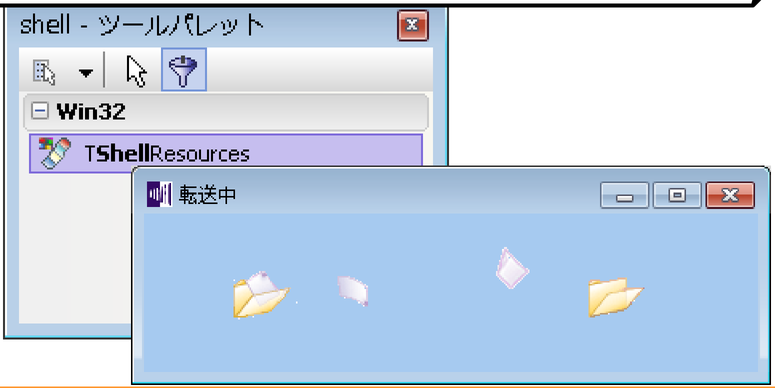
■Q1. IBM i を使ったファイルのアップロード/ダウンロード

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





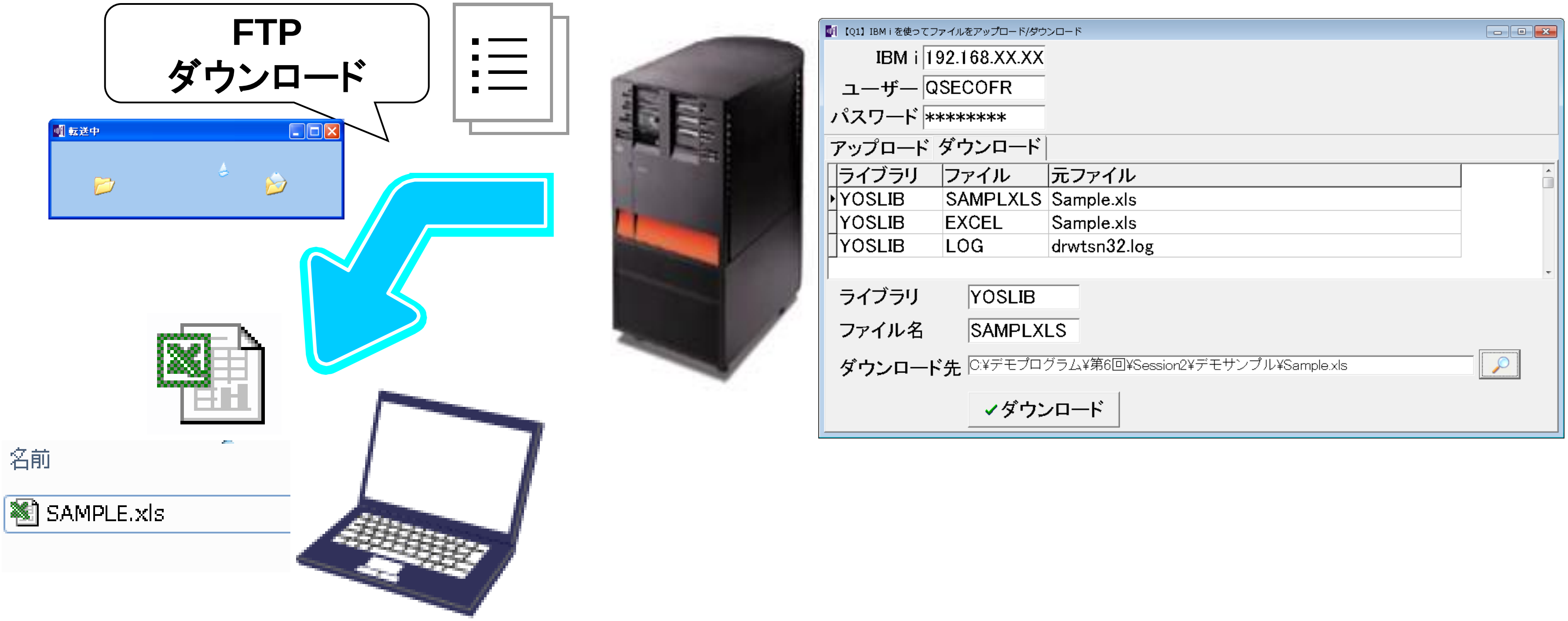
Windows Vista からは
リソースの管理が変わっており、
Shell32.dll にアニメーションで必要な
avi が含まれていないので表示できません。
Delphi2007以降は
TShellResourceコンポーネントを置く
ことで新しいアニメーションを表示できます。



本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q1. IBM i を使ったファイルのアップロード/ダウンロード

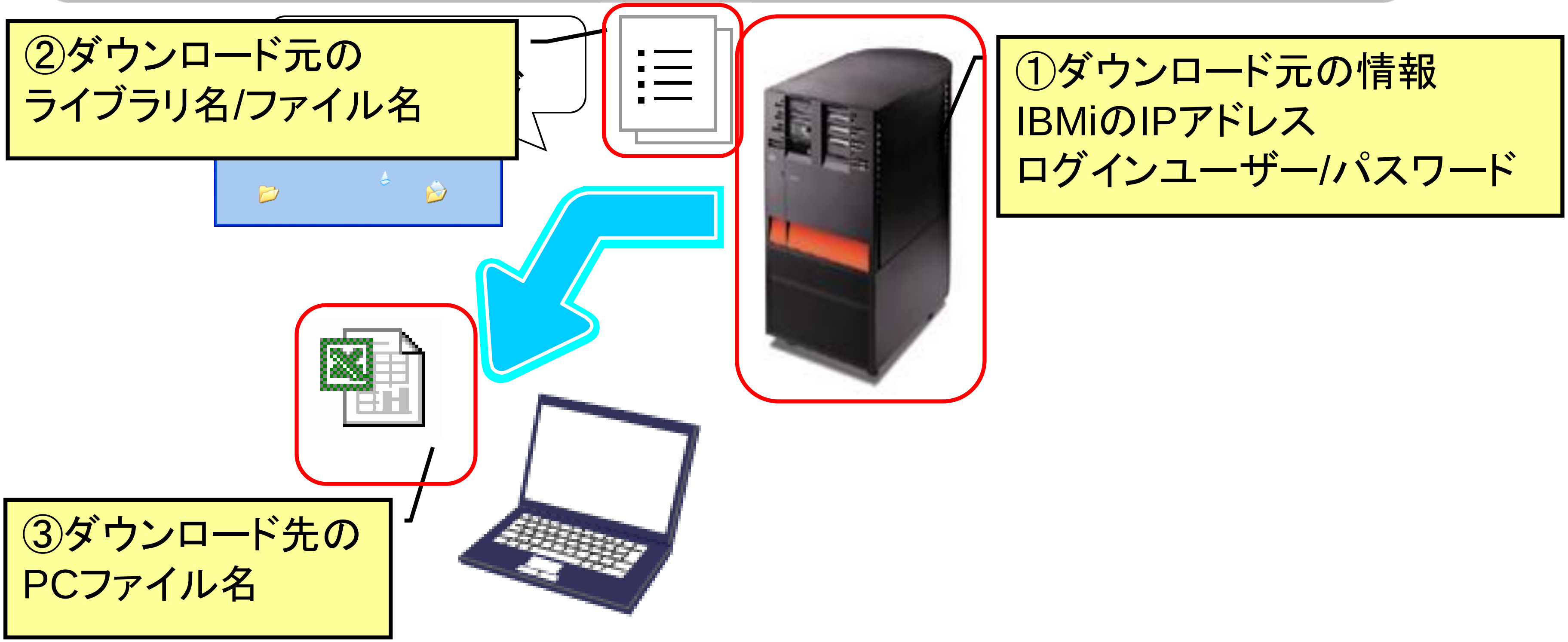
IBM i のファイルをダウンロードする



本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q1．IBM i を使ったファイルのアップロード/ダウンロード

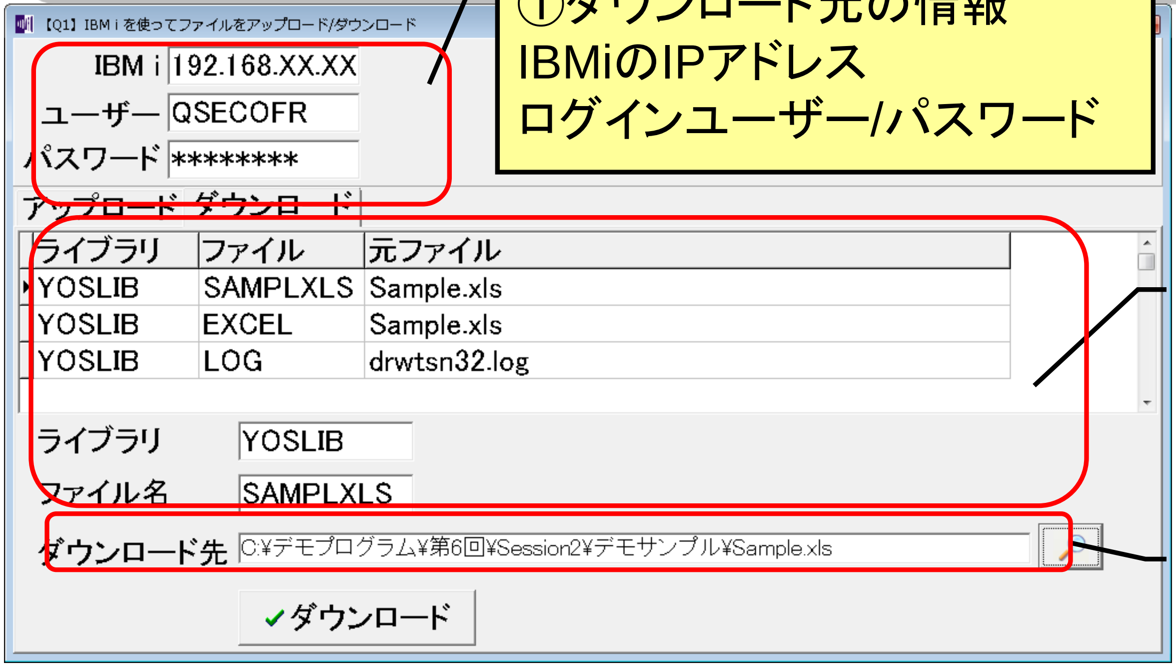
ファイルをFTPダウンロードするために必要な情報



本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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



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

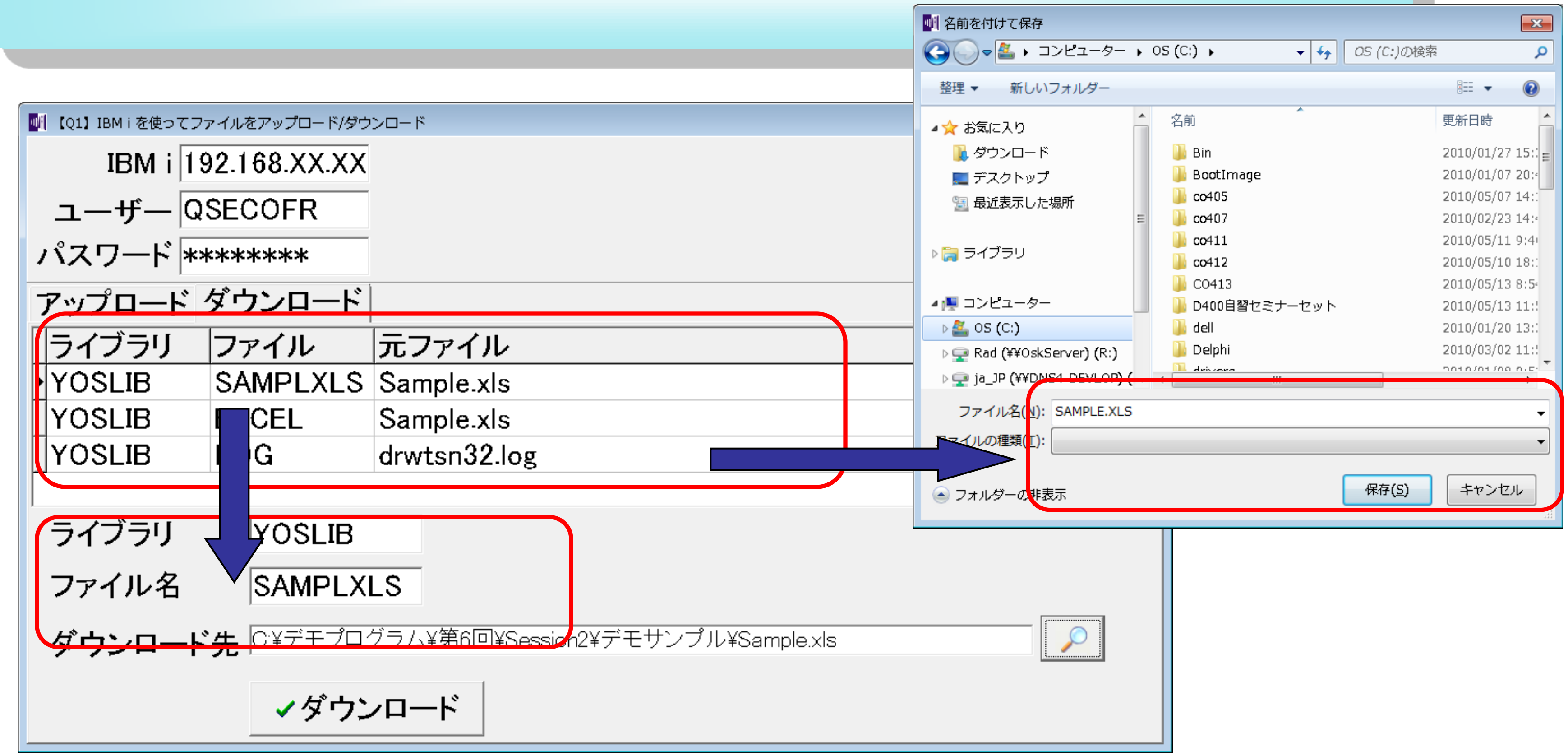
②ダウンロード元の
ライブラリ名/ファイル名

③ダウンロード先の
PCファイル名

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q1．IBM i を使ったファイルのアップロード/ダウンロード

インターフェースの工夫



本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q1. IBM i を使ったファイルのアップロード/ダウンロード

ダウン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

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q1. IBM i を使ったファイルのアップロード/ダウンロード

```
        //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

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q2. トレイアイコンを利用したアプリケーションの常駐

【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。

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

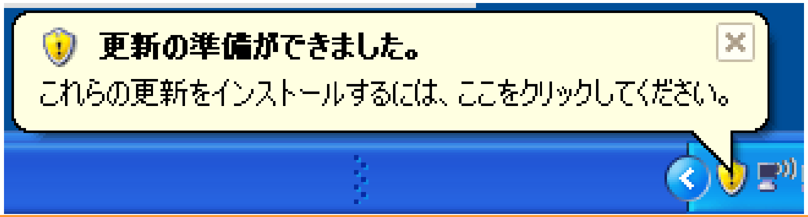
■Q2. トレイアイコンを利用したアプリケーションの常駐

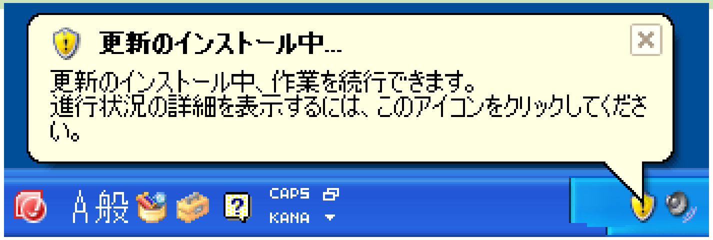
トレイアイコンとはデスクトップの右下のタスクトレイに
格納された常駐しているアプリケーションのアイコンです。

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

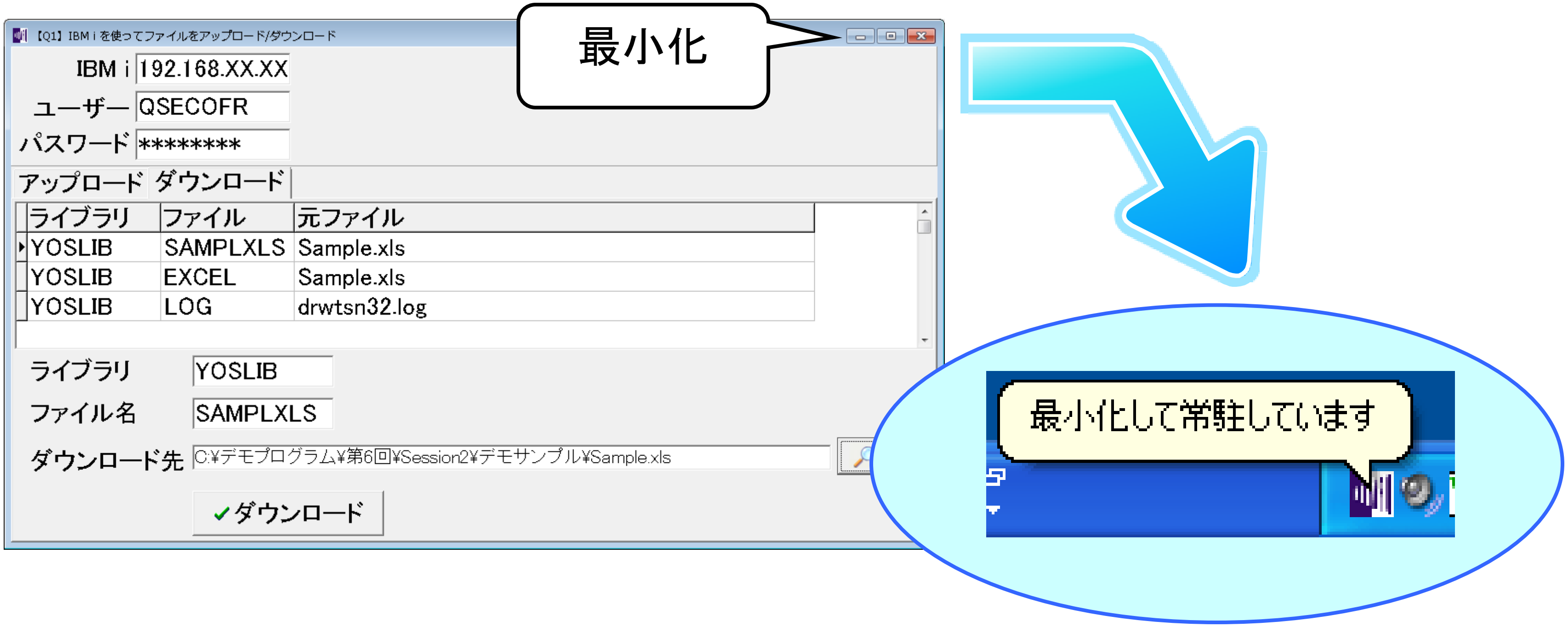




本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q2. トレイアイコンを利用したアプリケーションの常駐

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

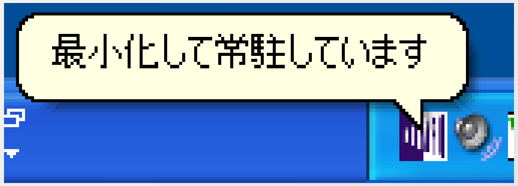


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q2. トレイアイコンを利用したアプリケーションの常駐

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

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q2. トレイアイコンを利用したアプリケーションの常駐

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

①最小化時に
アップロード件数
を取得

②タイマーコンポーネント
で定期的にアップロード
件数を取得

①より②の
アップロード件数が多ければ
ばメッセージ通知

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q2. トレイアイコンを利用したアプリケーションの常駐

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ｺﾝﾎﾟｰﾈﾝﾄを利用

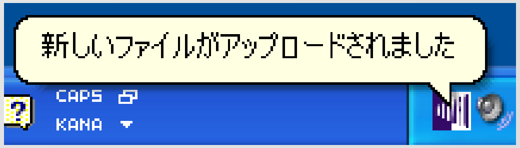
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q2. トレイアイコンを利用したアプリケーションの常駐

タイマーイベント（アップロード検知処理）

```
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;
```

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



本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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

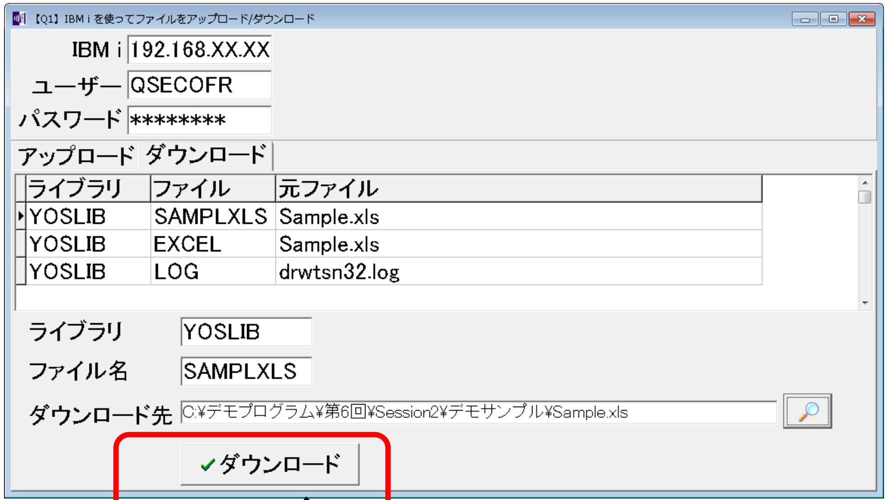
```
procedure TfrmQ2.TrayIcon1Click(Sender: TObject);
begin
  Self.WindowState   := wsNormal;   //画面サイズを通常に設定
  Self.Visible       := True;       //画面表示
  TrayIcon1.Visible  := False;      //トレイアイコン非表示
  Timer1.Enabled     := False;      //常駐タイマーOFF
end;n
```

トレイアイコン終了の際にタイマーも解除

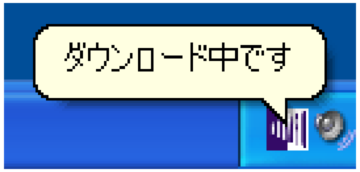
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

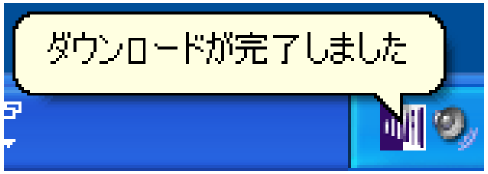
■Q2. トレイアイコンを利用したアプリケーションの常駐

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



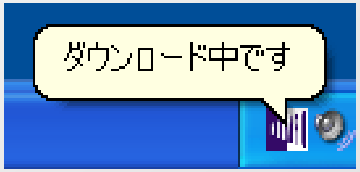


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q2. トレイアイコンを利用したアプリケーションの常駐

ダウン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);
```

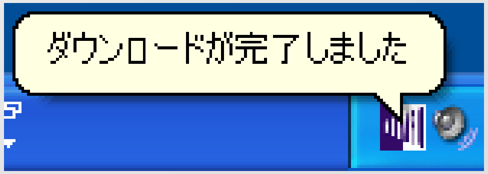


待ち受け画面の
代わりにトレイアイコン
でメッセージ表示

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q2. トレイアイコンを利用したアプリケーションの常駐

```
    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

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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。

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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。**

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

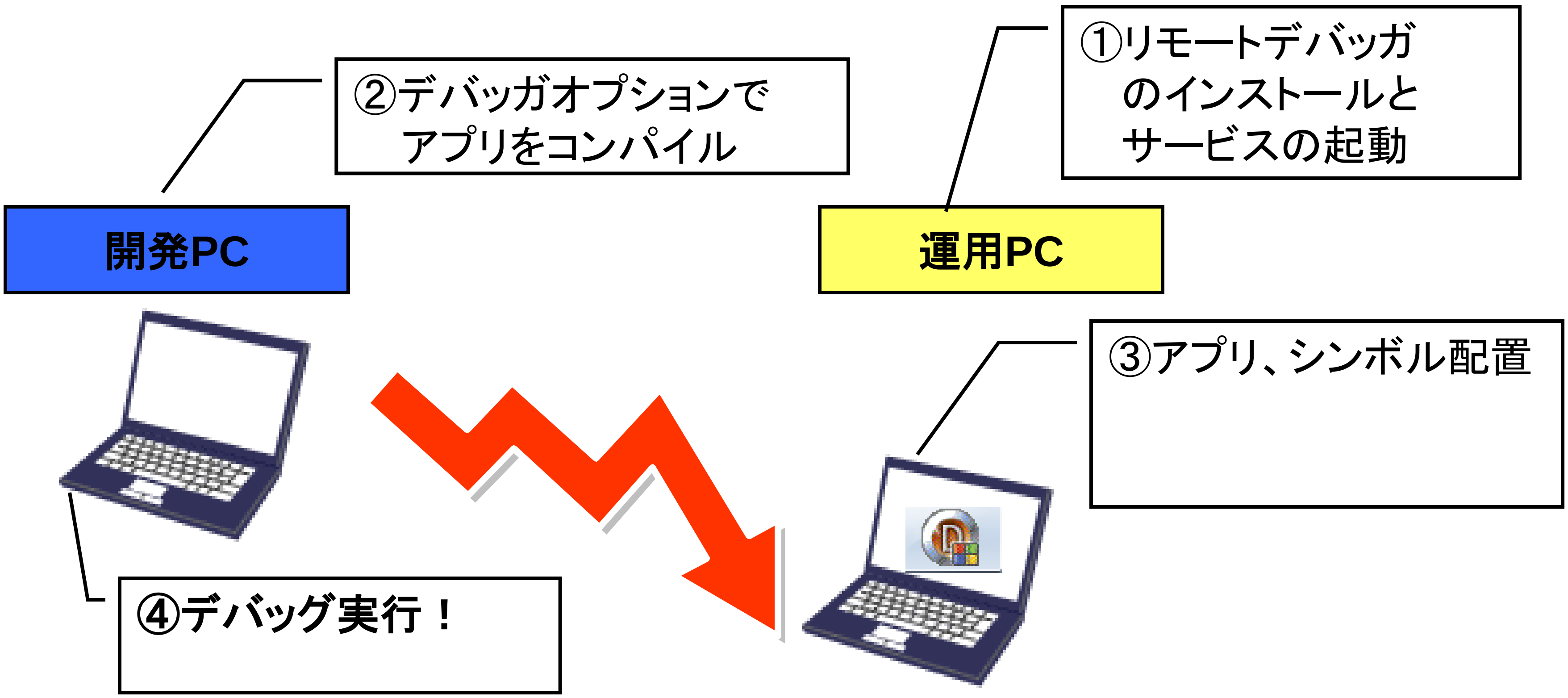
■Q3. リモートデバッガで運用PCのデバッグ調査

■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。

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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

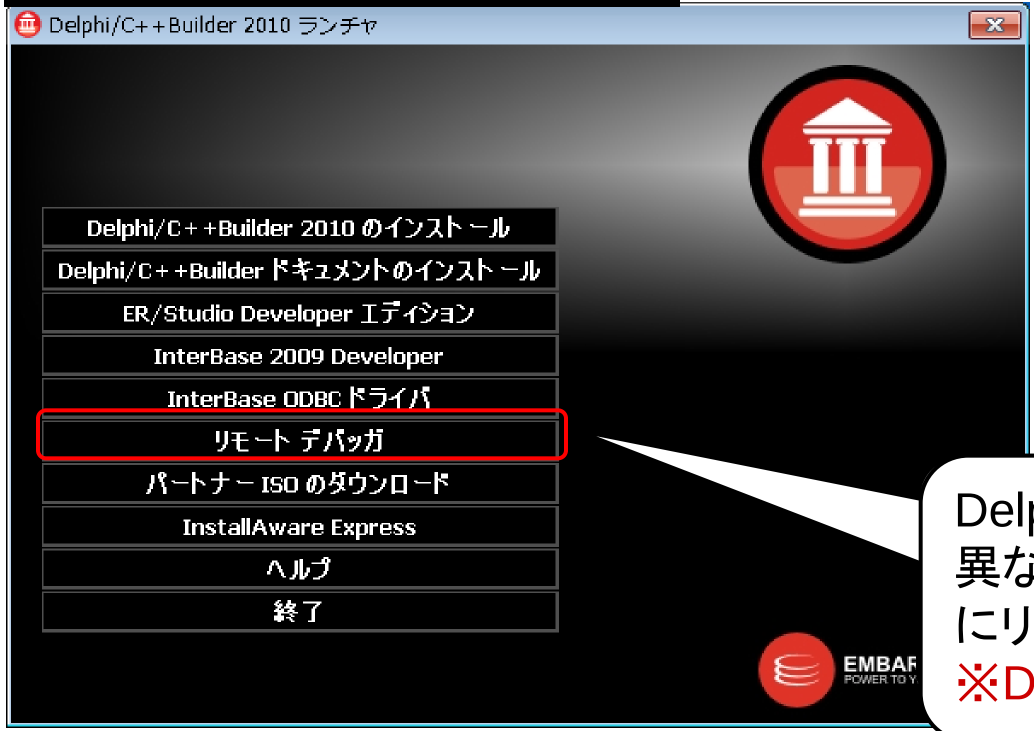


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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

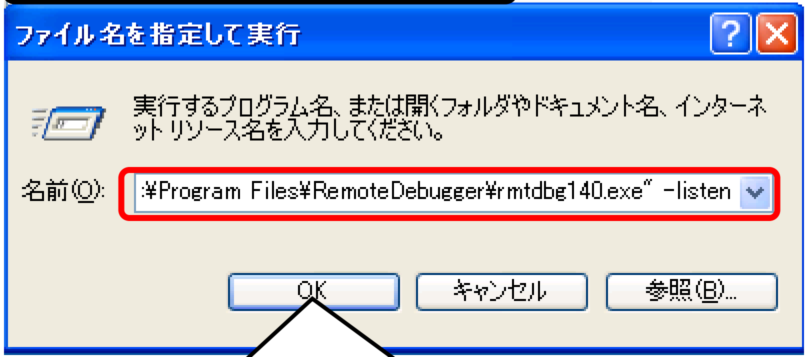
Delphiのバージョンによって名称等異なりますがDelphiのインストーラにリモートデバッガが含まれています。
※Delphi2005だけは付属していません。

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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。

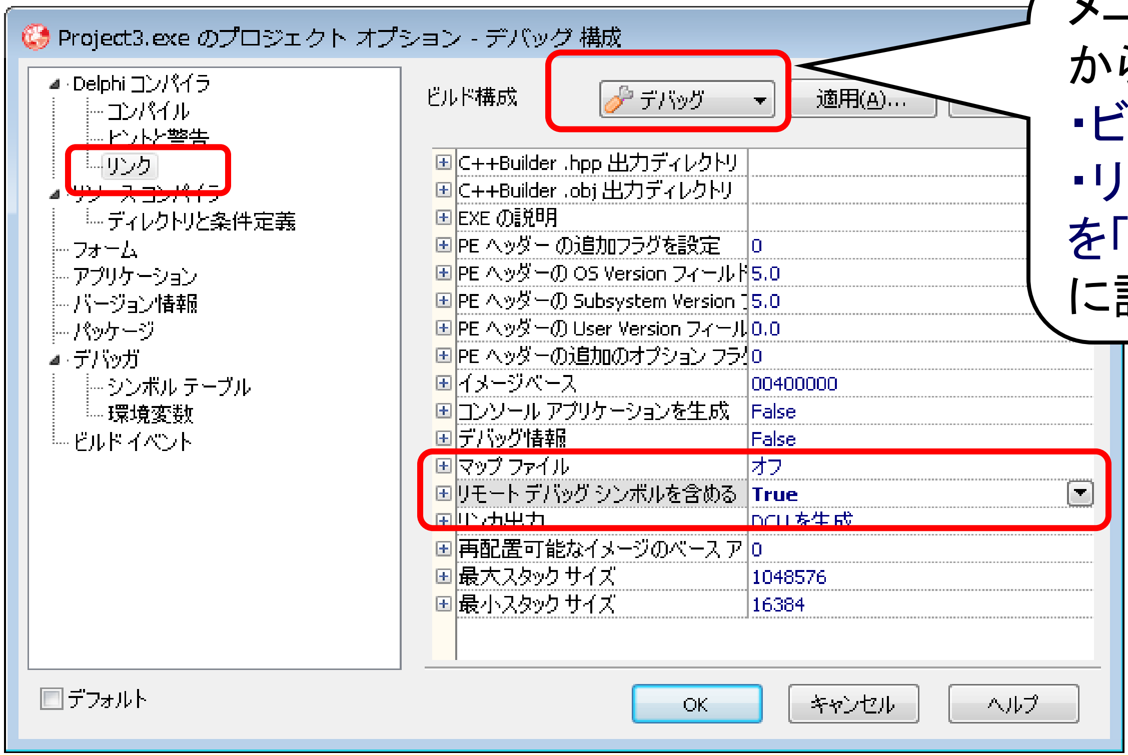
サービスが起動すると
タスクトレイに虫のアイコン
が表示されます。
ウィルスではありません！

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q3. リモートデバッガで運用PCのデバッグ調査

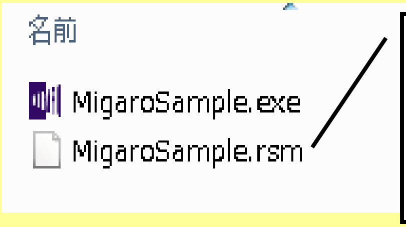
②デバッガオプションでアプリをコンパイル

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



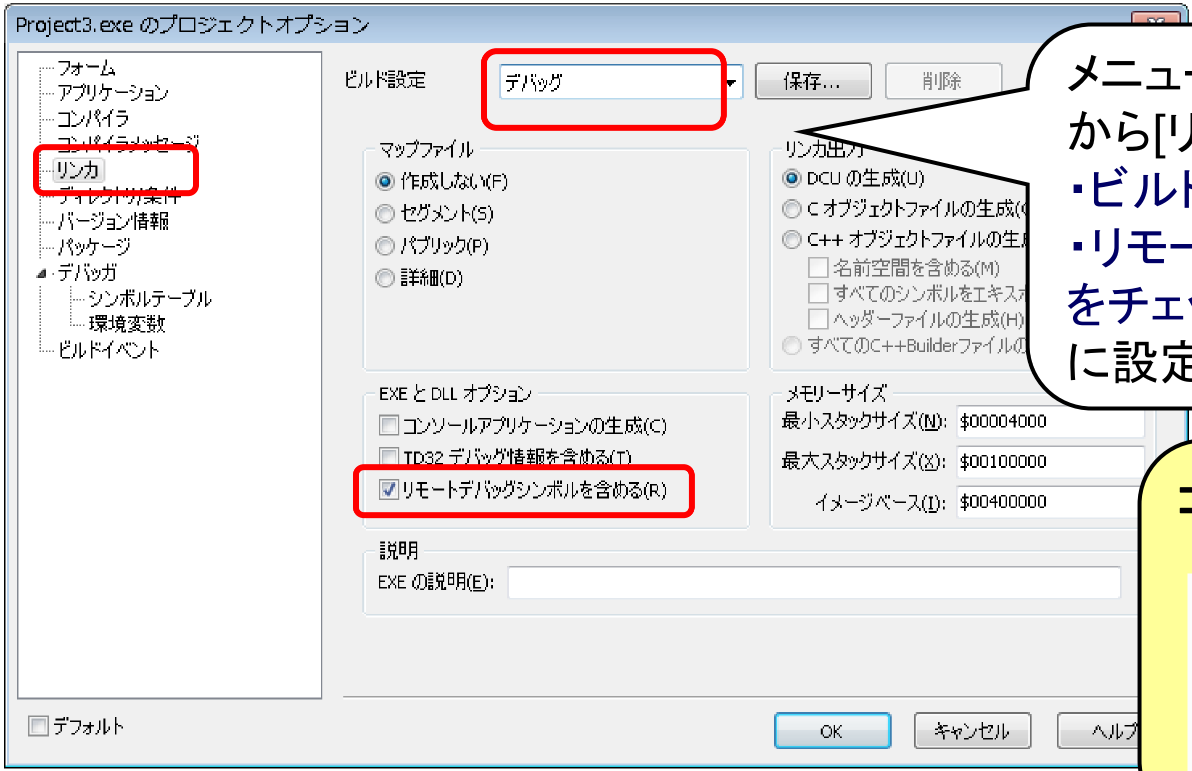
シンボルファイル(.rsm)

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q3. リモートデバッガで運用PCのデバッグ調査

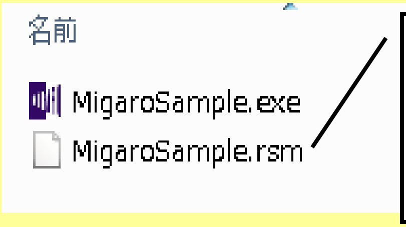
②デバッガオプションでアプリをコンパイル

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



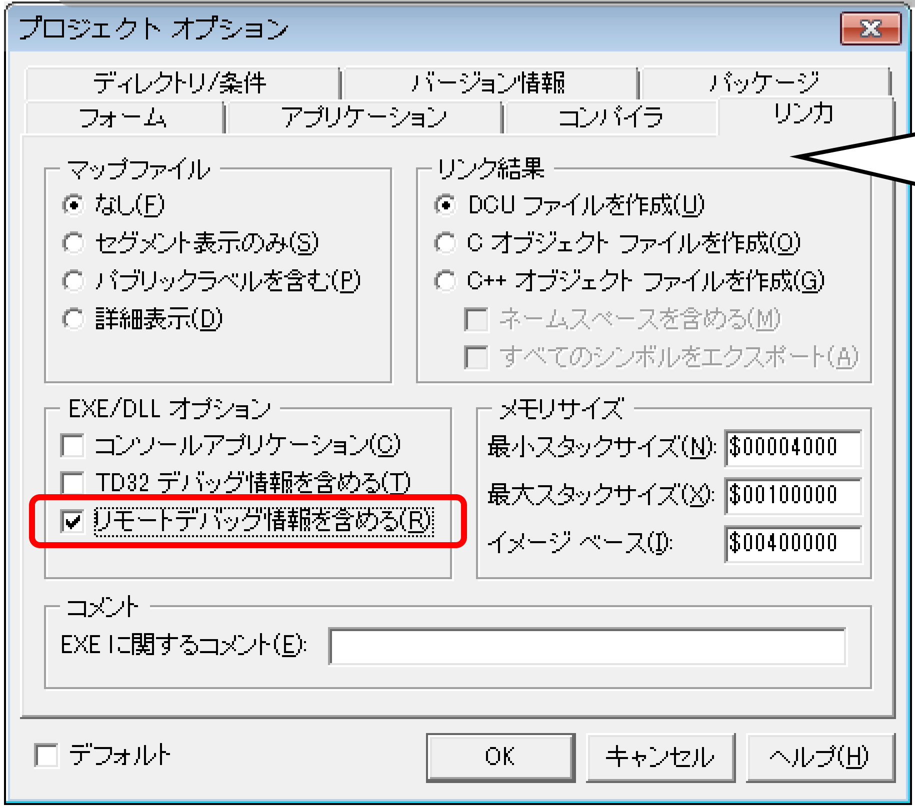
シンボルファイル(.rsm)

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q3. リモートデバッガで運用PCのデバッグ調査

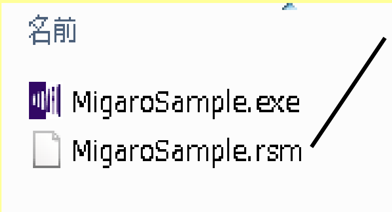
②デバッガオプションでアプリをコンパイル

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

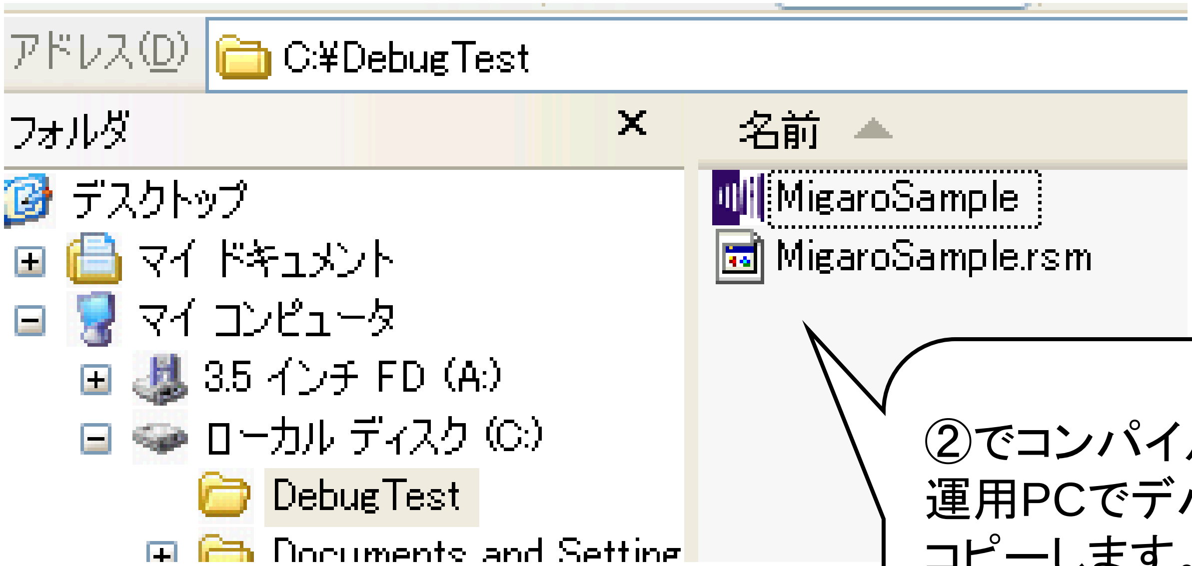


シンボルファイル(.rsm)

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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。

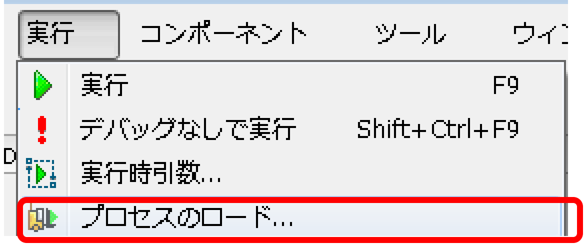
・アプリケーション(.exeや.dll)
・シンボルファイル(.rsm)

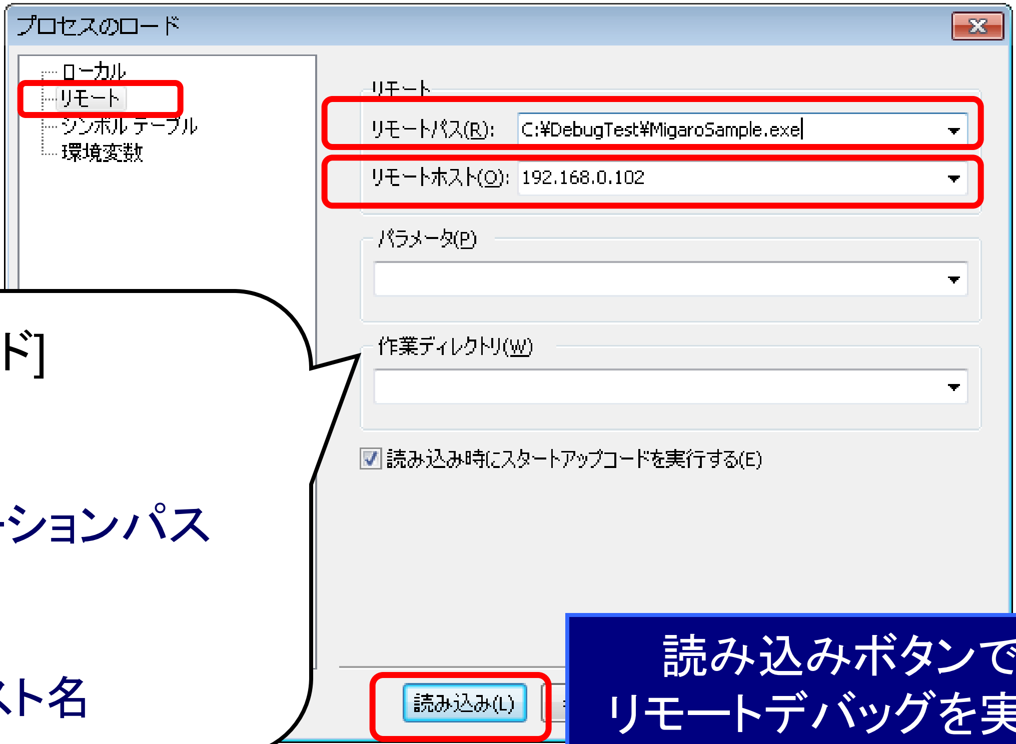
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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

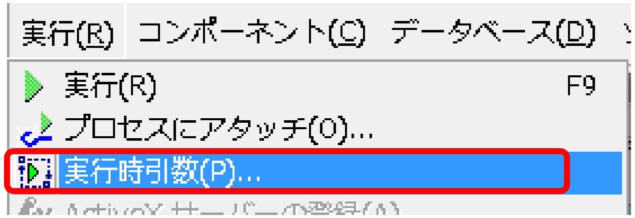
読み込みボタンで
リモートデバッグを実行

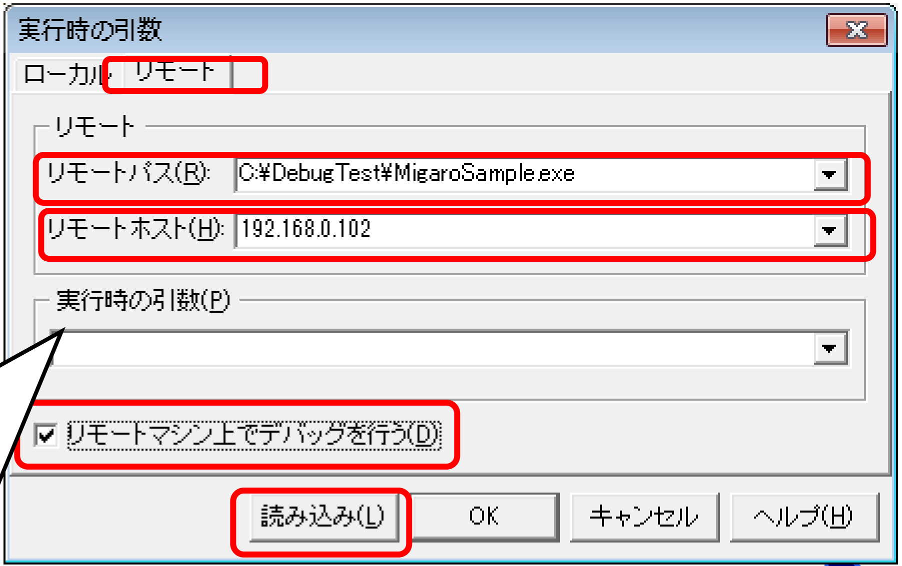
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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

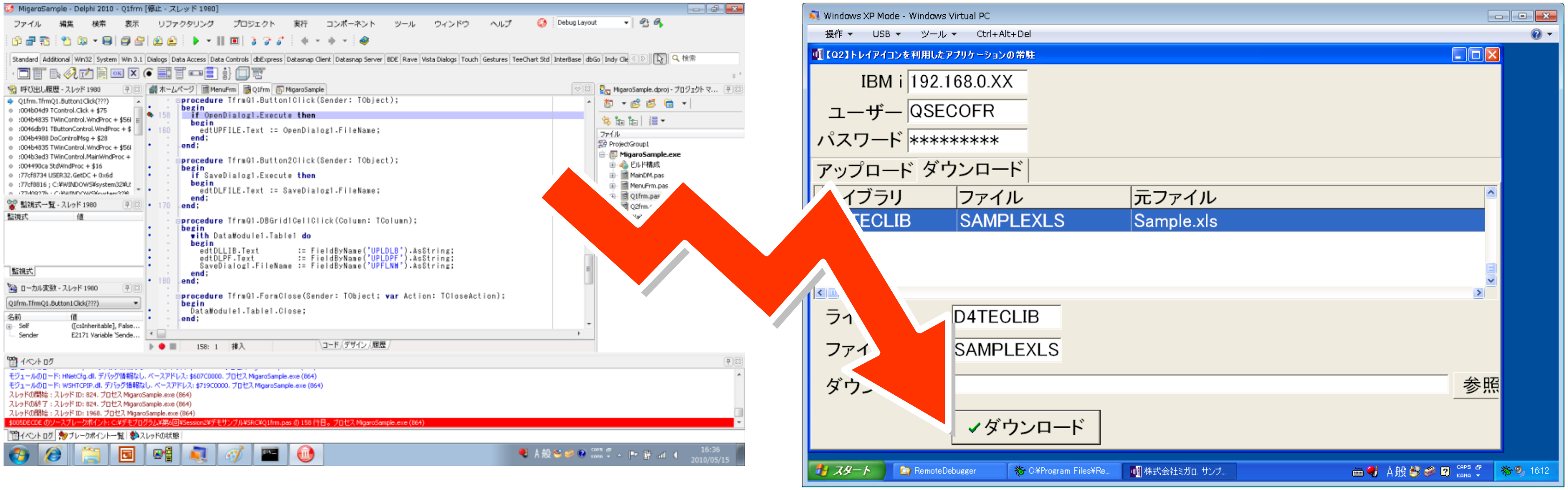
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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

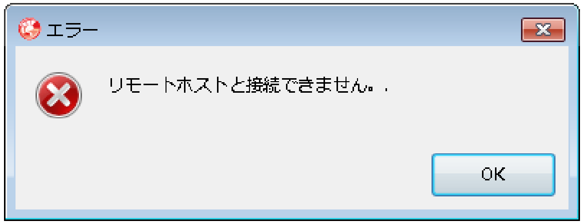


運用PCで実行しながら開発環境でデバッグ調査が可能に！

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q3. リモートデバッガで運用PCのデバッグ調査

補足



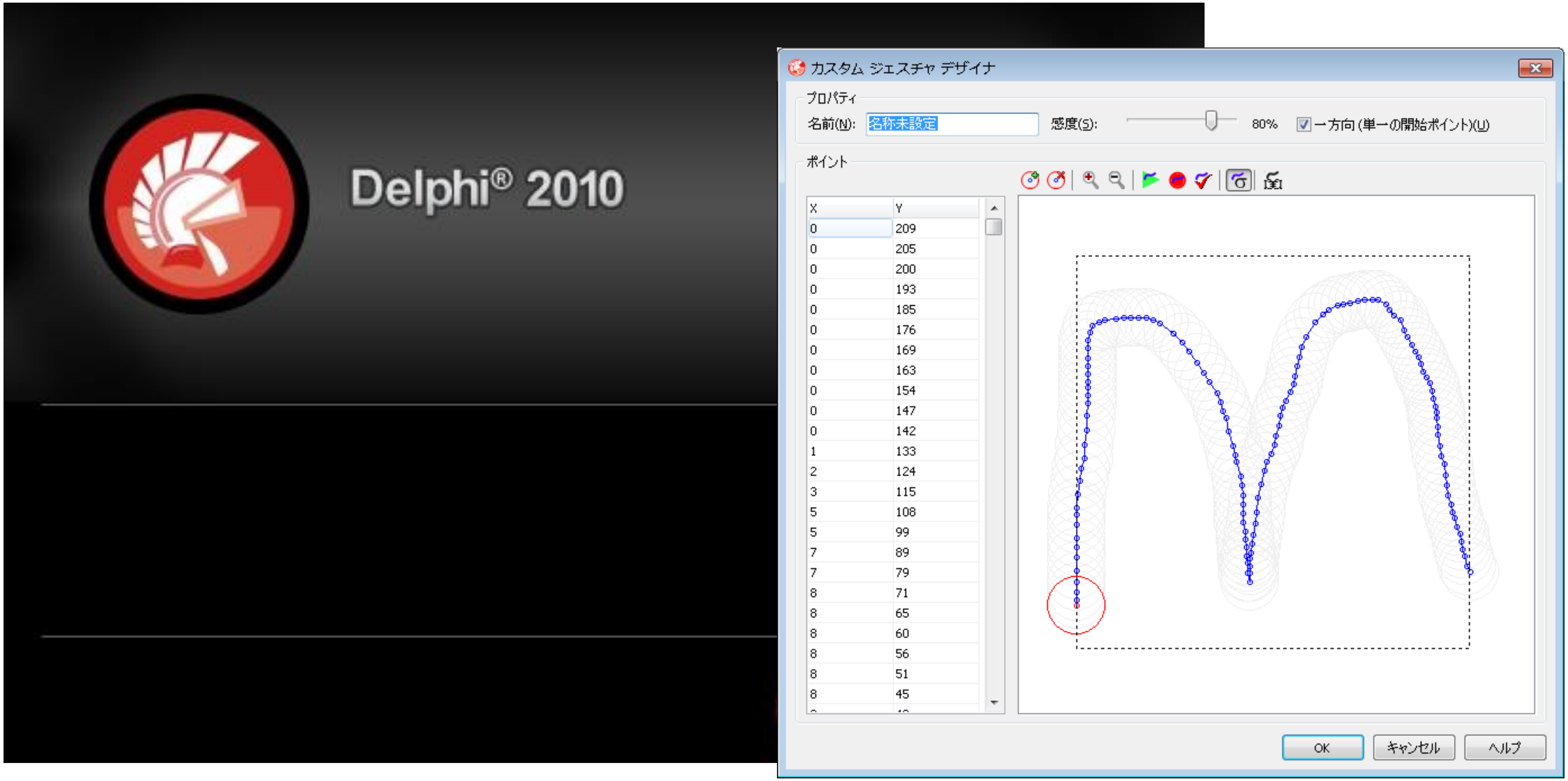
運用PCにリモート接続ができない場合は
次のような点を確認してください。
・サービスが起動しているか
・シンボルファイルが配布されているか
・Windowsファイアーウォールや
ウィルス対策ソフトで規制されていないか

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

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q4 Delphi/400 V2010 新機能紹介

－ ジェスチャアプリケーション作成 －



本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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。

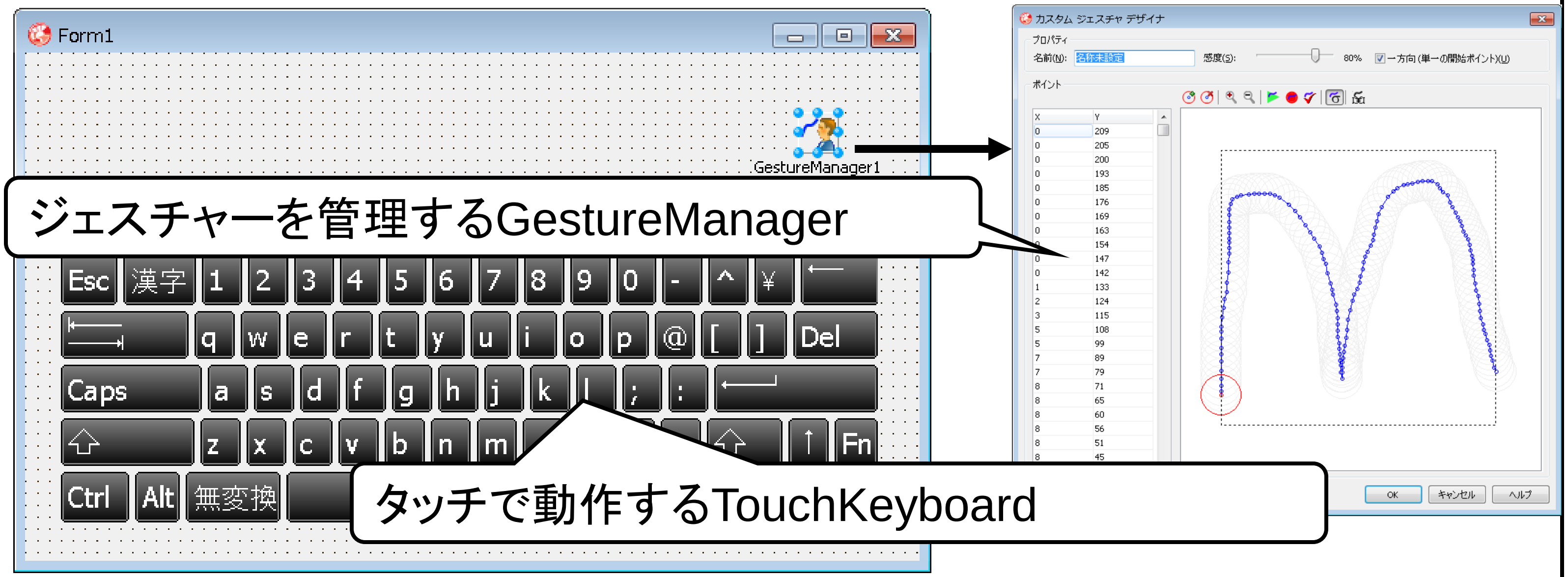
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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。

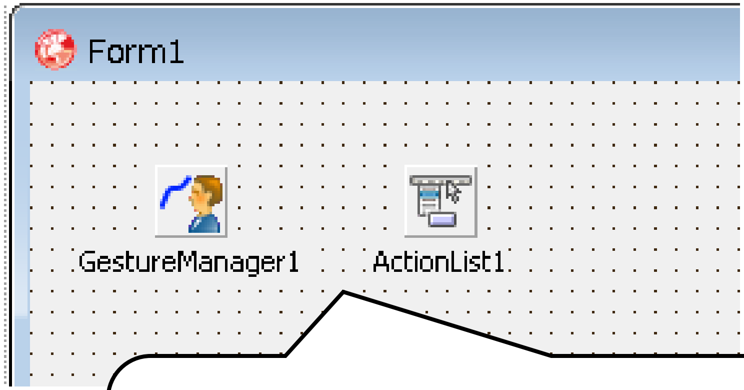


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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①



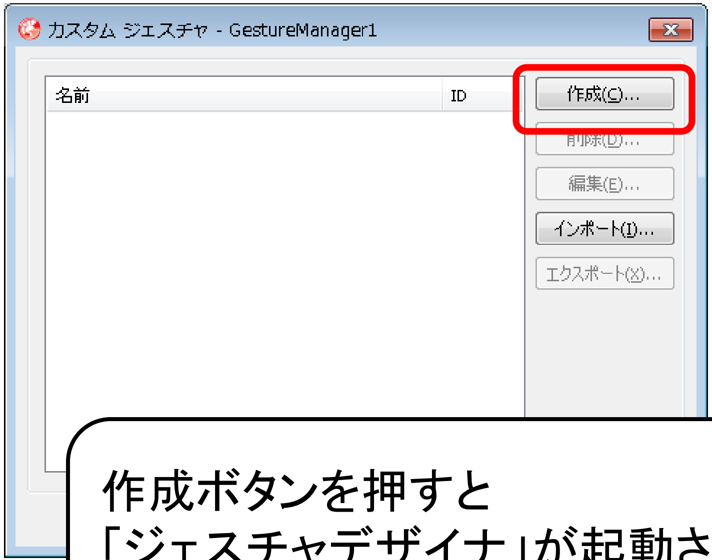
GestureManagerコンポーネントと
ActionListコンポーネントを配置して
GestureManegerをダブルクリック

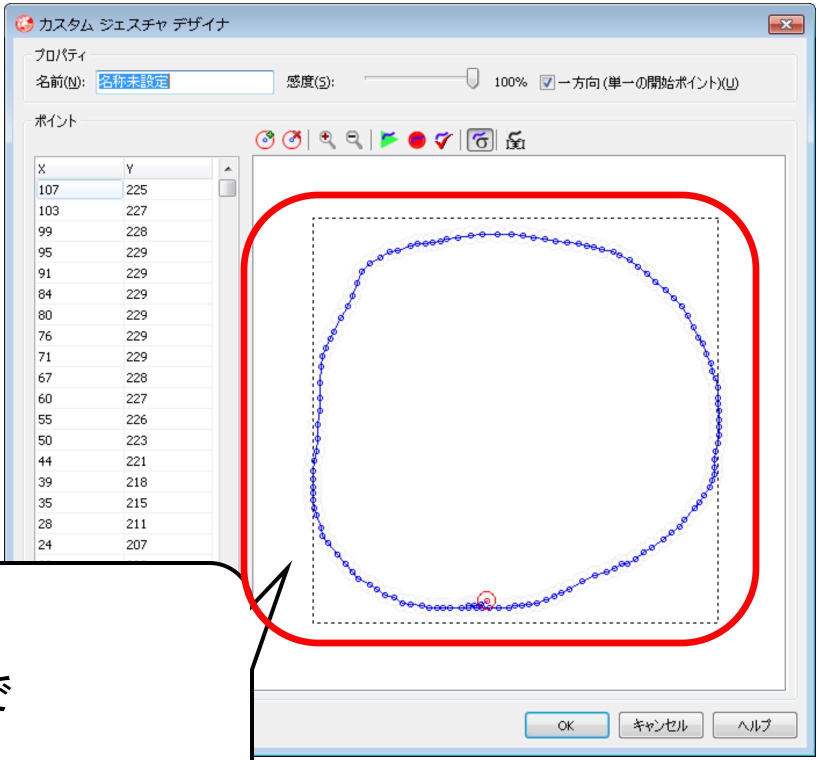
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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





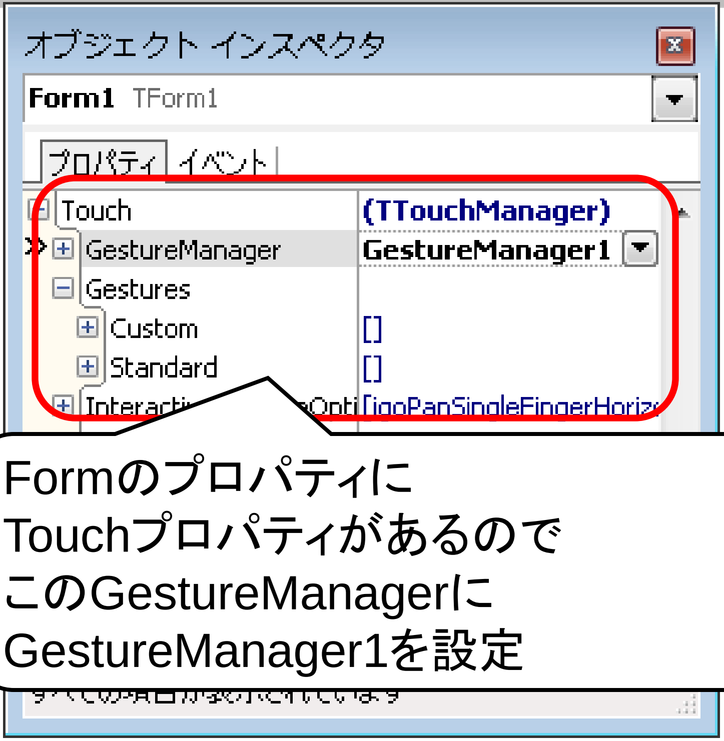
作成ボタンを押すと
「ジェスチャデザイナ」が起動されるので
自由なジェスチャを作成して登録

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

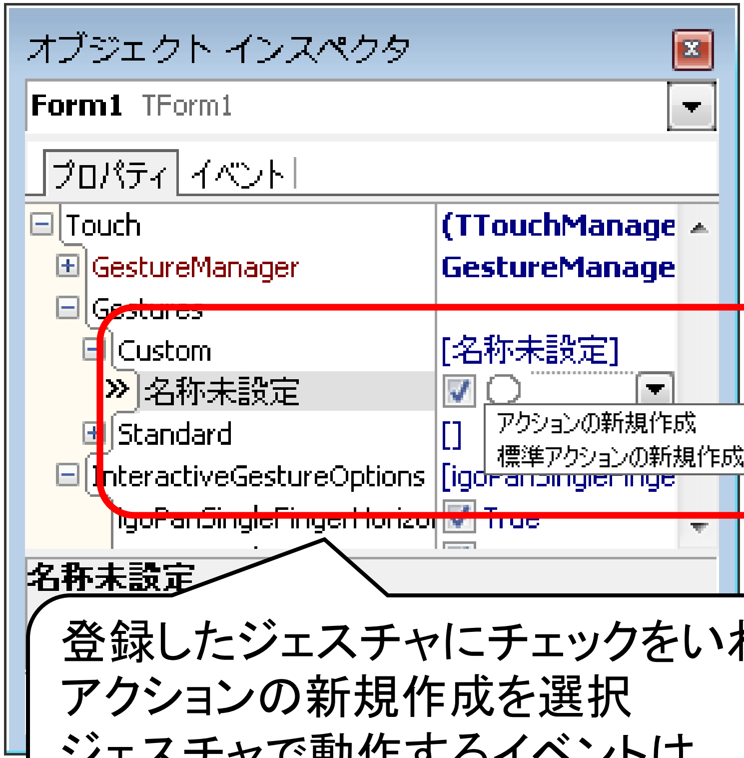
■Q4 Delphi/400 V2010 新機能紹介

– 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

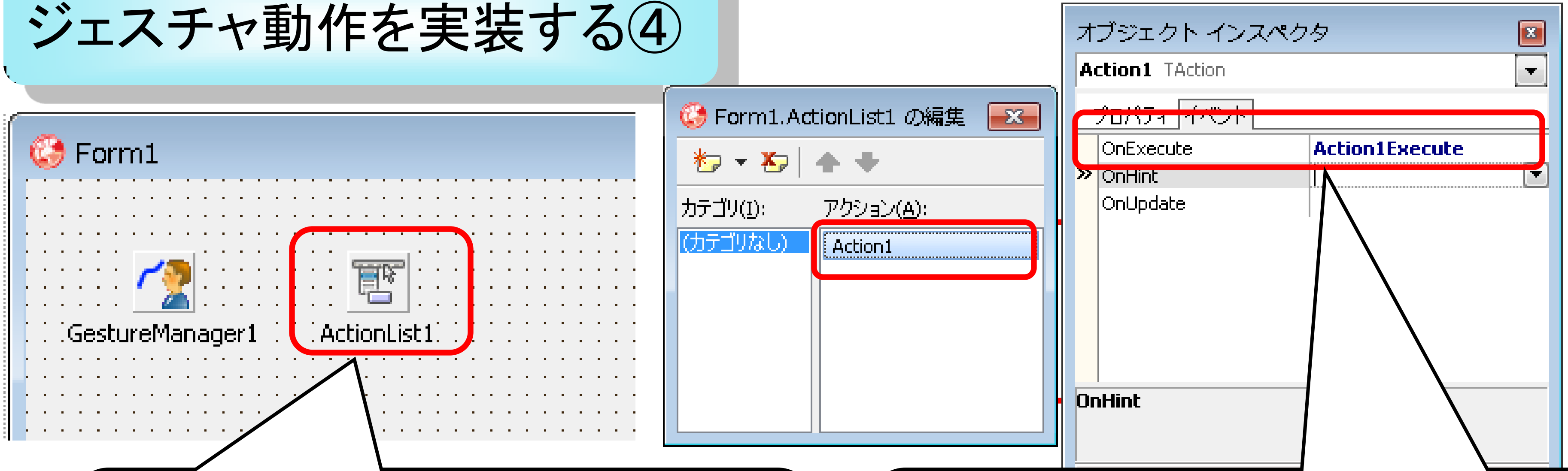


登録したジェスチャにチェックをいれると
アクションの新規作成を選択
ジェスチャで動作するイベントは
ActionListのActionで管理

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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

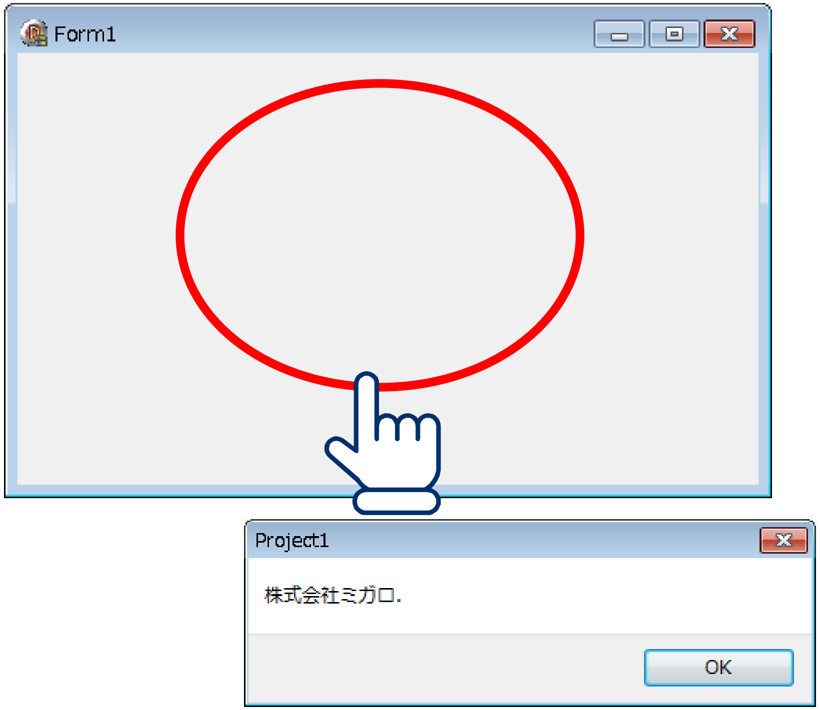
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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！



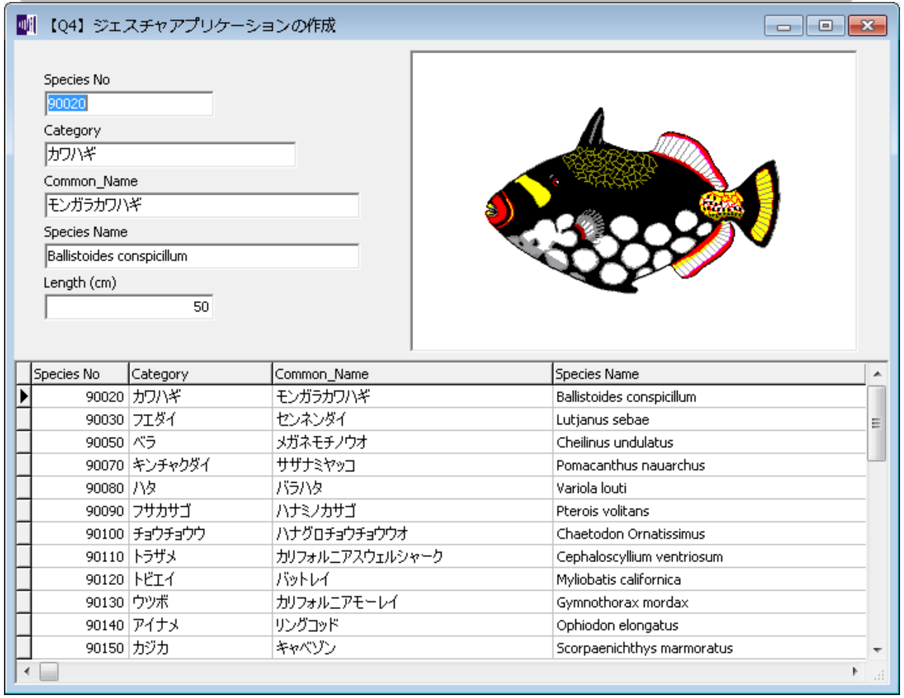
ジェスチャはOnGestureイベントでハンドリングもできますがActionで処理すると作成が容易です。

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q4 Delphi/400 V2010 新機能紹介

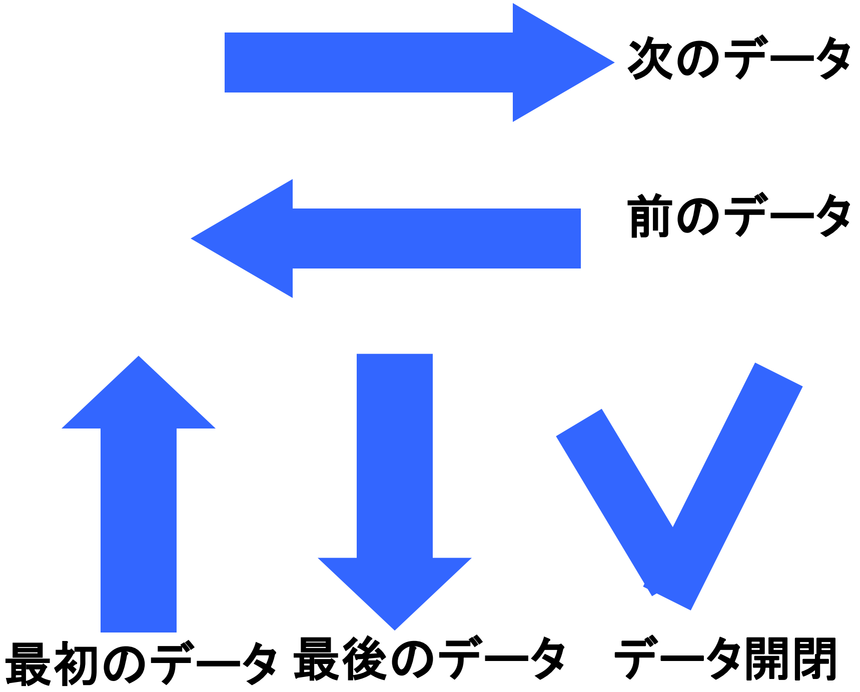
－ ジェスチャアプリケーション作成 －

既存アプリへの応用実装



Species No	Category	Common_Name	Species Name
90020	カワハギ	モンガラカワハギ	Ballistoides conspicillum
90030	フエダイ	センネンダイ	Lutjanus sebae
90050	ベラ	メガネモチノウオ	Cheilinus undulatus
90070	キンチャクダイ	サザナミヤッコ	Pomacanthus nauarchus
90080	ハタ	バラハタ	Variola louti
90090	フサカサゴ	ハナミノカサゴ	Pterois volitans
90100	チョウチョウウ	ハナグロチョウチョウウオ	Chaetodon Ornatissimus
90110	トラザメ	カリフォルニアスウェルシャーク	Cephaloscyllium ventriosum
90120	トビエイ	バットレイ	Myliobatis californica
90130	ウツボ	カリフォルニアモーレイ	Gymnothorax mordax
90140	アイナメ	リングコッド	Ophiodon elongatus
90150	カジカ	キャベゾン	Scorpaenichthys marmoratus

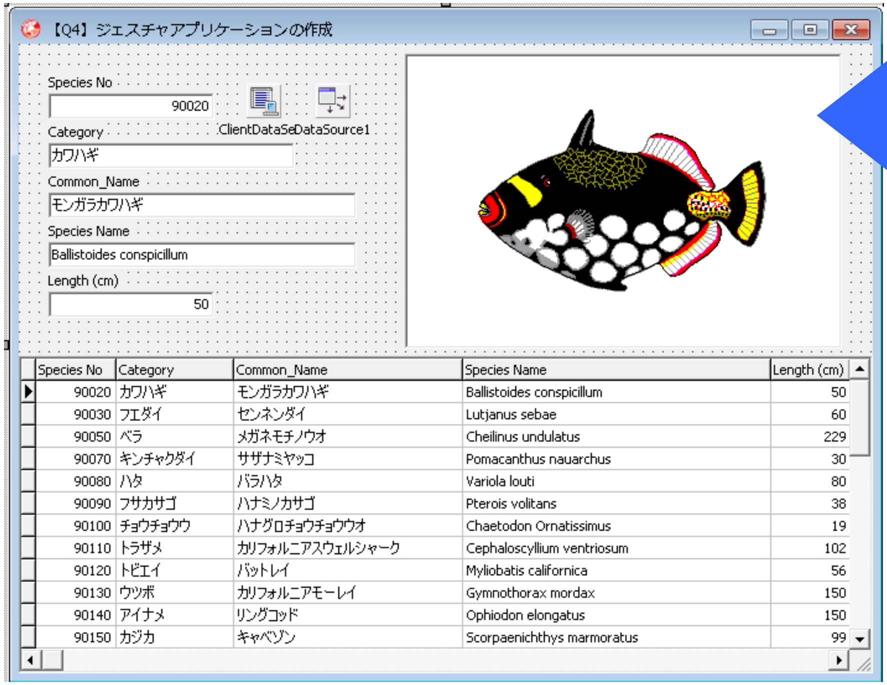
ジェスチャ設計



本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

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

既存アプリへの応用実装①

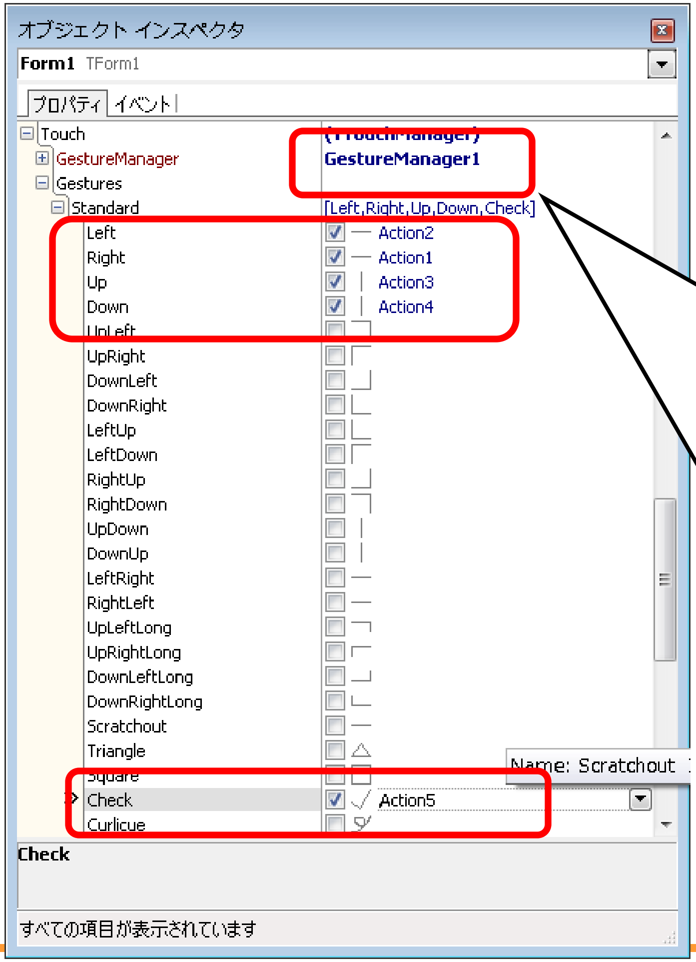


データを表示している画面に
GestureManagerとActionListを
設定

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q4 Delphi/400 V2010 新機能紹介

– ジェスチャアプリケーション作成 –



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

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

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

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

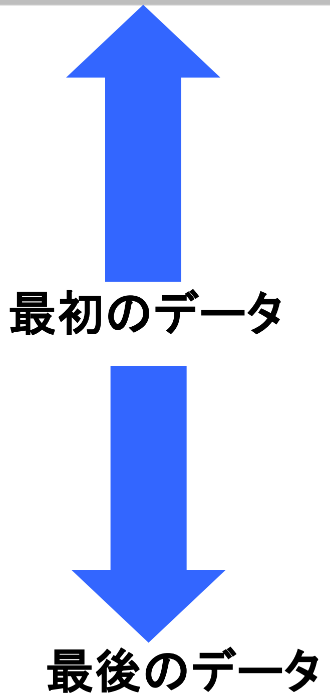
```
procedure TForm1.Action2Execute(Sender: TObject);
begin
  //前のデータへ
  ClientDataSet1.prior;
end;
```


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■Q4 Delphi/400 V2010 新機能紹介

– ジェスチャアプリケーション作成 –

既存アプリへの応用実装④




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

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


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

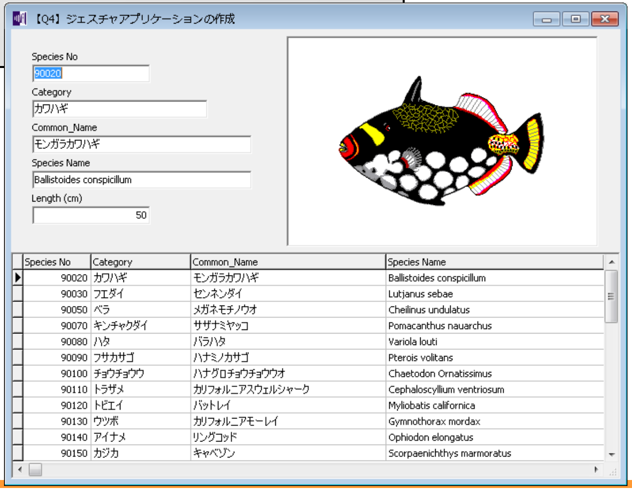
■Q4 Delphi/400 V2010 新機能紹介

－ ジェスチャアプリケーション作成 －

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

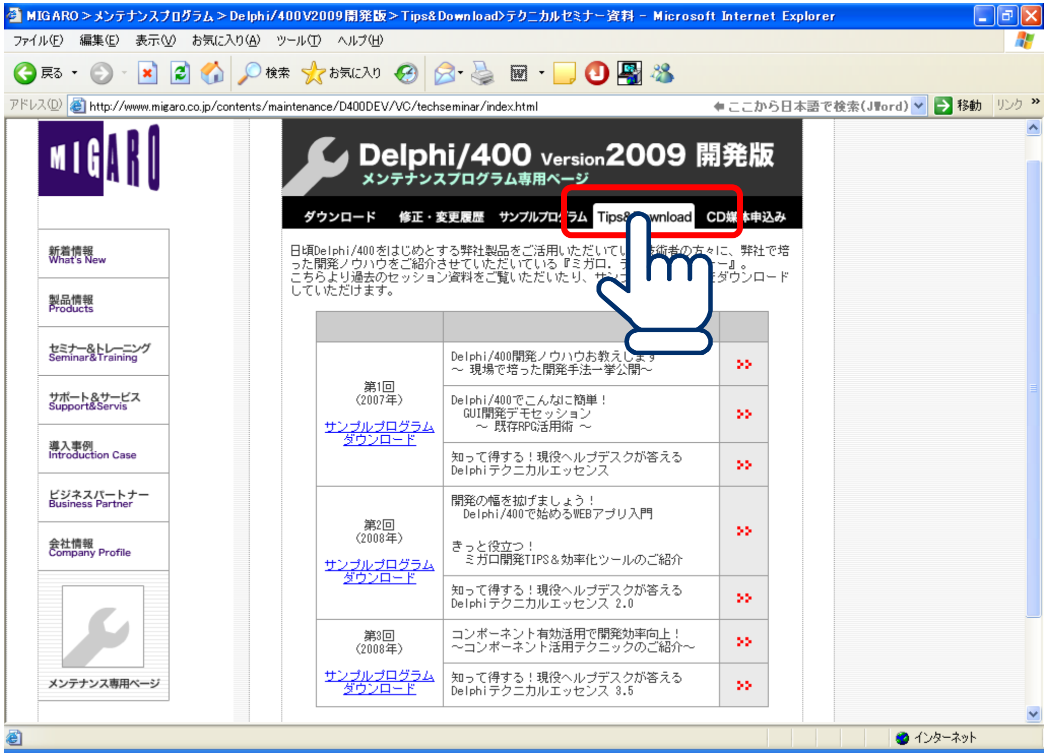
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

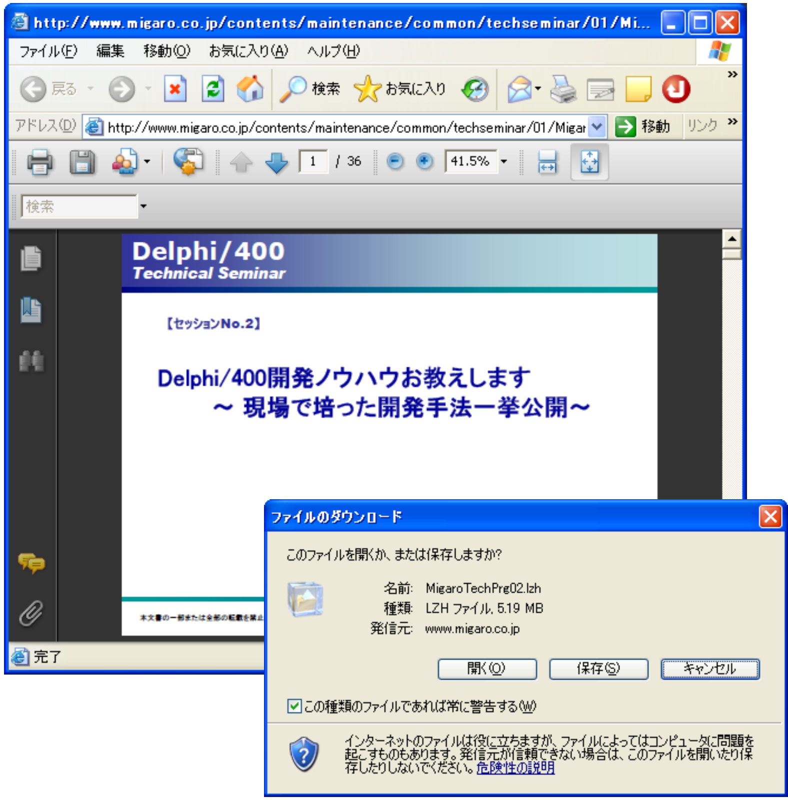
テクニカルサポートからのお知らせ

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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

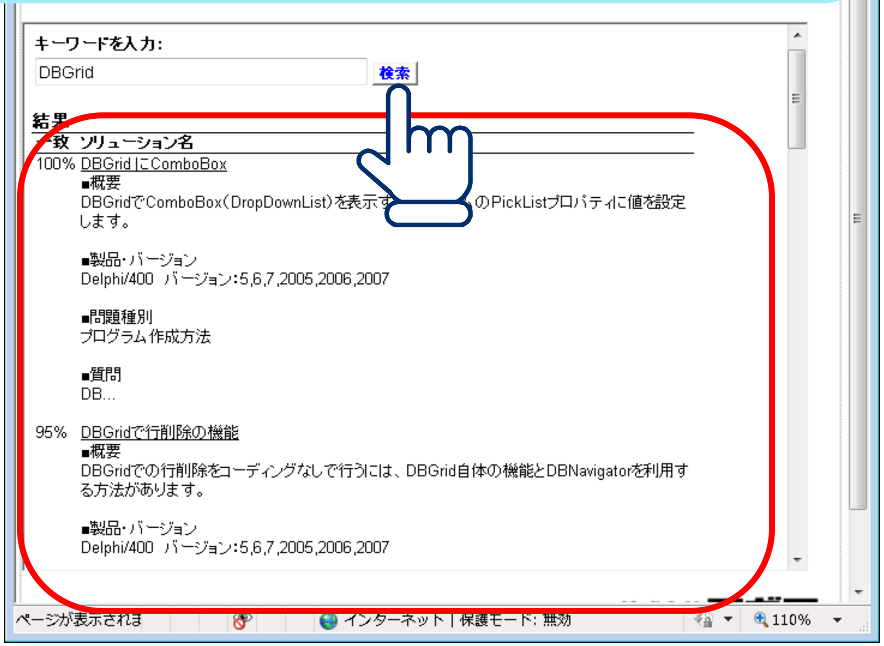




本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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 **1400件以上公開** しています
是非ご活用ください！



本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

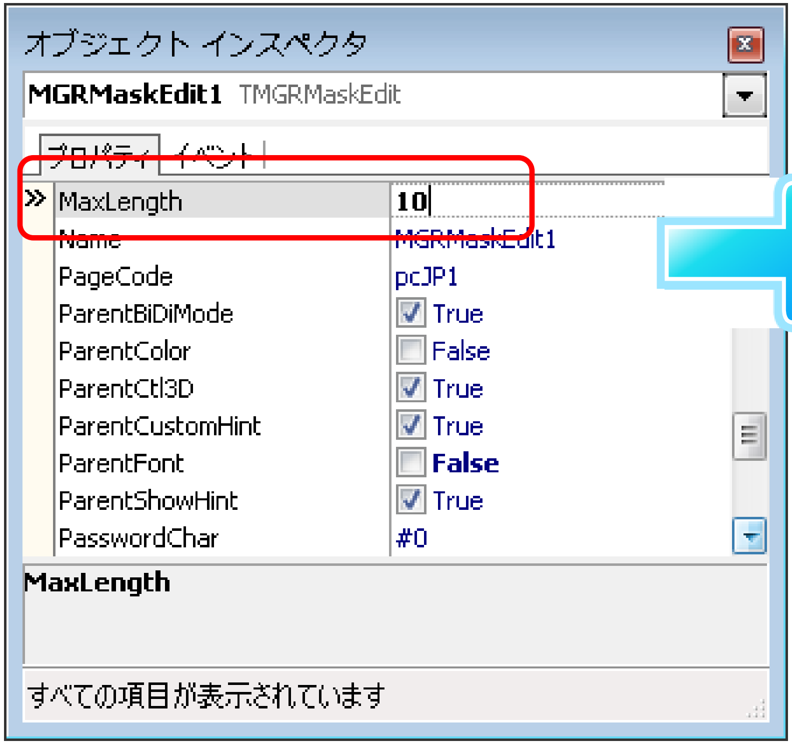
おまけ資料

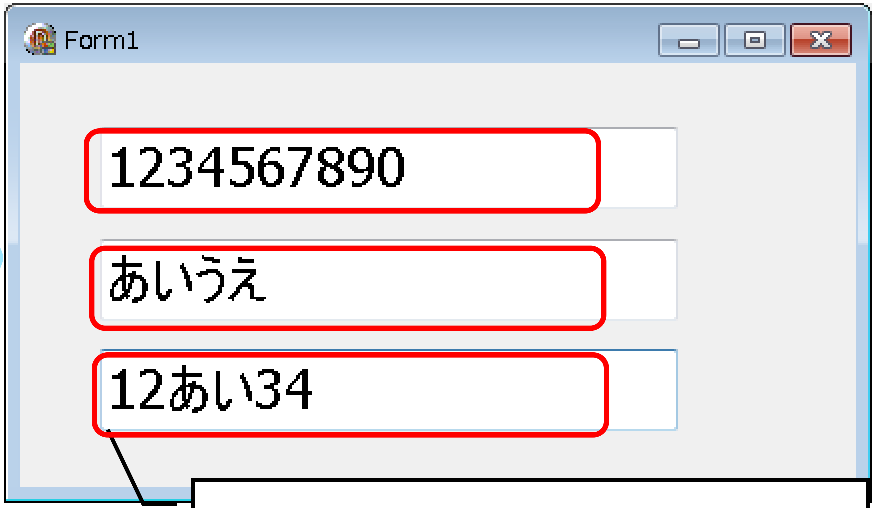
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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。

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

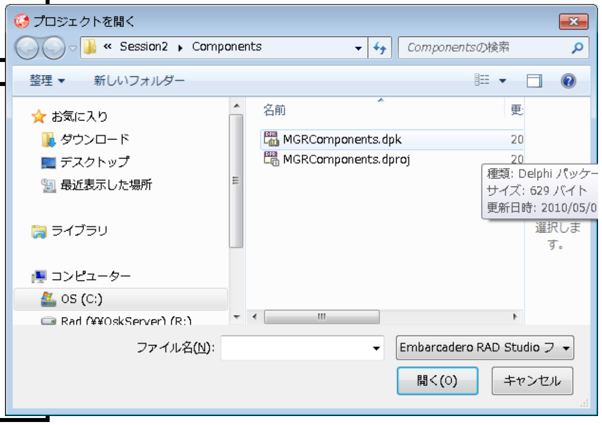
■おまけ資料

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

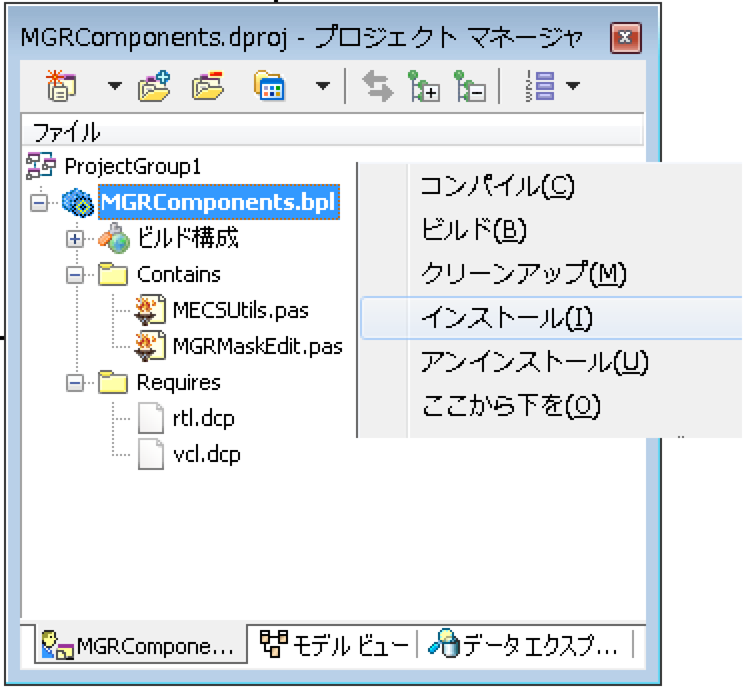


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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。

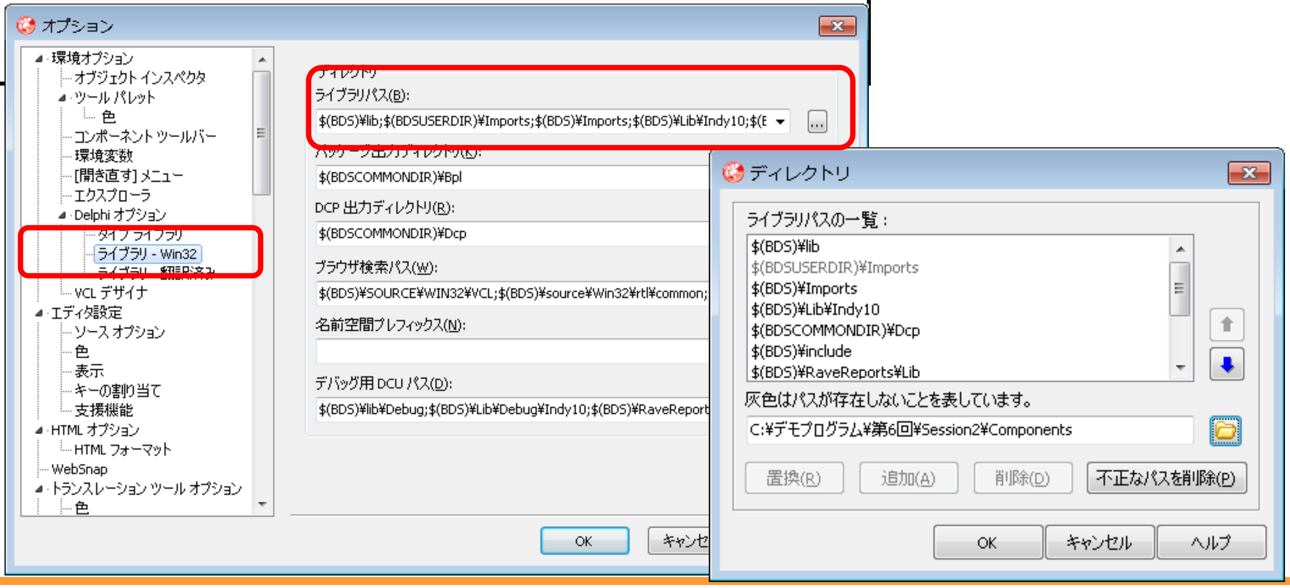


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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(AFnt)) 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

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

■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

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。