



【C2】 Delphiテクニカルセッション

「業務アプリ開発で活かせる Delphiプログラミングテクニック」

富永 英明 (a.k.a. DEKO)



はじめに

今回は？

- いつになく、ゆるい話です。
- 「あー、あるある (w)」という話です。
- アナタにとって「あー、あるあ...ねえよ！」とか、
「！？ヤベえ！」と冷や汗が出るような話ができるばいいなあ。
- タイトルとは関係のない話が含まれているかもしれません。
- サブタイトルとも関係のない話をするかもしれません、
そして内容は斜め上の話かもしれません。



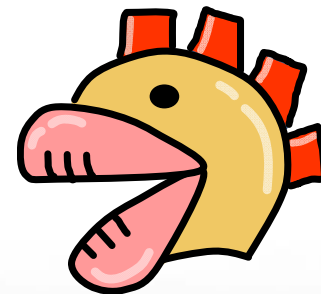
2

自分が持っていない環境
(で | を) テストする

Windows XP で動作する アプリケーションのデバッグは？

- Windows XP は 2014/04/08 でサポートが切れました。
- 「それでも現実問題として XP 上でアプリケーションを動作させなくてはならない事がある」
- XP ベースの Windows Embedded はまだサポートが切れていない。(2019/01/08 まで)
- でも、自分の開発環境は Windows 7 / Windows 8.x で、XP 固有の問題を再現できない。
- 実機を用意しようとしても、最近の PC はドライバがなかったりして XP がマトモに動作しない事がある。
- **さて…どうしましょう？**

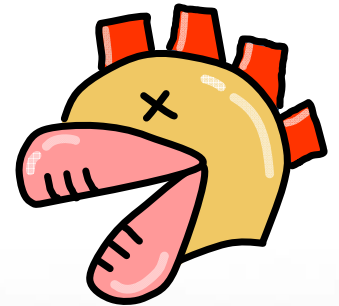
…どうしましょ



XP モード？ VMware？ VirtualBox？

- OS の仮想化を考えますよね？
- XP モードは **Windows 7 Professional** 以上でしか使えません。
- 当然、動作中は作業 PC が重くなります。
- PC を別に用意すれば重くなりませんが...OS どうしましょ？
Linux でやりますか？
- 先生。イチイチ別 PC で作業するのはメンドイです。

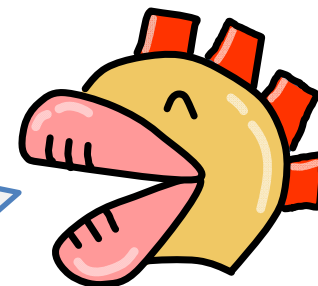
...どうしろと？



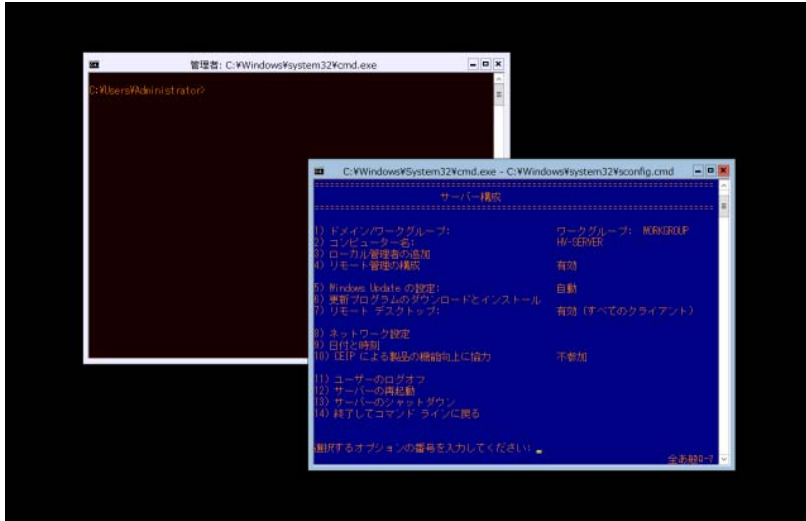
Hyper-V Server 2012 を 使えばいいじゃない？

- Hyper-V Server 2012 を使いましょう。
- 無償で使える**サーバ OS** です。
 - 64bit 対応 CPU (**必須**)
 - VT (Intel: VT-x、AMD: AMD-V) 対応 (**必須**)
 - DEP (Intel: XD、AMD: NX) 対応 (**必須**)
 - SLAT (Intel: EPT / VT-x2、AMD: NPT / RVI) 対応 (**推奨**...でも**必須**)
 - **メモリは積めるだけ積みましょう。**
- 安い PC で一台デッチ上げましょう。
PC が壊れても OS イメージさえ無事なら復旧は容易です。

VPC の VHD が
そのまま使えるヨ

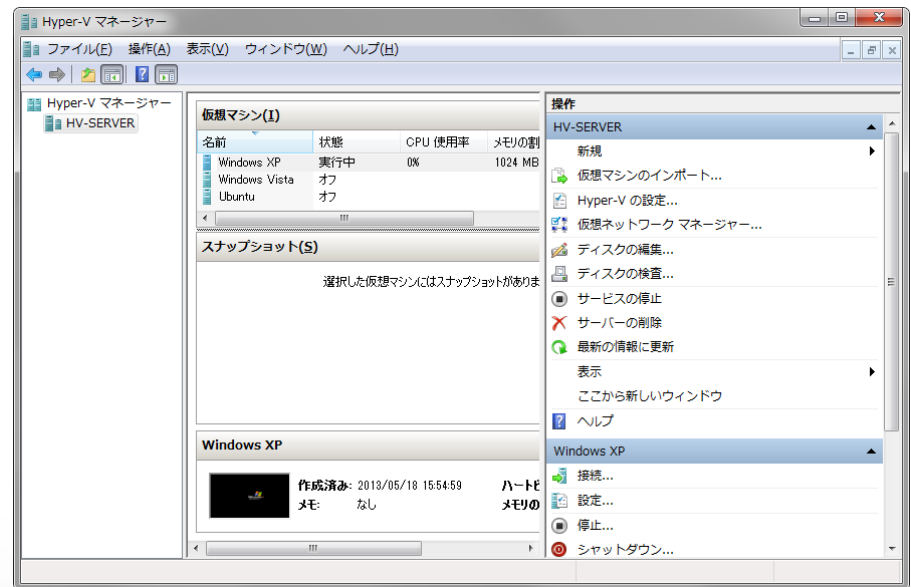


Hyper-V Server 2012 ってこんなの (1)



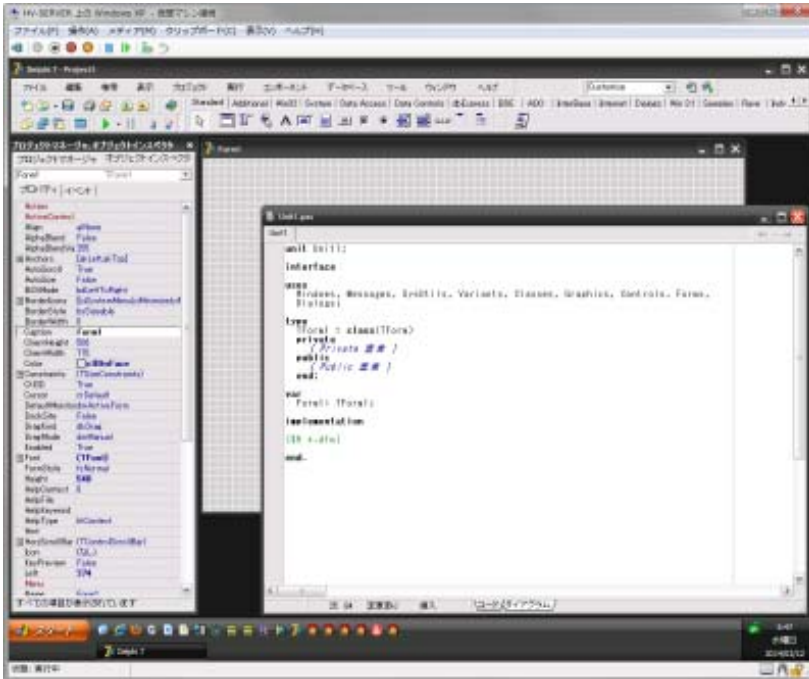
Hyper-V Server 2012 の
スクリーンショットです。
GUI が使えないので地味な印象です。

Hyper-V の管理コンソールです。
クライアント PC にインストールし
て使います。このコンソールから
ゲスト OS を操作できますし、
リモートデスクトップからも
接続できます。



Hyper-V Server 2012 ってこんなの (2)

Hyper-V Server 2012 の
ゲスト OS (XP) 上で
Delphi 7 を動作させています。

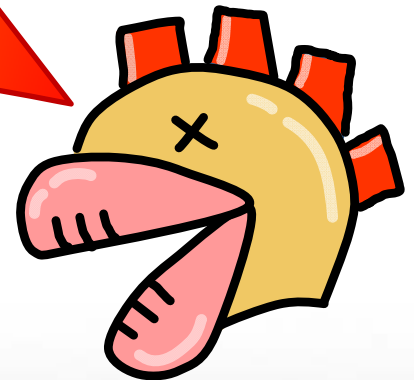


Windows 8 に
Android 用 RDP クライアントで
接続している所です。



それでもこんな時は困る。

お客さんトコの XP でしか
現象が出ないんだけど！



リモートデバッガを使いましょう。



お客さんトコで LAN 接続させてもらって、
リモートデバッガでデバッグしましょう。
Delphi のアクティベーション残数を
減らすことなくデバッグできます。
Delphi 7 Professional 以降 (それ以前は上
位版に付属)、**2005 以外**には
リモートデバッガが付属しています。

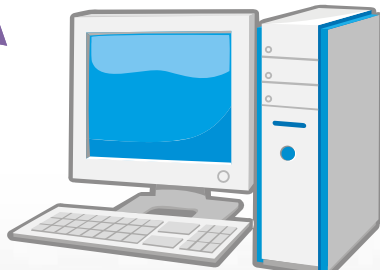


リモートデバッガを
インストール



Delphi
インストール済

実稼動 PC



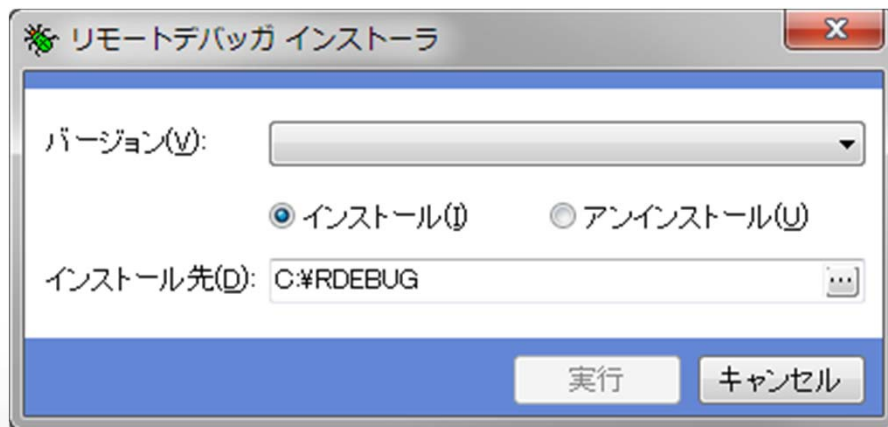
リモートデバッグ



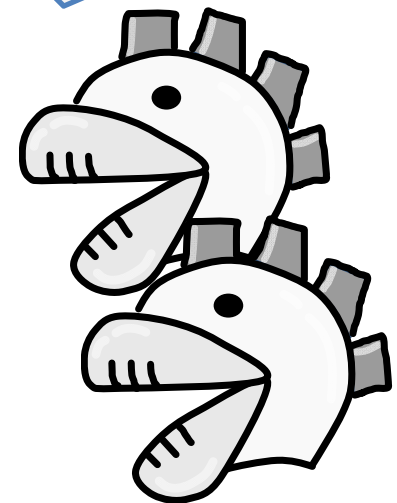
持ち込み PC

先生！リモートデバッガの 使い方がわかりません！！

- 「この余白はそれを書くには狭すぎる」
- Delphi 7 ～ XE のリモートデバッガの使い方
<http://ht-deko.minim.ne.jp/tech031.html>
- XE2 ～ のリモートデバッガ (PAServer) の使い方
<http://ht-deko.minim.ne.jp/tech071.html>
- リモートデバッガインストーラ
<http://ht-deko.minim.ne.jp/tech053.html#RMDINST>



投げやがった…



逃げやがった…

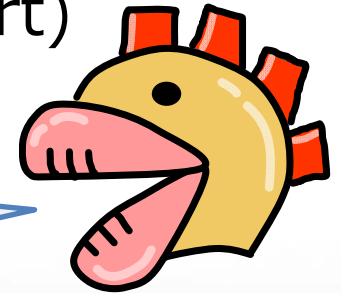


使える武器は多いに
越したことはない

Windows Vista (1)

- Unicode (UTF-16) のサロゲートペア - 2009
- メイリオフォント (英語圏だと Segoe フォント)
- DWM (Desktop Window Manager)
- Direct 2D - XE2 (FireMonkey)
- Direct Write - XE2 (FireMonkey)
- TSF (Text Services Framework) /
CUAS (Cicero Unaware Application Support)

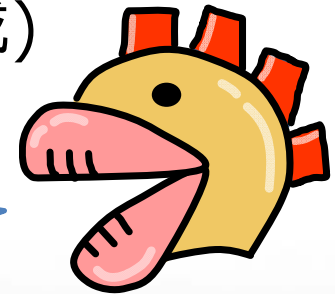
Unicode 関連は以前やったよネ？



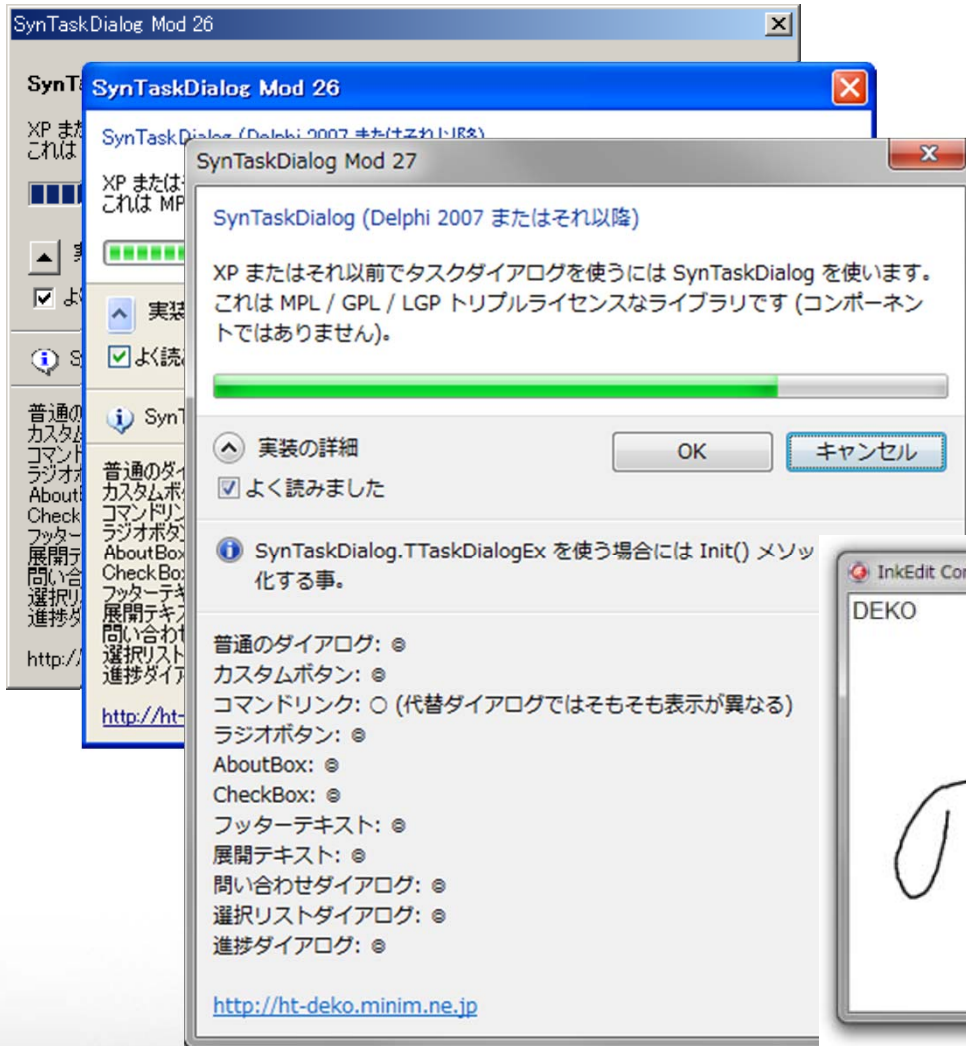
Windows Vista (2)

- PNG アイコン - XE
- タスクダイアログ - 2007 (できれば XE2 以降で)
- コマンドリンクボタン - 2009
- コンボボックスのテキストヒント - 2009
- Ink Edit (RichEdit のスーパーセット)
- フリック (WM_TABLET_FLICK) - 2010
- Microsoft Speech Platform (音声認識・合成)
- タスクスケジューラ 2.0

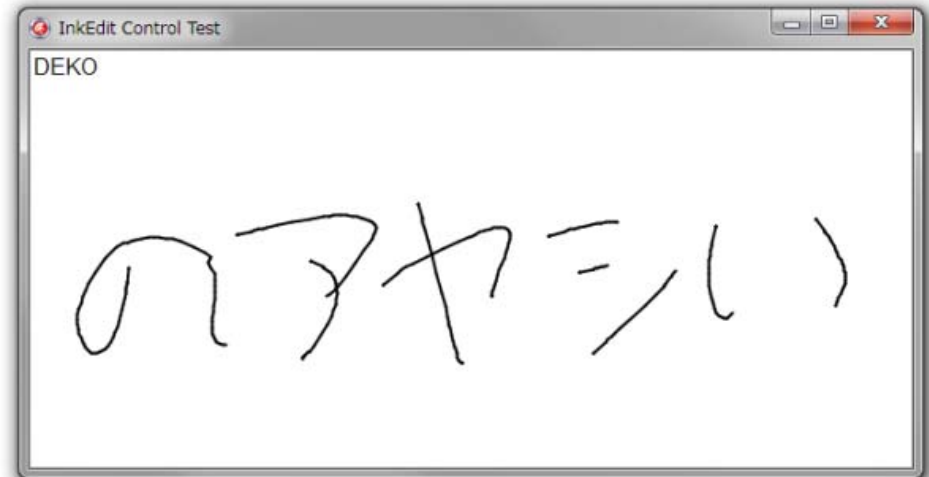
コレやってないわ。



タスクダイアログと InkEdit



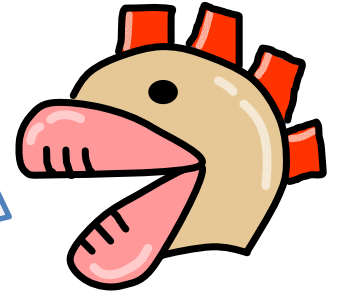
左のタスクダイアログは OS
ネイティブのものではなく、
SynTaskDialog という互換ダ
イアログで Windows 2000
でも使えます。
InkEdit は見ての通り手書き
認識ができる RichEdit です。
(但し 32bit 専用)



Windows 7

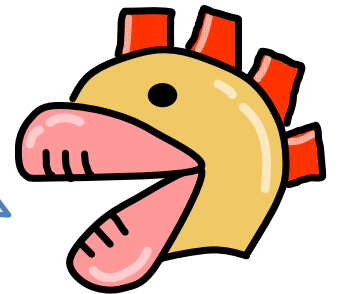
- タスクバーボタン - XE6
- ジャンプリスト
- センサー API - XE3
- ロケーション API - XE3
- ジェスチャ (WM_GESTURE) - 2010
- タッチ (WM_TOUCH) - 2010

ジャンプリストは
過去のデブキャンで
細川さんがやってたヨ



センサー & ロケーションは VCL でも使える。
タッチとジェスチャの話は
"スレート PC プログラミング" で。

<http://ht-deko.minim.ne.jp/tw317/#05>





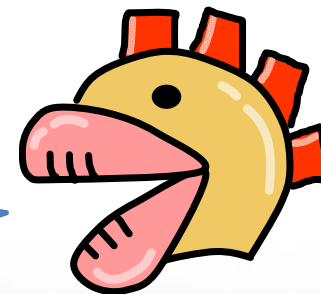
4

管理者の権限が必要だ！

管理者の権限が必要だ！

- BDE はインストールするのにも**管理者権限**が必要。
(インストーラ側で対処していなければ)
- BDE Administrator は**管理者権限**で実行する必要がある。
コントロールパネルの**アプレット**からは**実行できない**。
- マニフェストの存在しないアプリケーションは
バーチャルストアの影響を受けるため、ファイルやレジストリが
別の場所へリダイレクトされてしまう。
- UAC そのものについては過去のデブキャン
資料を参照の事。

おおコワイコワイ



UAC と requestedExecutionLevel の関係

- マニフェストなし

バーチャルストアの影響を受ける。標準ユーザ権限で動作する。

- **asInvoker**

標準ユーザ権限で動作する。Delphi のマニフェストのデフォルトはコレ。

- **highestAvailable**

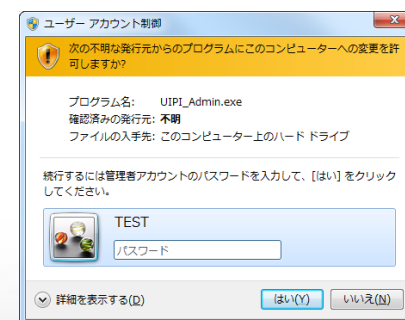
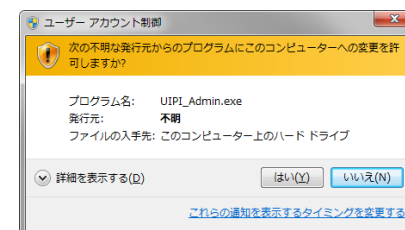
アカウントが管理者であれば管理者権限への昇格を促す。

標準ユーザだとそのまま実行される。

- **requireAdministrator**

アカウントが管理者であれば管理者権限への昇格を促す。

標準ユーザだと管理者アカウントの PW を要求される。

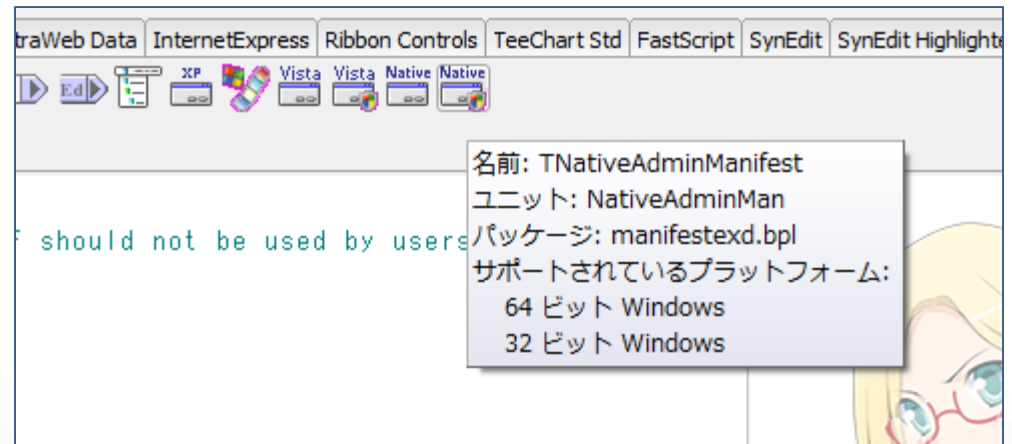


何はともあれ

- マニフェストファイルを追加しましょう。
追加しないと先述の通り、面倒な事になります。
- Manifest Ex コンポーネントをインストールすると楽ですよ。
(Delphi 6 ～)

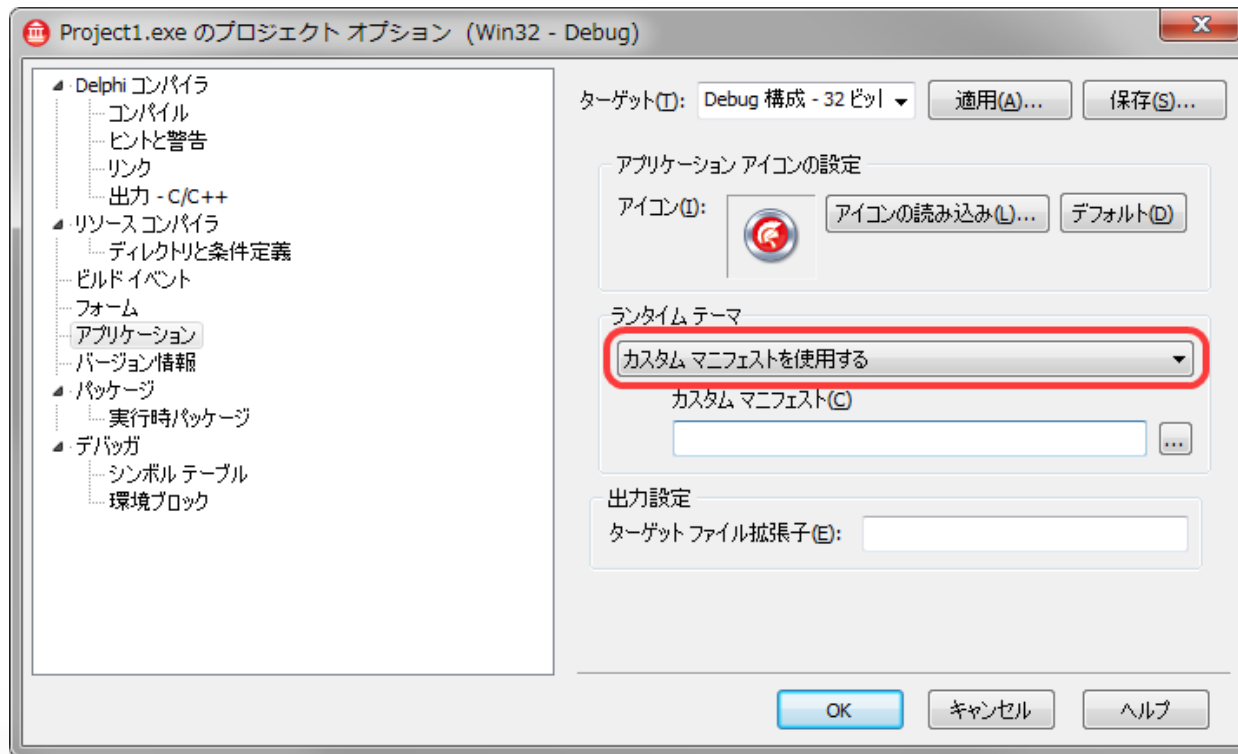
<http://ht-deko.minim.ne.jp/delphiforum/?vasthtmlaction=viewtopic&t=1099>

目的に応じた
コンポーネントを
貼るだけです。
(VCL / FMX 両対応)



カスタムマニフェスト

- **XE2** 以降だとプロジェクトオプションでカスタムマニフェストを指定する事ができます。



カスタムマニフェストといえば...

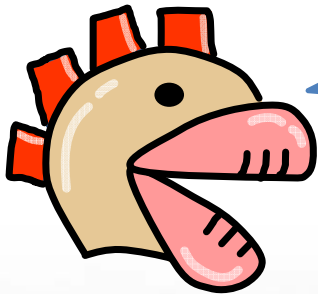
- XML に以下の compatibility 要素がない場合、アプリケーションは **Vista 互換モード** で動作します。

```
<compatibility xmlns="urn:schemas-microsoft-com:compatibility.v1">
  <application>
    <!--Windows 8.1-->
    <supportedOS Id="{1f676c76-80e1-4239-95bb-83d0f6d0da78}" />
    <!--Windows 8-->
    <supportedOS Id="{4a2f28e3-53b9-4441-ba9c-d69d4a4a6e38}" />
    <!--Windows 7-->
    <supportedOS Id="{35138b9a-5d96-4fbd-8e2d-a2440225f93a}" />
    <!--Windows Vista-->
    <supportedOS Id="{e2011457-1546-43c5-a5fe-008deeee3d3f0}" />
  </application>
</compatibility>
```

CPU ■ CPU 使用率 20% ■ 最大周波数 100%							
イメージ	PID	説明	状態	スレッド	CPU	平均 CPU	オペレーティング
GoogleIMEJaRenderer.exe	6736	Google 日本語入力 レンダ...	実行中	4	0	0.00	Windows 7
HydraDM.exe	4104	HydraDM	実行中	1	0	0.00	Windows Vista
IBConsole.exe	5468	IBConsole.exe	実行中	5	0	0.02	Windows Vista
IMEDICTUPDATE.EXE	2612	Microsoft Office IME 2010	実行中	4	0	0.00	Windows Vista
InputPersonalization.exe	7960	入力の個人用設定サーバー	実行中	8	0	0.00	Windows 7

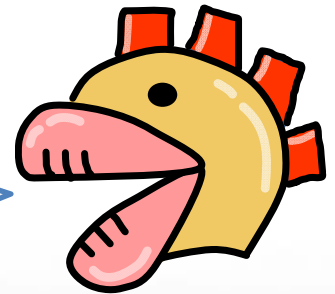
カスタムマニフェストのデフォルト

- IDE が生成するアプリケーションマニフェストのデフォルト設定を変更するには、\$(BDS)¥bin にある default_app.manifest を書き換えます。
- デフォルトのマニフェストのままだと、Windows 7 / 8.x でアプリケーションを実行しても **Vista 互換モード**になります。
- **XE2 以降**でのみ使える手法です。



Vista 互換モードだと何がイカンの？

プログラム互換性アシスタントの影響を受けるんだってさ。



プログラム互換性アシスタント (PCA)

- ざっくり言うと「このプログラムは正しくインストールされなかった可能性があります」とか出るアレです。
- Windows 7 だと自動でエアロをオフにしてくれちゃうアレです。
- コレのお世話になるという事は、
IE で言うなら互換表示を使っているようなものです。
- PCA が介入する以上、速度低下のペナルティがある可能性は否定できません。
- PCA で動作しても、問題を先送りしているに過ぎないので、ネイティブモードのマニフェストを組み込んで、問題点を顕在化させるべきだと思います。

アヤしいアプリは管理者権限で実行！

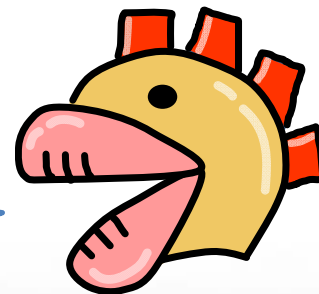
- ...すればいいってもんじゃないありません。

ユーザー インターフェイス特権の分離 (UIPI)

(前略) より具体的に言うと、より高い権限が必要なアプリケーションが `ChangeWindowMessageFilter()` を呼び出してメッセージを明示的に許可しない限り、権限モードが低いアプリケーションは、一般的に、より高い権限が必要なアプリケーションに **メッセージを送信することはできません**。同様に、必要な権限がより低いアプリケーションは、必要な権限がより高いアプリケーションが所有する HWND を読むことはできますが、**変更はできません**。

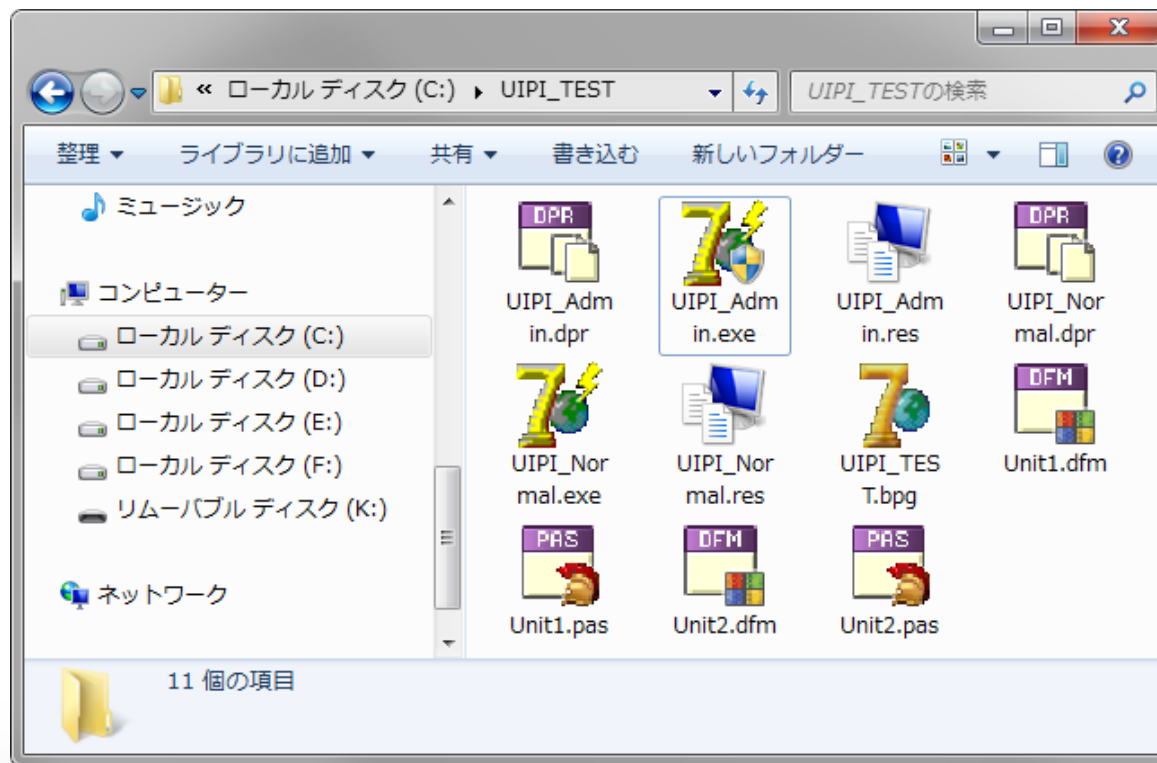
<http://msdn.microsoft.com/ja-jp/library/aa480152.aspx#EKFAG>

ワケワカラン



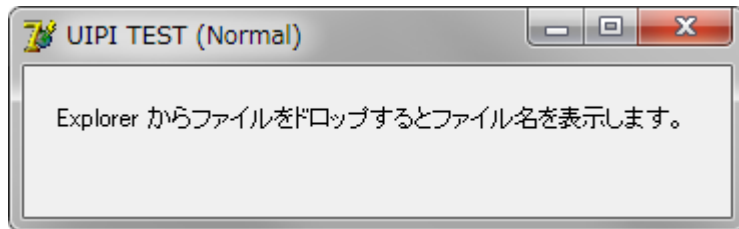
論より証拠

- 通常のマニフェストと管理者権限マニフェストを組み込んだテストアプリケーションを作成 (Delphi 7)



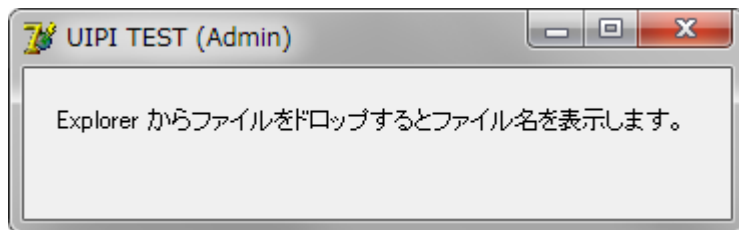
機能は見ての通り

- 普通のマニフェストを組み込んだもの (UIPI_Normal.exe)

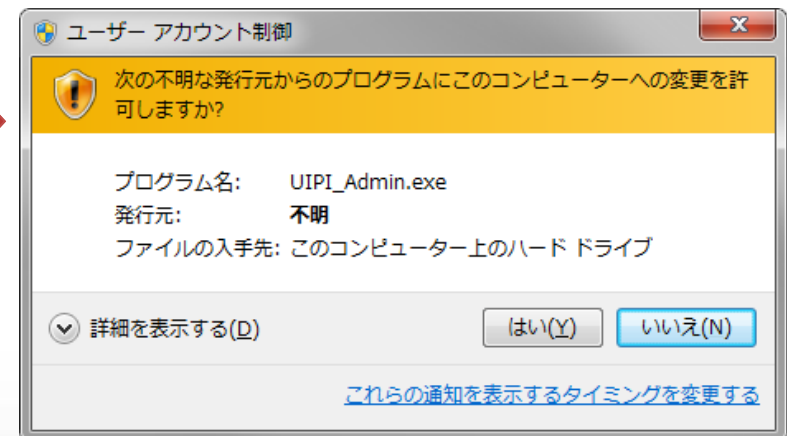


※ NativeMan 使用

- 管理者権限マニフェスト を組み込んだもの(UIPI_Admin.exe)



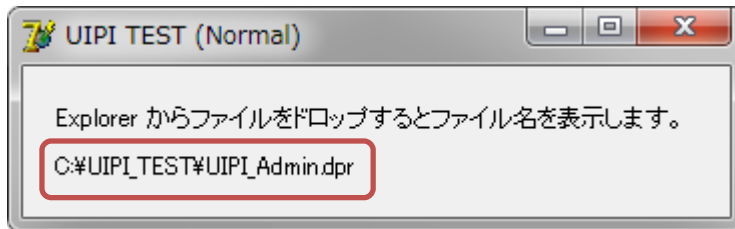
※ NativeAdminMan 使用



昇格を要求される

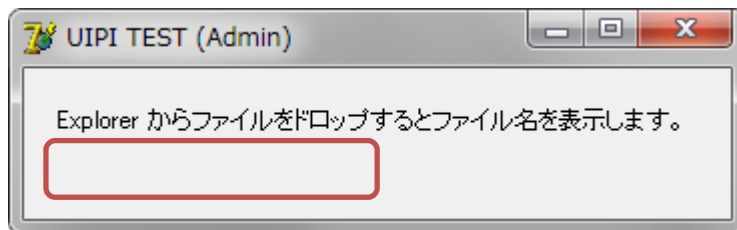
Explorer からファイルをドロップすると？

- 普通のマニフェストを組み込んだもの (UIPI_Normal.exe)



普通に表示される

- 管理者権限マニフェスト を組み込んだもの (UIPI_Admin.exe)



何も表示されず、
エラーにもならない！

Explorer が標準ユーザ権限、アプリケーションが管理者権限で
実行されているためにこのような現象が起こります。

ChangeWindowMessageFilter() API の使い方

```
procedure TForm2.FormCreate(Sender: TObject);
const
  MSGFLT_ADD      = 1;
  MSGFLT_REMOVE  = 2;
  WM_COPYGLOBALDATA = $0049;
var
  pa: TFarProc;
  DLLWnd: THandle;
  ChangeWindowMessageFilter: function (message: UINT; dwFlag: DWORD): BOOL; stdcall;
begin
  // UIPI 対策
  if CheckWin32Version(6, 0) then
    begin
      DLLWnd := LoadLibrary('user32.dll');
      if DLLWnd > 0 then
        begin
          pa := GetProcAddress(DLLWnd, 'ChangeWindowMessageFilter');
          if pa <> nil then
            begin
              @ChangeWindowMessageFilter := pa;
              ChangeWindowMessageFilter(WM_DROPFILES, MSGFLT_ADD);
              ChangeWindowMessageFilter(WM_COPYDATA, MSGFLT_ADD);
              ChangeWindowMessageFilter(WM_COPYGLOBALDATA, MSGFLT_ADD);
            end;
          end;
        end;
      end;
  // ドロップ受け入れ
  DragAcceptFiles(Handle, True);
end;
```

この API は Vista 以降で
実装されているので、
LoadLibrary で利用する。

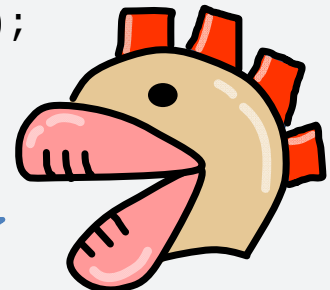
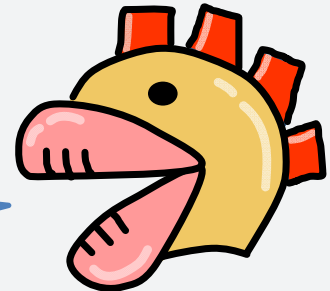
念の為に...

- メッセージハンドラ部分

```
...  
public  
  { Public 宣言 }  
  procedure WMDropFiles(var msg: TWMDropFiles); message WM_DROPFILES;  
end;  
  
procedure TForm2.WMDropFiles(var msg: TWMDropFiles);  
var  
  FileName: array [0..32768] of Char;  
begin  
  Label2.Caption := '';  
  try  
    // 単一ファイルしか受け入れない手抜きコードなので注意  
    DragQueryFile(msg.Drop, 0, FileName, SizeOf(FileName));  
    Label2.Caption := StrPas(FileName);  
  finally  
    DragFinish(msg.Drop);  
  end;  
end;
```

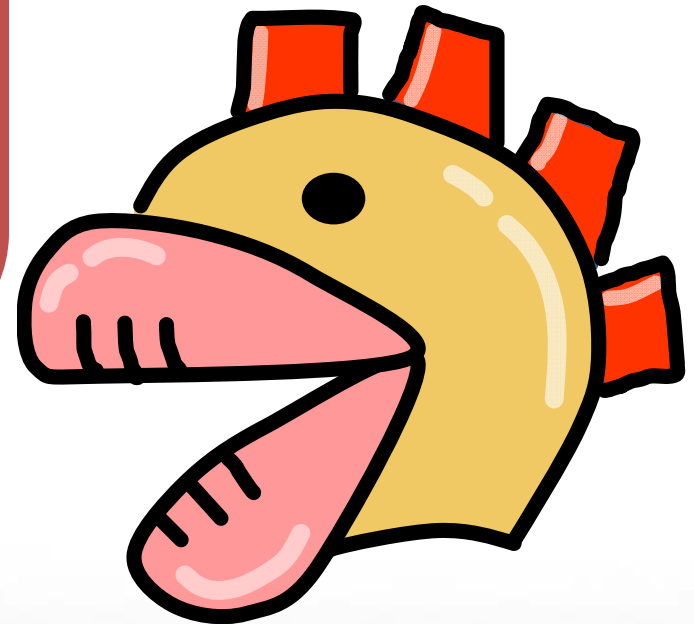
えらくデカイバッファだな

MAX_PATH 定数を
知らないんじゃない？



管理者権限へ昇格すればいいってもんじゃない。

管理者権限へ
昇格させるのはいいけど、
面倒な処理を余計に
書かなきゃいけない事が
あるヨ。



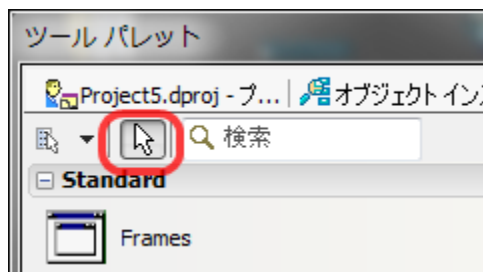
※安易に管理者権限へ昇格させちゃダメ。



大量のコントロールを
貼るのが憂鬱で仕方ない

コンポーネントは連続貼り付けできる

- コンポーネントパレットあるいはコンポーネントツールバーのコンポーネントを **Shift** を押しながらクリックし、フォームデザイナーの上でマウスのボタンを連打！！
- 連続貼り付けモードを解除するには、コンポーネントパレットあるいはコンポーネントツールバーのマウスカーソルの形をしたボタンをクリックしてボタンが引っ込んだ状態にする。



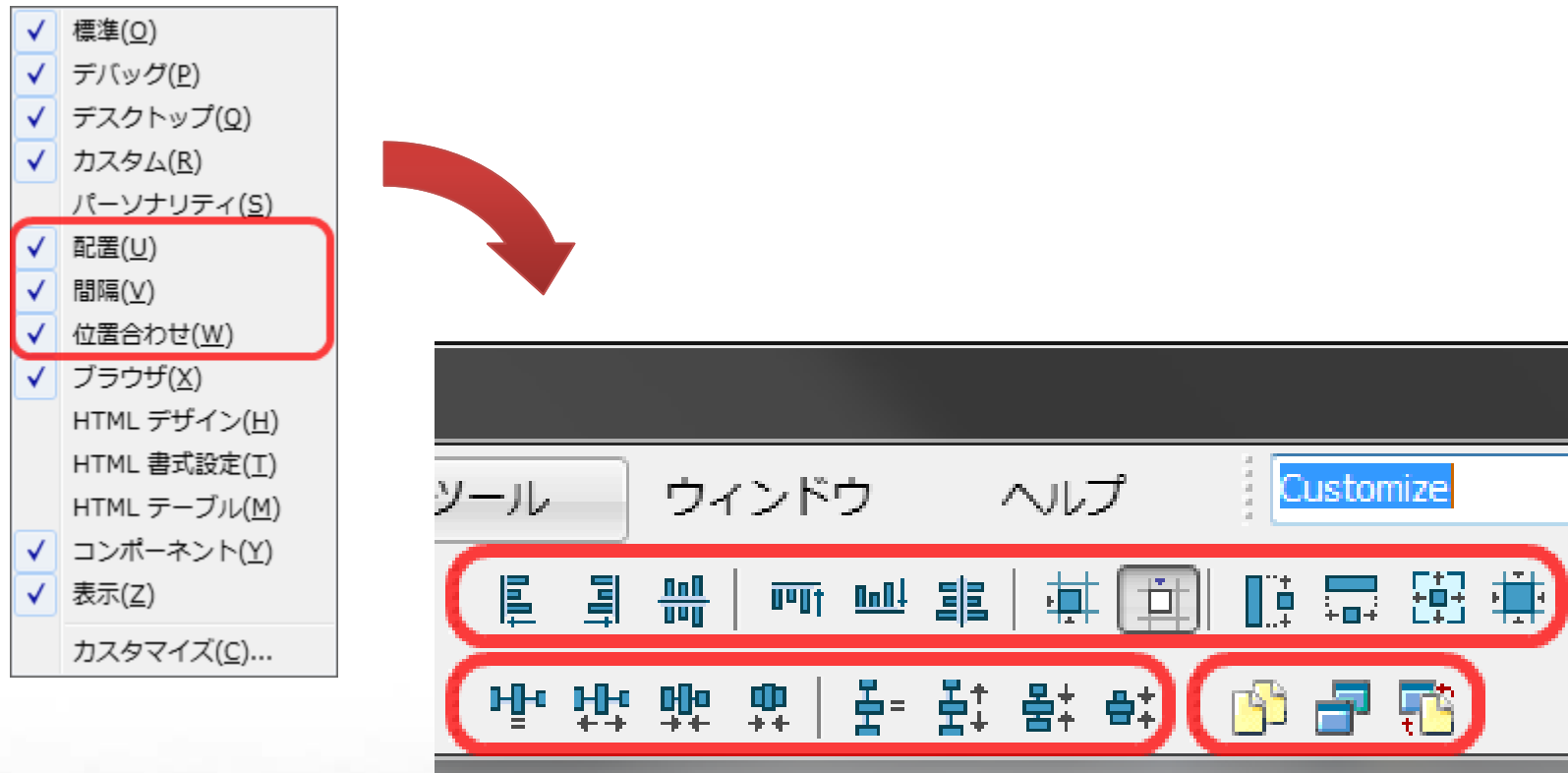
コンポーネントパレット



コンポーネントツールバー

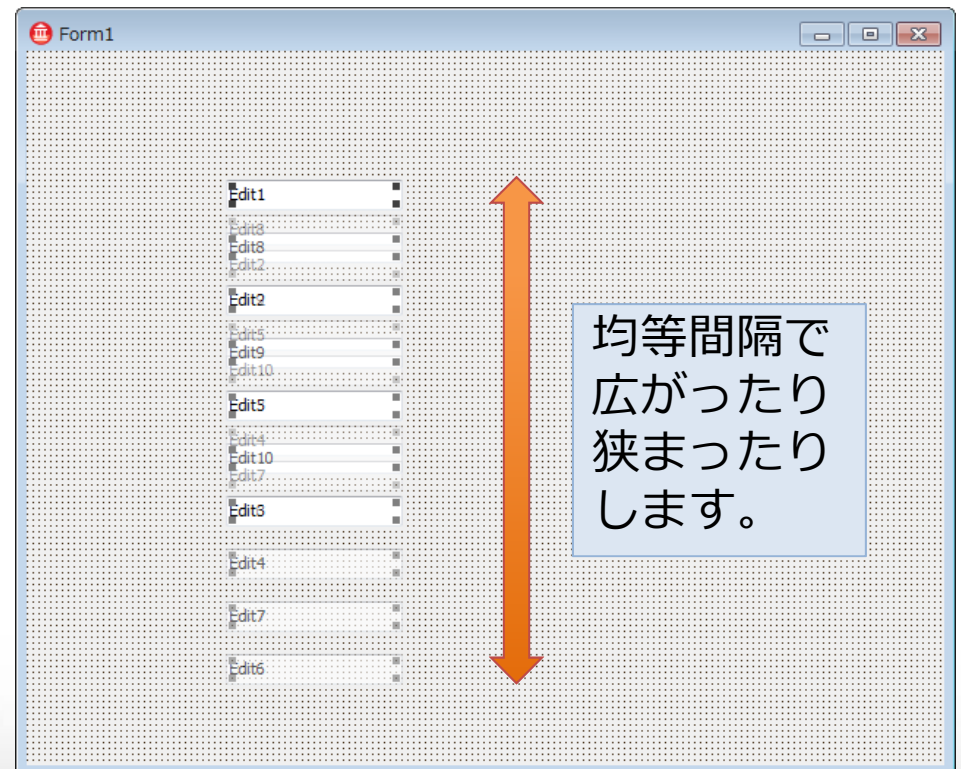
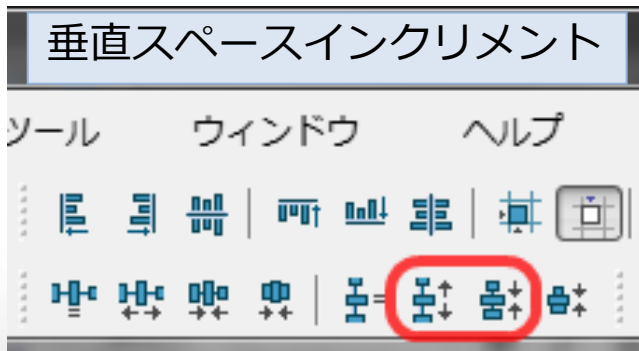
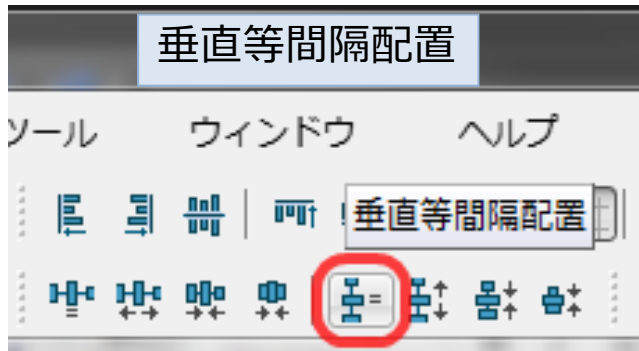
コンポーネントの整列がメンドイ

- ツールバーを右クリックし、「配置、間隔、位置合わせ」のツールバーを表示させましょう。



そして...

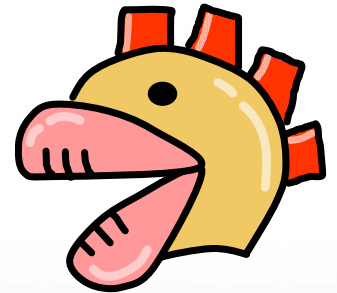
この機能は **Delphi 2005** からあります。



幅とかの設定は...

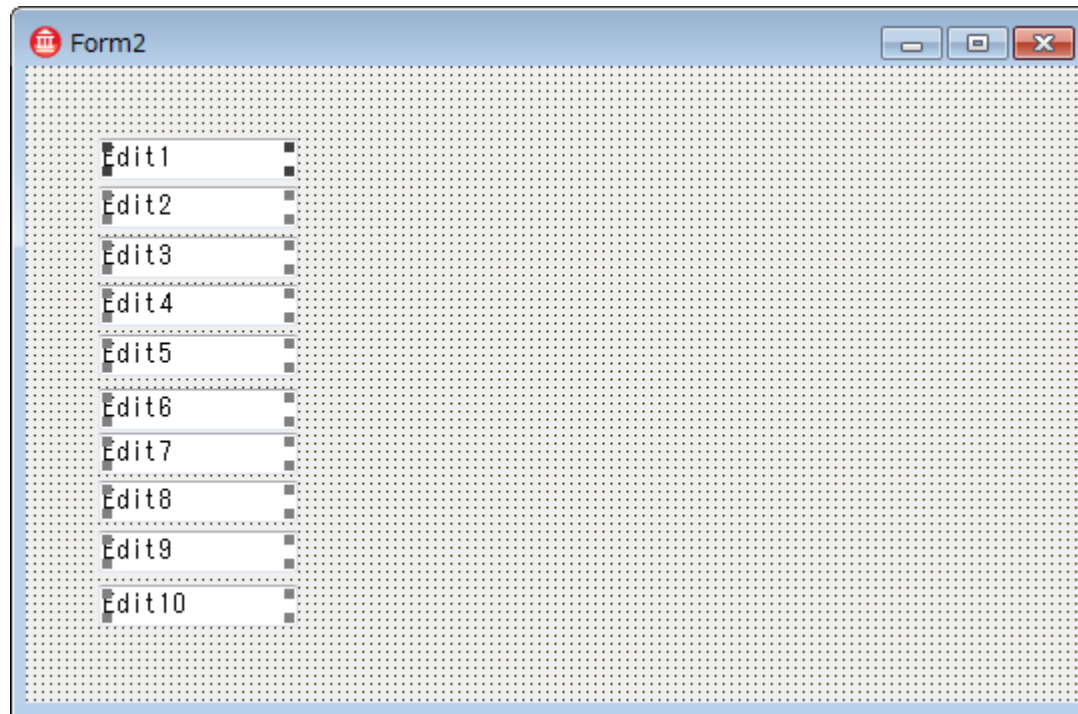
- 「さっきのって **XE 4 以前の FMX フォーム**ではできないよね？」
(XE5 以降または AppMethod では可能です)
- 「幅はどうするの？」
- まず “**Value Calc プロパティエディタ**” をインストールします。
<http://ht-deko.minim.ne.jp/delphiforum/?vasthtmlaction=viewtopic&t=1349>
(**Delphi 6** 以降。フル機能は **2010** 以降)

なんぞそれ？



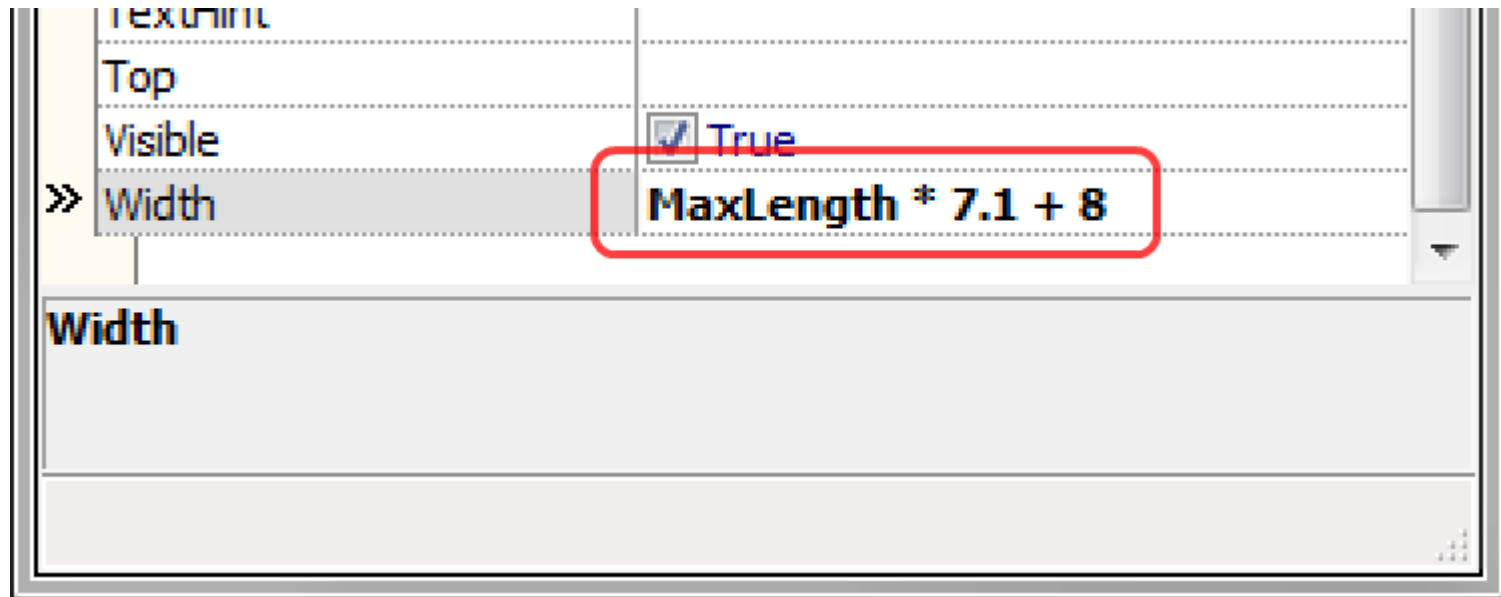
コントロールの幅を一気に設定する (1)

- MaxLength を設定し、コントロールを選択します。
Edit のフォントは MS ゴシック 10pt とします。



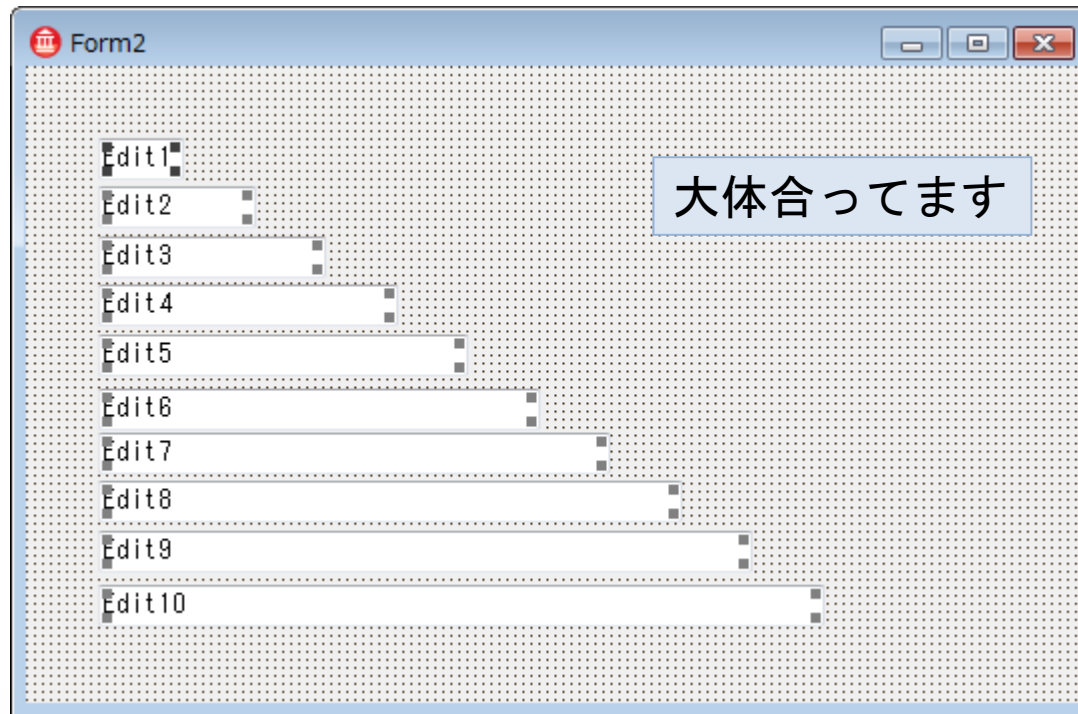
コントロールの幅を一気に設定する (2)

- オブジェクトインスペクタの Width プロパティに
 $\text{MaxLength} * 7.1 + 8$
と入力し、Enter を押します。



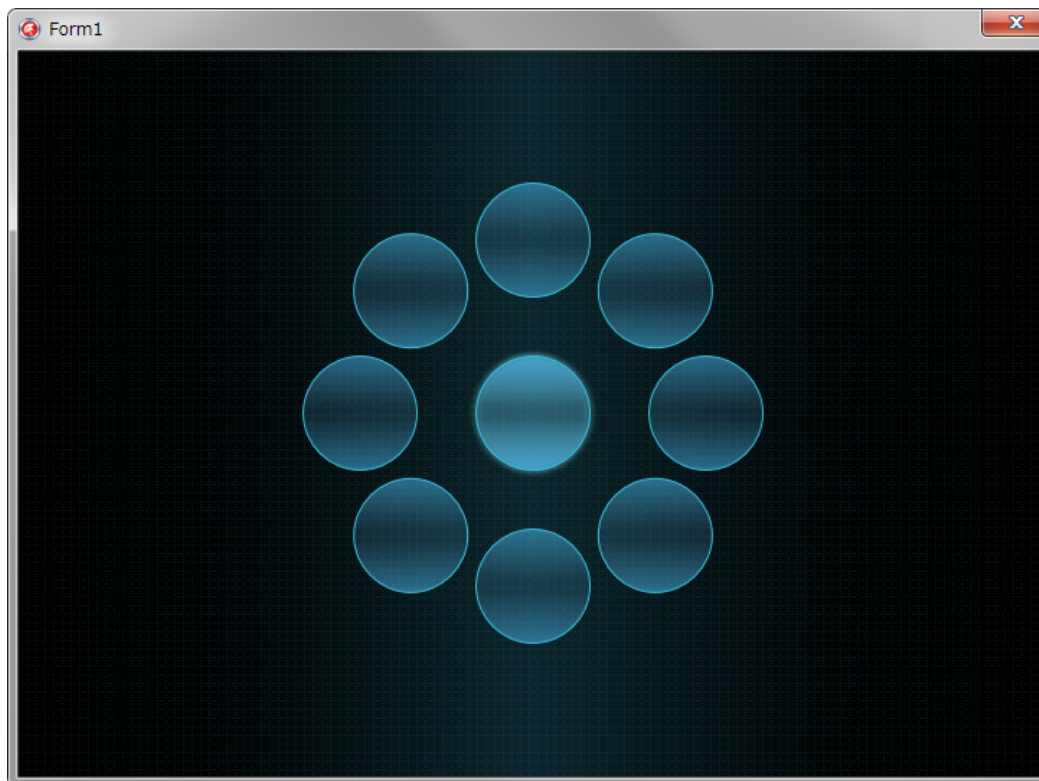
コントロールの幅を一気に設定する (3)

- すると、こうなります (ドヤア)
- 先程やった位置合わせも計算で行う事ができます。



余談ですが...

- プロパティでの計算は VCL / FMX フォームいずれも可能。
三角関数を使えば以下のような配置を設計時に行えます。



関数	引数	戻り値の型	説明
Abs	Float	Float	引数に与えられた実数を絶対値で返します
ArcCos	Float	Float	アークコサイン
ArcCosh	Float	Float	ハイパボリックアークコサイン
ArcCot	Float	Float	アークコタンジェント
ArcCotH	Float	Float	ハイパボリックアークコタンジェント
ArcCsc	Float	Float	アークコセカント
ArcCscH	Float	Float	ハイパボリックアークコセカント
ArcSec	Float	Float	アークセカント
ArcSecH	Float	Float	ハイパボリックアークセカント
ArcSin	Float	Float	アークサイン
ArcSinh	Float	Float	ハイパボリックアークサイン
ArcTan	Float	Float	アークタンジェント
ArcTanh	Float	Float	ハイパボリックアークタンジェント
Cos	Float	Float	コサイン
Cosh	Float	Float	ハイパボリックコサイン
Cot	Float	Float	コタンジェント
CotH	Float	Float	ハイパボリックコタンジェント
Csc	Float	Float	コセカント
CscH	Float	Float	ハイパボリックコセカント
CycleToDeg	Float	Float	周から角度に変換します
CycleToGrad	Float	Float	周からグラードに変換します
CycleToRad	Float	Float	周からラジアンに変換します
DegToCycle	Float	Float	角度から周に変換します
DegToGrad	Float	Float	角度からグラードに変換します
DegToRad	Float	Float	角度からラジアンに変換します
Float	Integer	Float	引数に与えられた整数を実数にして返します
Exp	Float	Float	e の <引数> 乗を返します
GradToCycle	Float	Float	グラードから周に変換します
GradToDeg	Float	Float	グラードから角度に変換します
GradToRad	Float	Float	グラードからラジアンに変換します
Int	Float	Float	引数に与えられた実数の整数部分を返します
Ln	Float	Float	<引数> の自然対数を返します
LnXP1	Float	Float	<引数>+1 の自然対数を返します
Log10	Float	Float	10 を底とする <引数> の対数を返します
Log2	Float	Float	2 を底とする <引数> の対数を返します
Pi	(なし)	Float	円周率。3.14... を返します
Sec	Float	Float	セカント
Secant	Float	Float	セカント
SecH	Float	Float	ハイパボリックセカント
SelectOrder	(なし)	Float	コンポーネントの選択順序を 0 ベースの実数で返します
Sin	Float	Float	サイン
Sinh	Float	Float	ハイパボリックサイン
Sqr	Float	Float	<引数> の 2 乗を返します
Sqrt	Float	Float	<引数> の平方根を返します
Tan	Float	Float	タンジェント
Tanh	Float	Float	ハイパボリックタンジェント

貼ったのはいいけど変数への出し入れが...

- 趣旨が違っているような...まあいいです。
- **Delphi XE3** 以降には**プリミティブ型に対するヘルパー**がありますので、それを使えば簡単です。

```
var  
    Value: Integer;  
begin  
    Value := 100;  
    Edit1.Text := Value.ToString;  
end;
```

それじゃ、逆方向は無理だよね？

- ...であれば、TCustomEdit 用のクラスヘルパーを作ればいいのか？

```
type
  { TCustomEditExtention }
TCustomEditExtention =
  class helper for TCustomEdit
  private
    ...
  published
    // Properties
    property AsBoolean: Boolean read GetBoolean write SetBoolean;
    property AsCurrency: Currency read GetCurrency write SetCurrency;
    property AsDate: TDate read GetDate write SetDate;
    property AsDateTime: TDateTime read GetDateTime write SetDateTime;
    property AsFloat: Double read GetFloat write SetFloat;
    property AsInt64: Int64 read GetInt64 write SetInt64;
    property AsInteger: Integer read GetInteger write SetInteger;
    property AsString: string read GetString write SetString;
    property AsTime: TTime read GetTime write SetTime;
  end;
```

そうすると...

- こんな感じで使えます。
何度も何度も型変換関数を通さなくて済みますね。

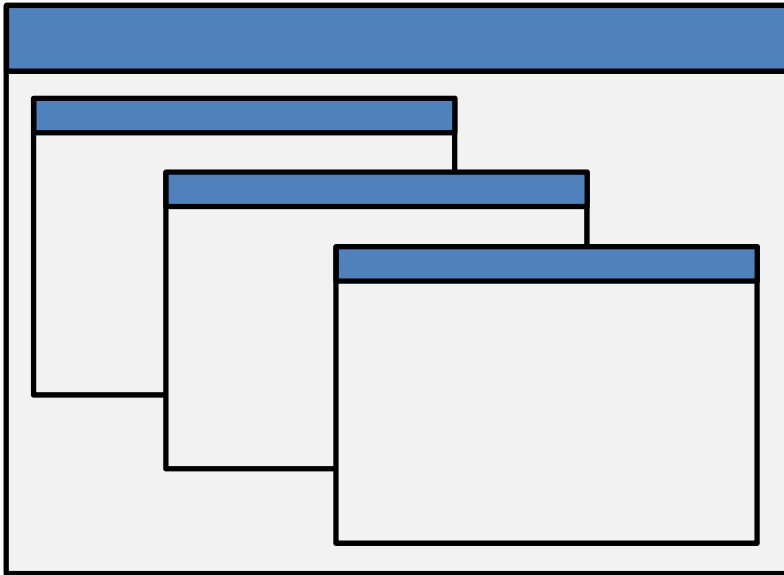
```
procedure TForm1.Button1Click(Sender: TObject);  
var  
    Value: Integer;  
begin  
    Value := 100;  
    Edit1.AsInteger := Value;  
end;  
  
procedure TForm1.Button2Click(Sender: TObject);  
var  
    Value: Integer;  
begin  
    Value := Edit1.AsInteger;  
    ShowMessage(Value.ToString);  
end;
```



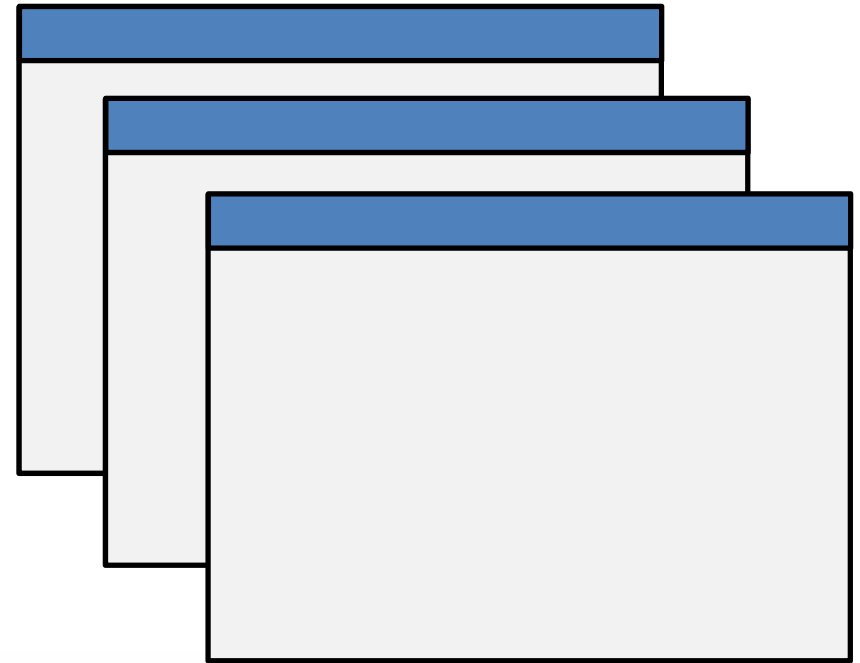
MDI と SDI

MDI は非推奨

- MDI = Multi Document Interface
- Microsoft は **MDI アプリケーション非推奨** だと言っています。



MDI

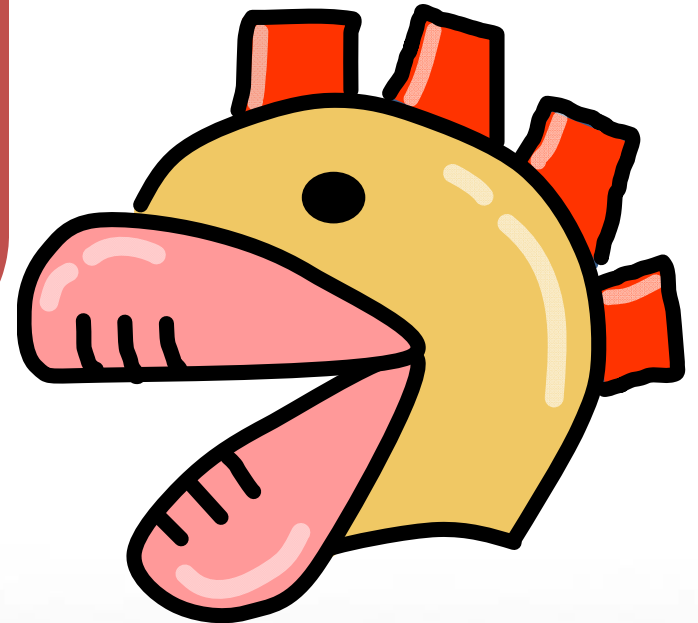


SDI

でも、Microsoft さん？

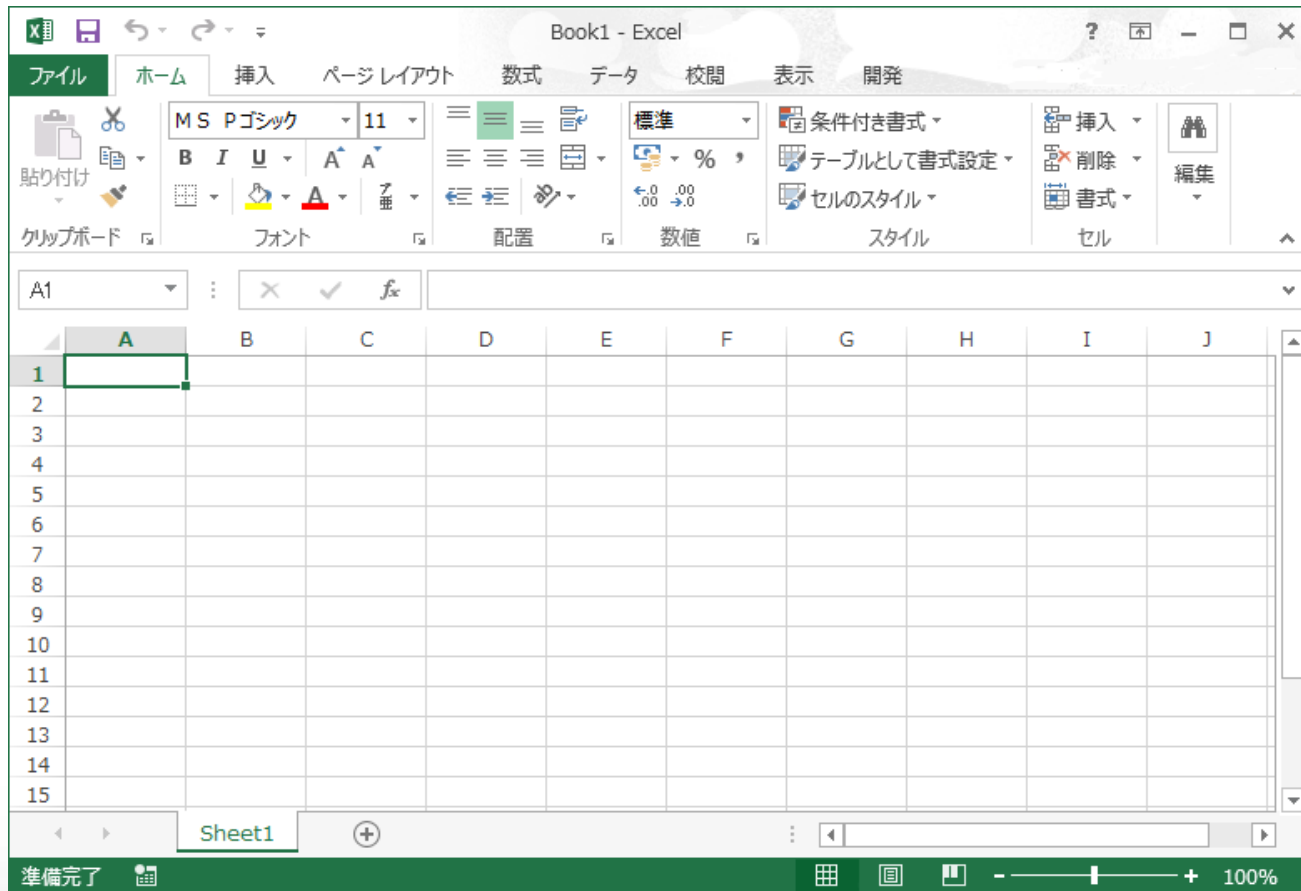


Excel は
MDI デスヨネ？



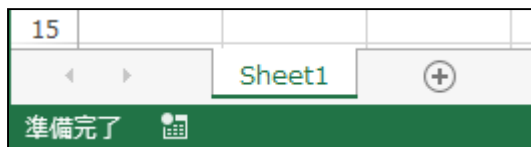
Excel 2010 まではね。

- Excel 2013 からは SDI アプリケーションになっています。



Excel 2013 の注意点

- ワークブックを削除するという事はアプリケーションを一つ終了させるという事。つまり、**次のフォーカスが Excel だとは限らない**。
- ワークブックを操作する時にはアクティブか否かを**明示的に指定**しなければならない。
- Excel 2013 起動時の**初期シートはひとつ** (従来は3つ)。最初に**シートが3つある前提で書かれたコード**はエラーを拝む。ちゃんとシート数をコードで確認する事。



Excel 2013 の互換性確認

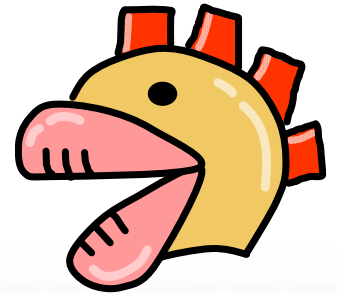
- Microsoft は開発者向けに Office 2013 の評価版を用意しているので、Excel 2013 を持っていないければこれで確認する。

<http://technet.microsoft.com/ja-jp/evalcenter/jj192782.aspx>

- Delphi からオートメーションで操作している場合には、ホワイトペーパー「Office 2013 マクロ互換性について」に目を通しておく事。

<http://technet.microsoft.com/ja-jp/office/jj149945.aspx>

ほー



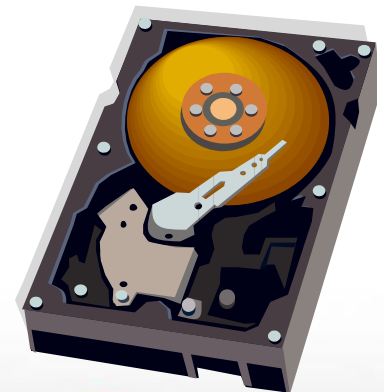


7

茶色の狐はノロマな犬を
何回飛び越えられるのか？

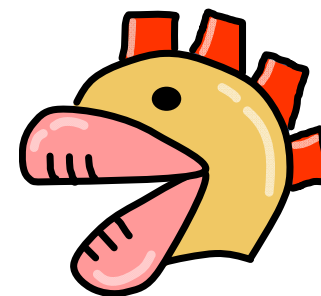
大容量ハードディスク

- HDD も大容量になりましたね。
- その分、ブッ壊れると復旧は大変デスヨネー。
- UEFI じゃないと大容量ハードディスクから起動できませんよねー。
- 古い PC で大容量 HDD を使う場合には 2.2TB 以下じゃないと起動ディスクにできませんよねー。データディスクとしては使えるけれど....。
- そんな話ではありません。



NAS やファイルサーバは...

- HDD の大容量化に伴い...
- どんどんドキュメントが蓄積されていきます。
- フォルダは増え、階層も深くなります。
- 名前がカブるのでファイル名も長くなります。



話が見えて
こないんだが...



何の話がしたいの？

The quick brown fox jumps over the lazy dog



(System).SysUtils.ForceDirectories()

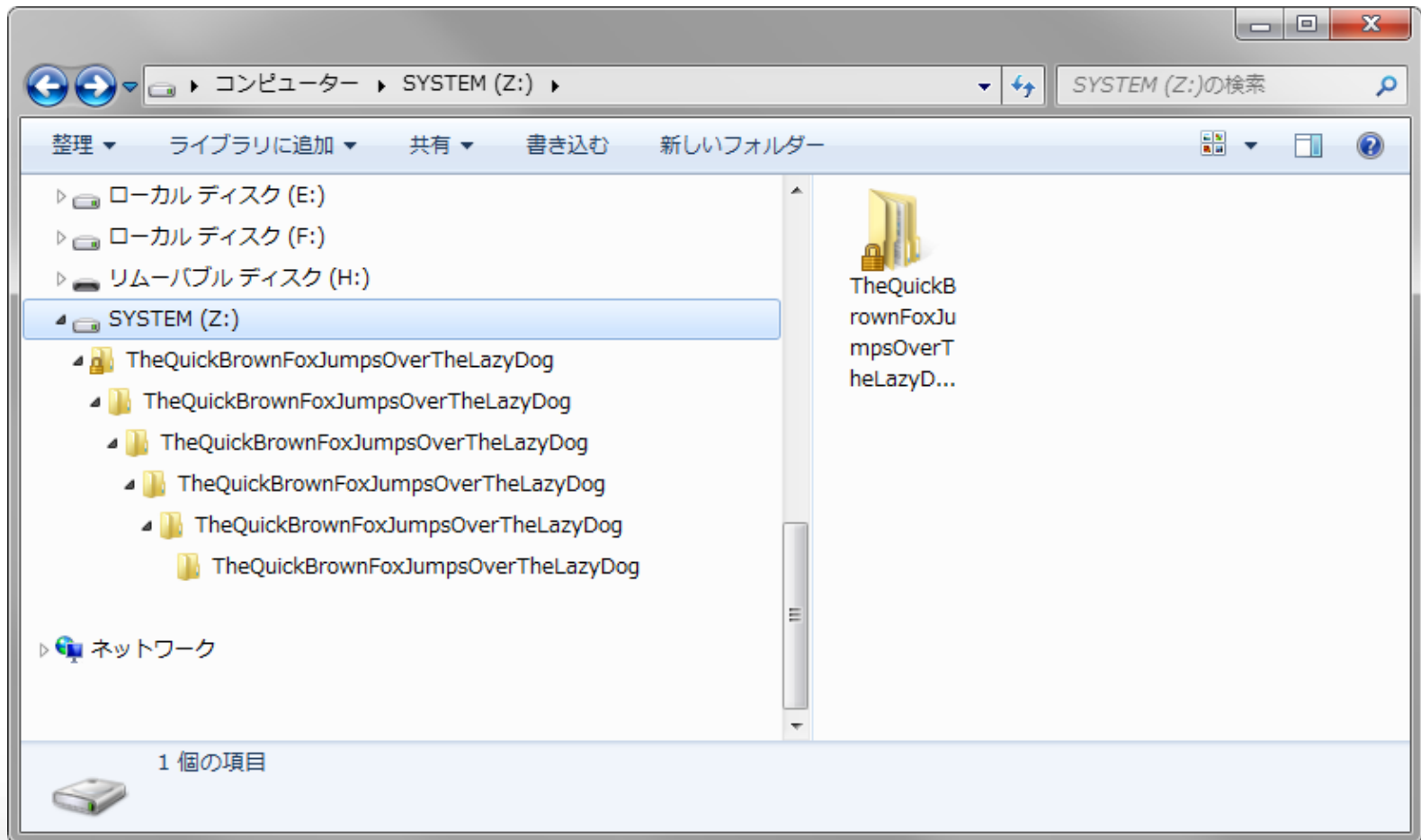
- 一気にサブフォルダまで掘ってくれる便利な関数。

```
procedure TForm1.Button1Click(Sender: TObject);
const
  DIR_STR = 'TheQuickBrownFoxJumpsOverTheLazyDog'; // 35 chars
var
  i: Integer;
  Dir: string;
begin
  Dir := 'Z: ';
  for i:=1 to 10 do
    Dir := SysUtils.IncludeTrailingPathDelimiter(Dir) + DIR_STR;
    SysUtils.ForceDirectories(Dir);
end;
```

- 上記コードで '10階層' 一気に掘れます。

あれ？

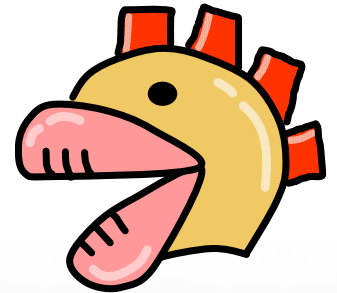
- 6回しか飛び越せんかった...orz



MAX_PATH

- SysUtils.ForceDirectories() は、そのままでは MAX_PATH の範囲でしかフォルダを掘れません。
- MAX_PATH = 260 です。
これはバイト数ではなく文字数 (UTF-16 換算) です。
- さっきの例だと 'TheQuickFox～' は 35文字ですから、10階層にしてドライブ文字列とパス区切り文字を含めると 362 文字のパスになってしまいます。
- 「**嫌な予感**がしてきましたか？」

…すごい。



拡張プレフィックスパス

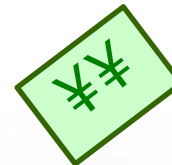
- 深い階層を表すには、
拡張プレフィックスをパスに付加すればいいのです。

- ローカルパスの場合

C:¥Foo → ¥¥?¥ C:¥Foo

- UNC パス (ネットワークパス) の場合

¥¥Server¥Share¥Bar
→ ¥¥?¥ UNC¥ Server¥Share¥Bar



びっくり
させんなよー



再度 (System).SysUtils.ForceDirectories()

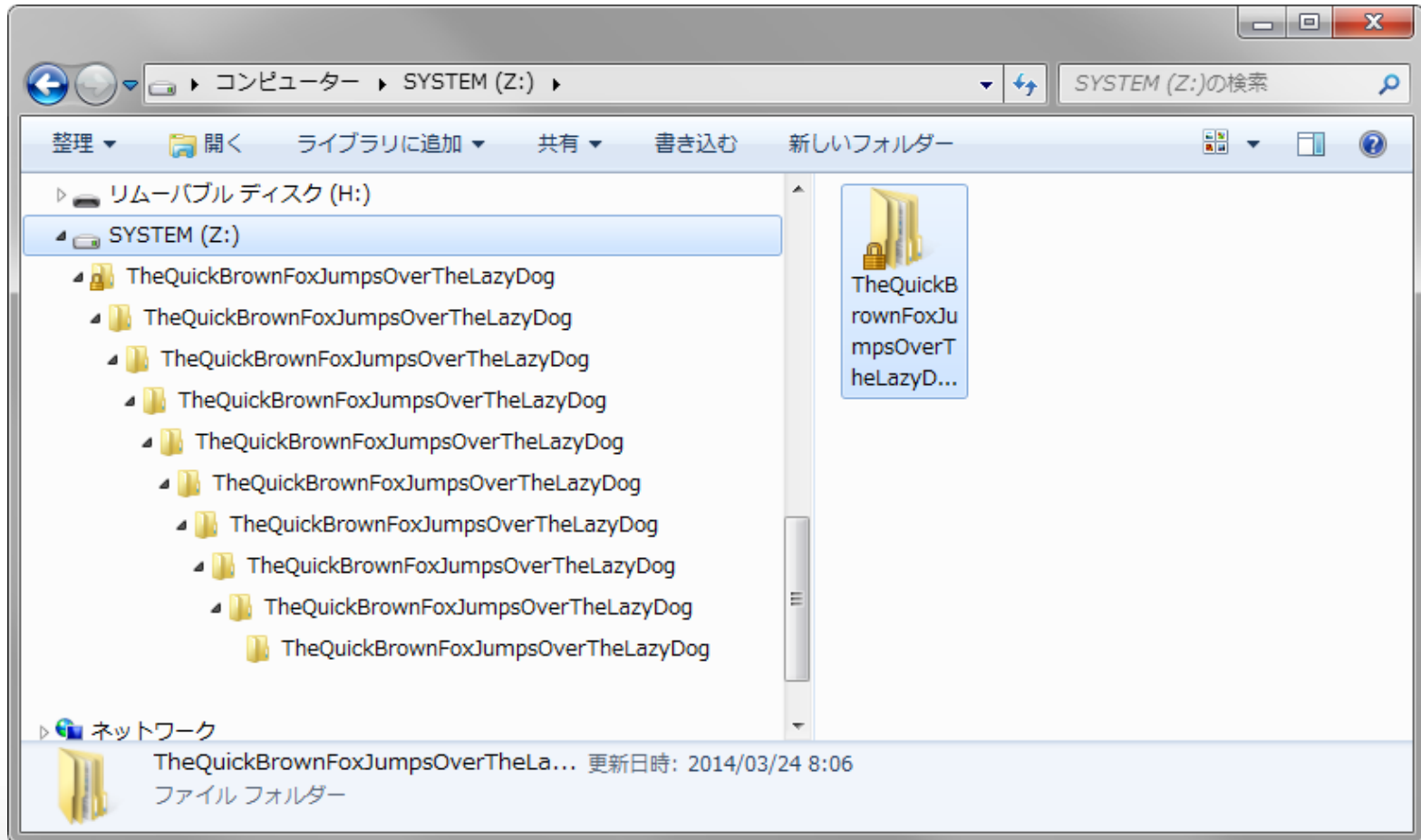
- 今度はどうでしょう？

```
procedure TForm1.Button1Click(Sender: TObject);
const
  DIR_STR = 'TheQuickBrownFoxJumpsOverTheLazyDog'; // 35 chars
var
  i: Integer;
  Dir: string;
begin
  Dir := '¥¥?¥Z: ';
  for i := 1 to 10 do
    Dir := IncludeTrailingPathDelimiter(Dir) + DIR_STR;
    System.SysUtils.ForceDirectories(Dir);
  end;
```

' ¥¥?¥Z: ' ;

今度は！

- 10回飛び越せた \ (^o^)/



ちなみに...

- A系 API は拡張プレフィクスパスに対応していない...ハズなのですが、**実際には通る**ものが多いです。
(全部は確認していませんけれど...)
- Delphi 2007 以前の ANSI 版 Delphi では内部で A系 API が使われています。
- 実際問題として、この件は ANSI 版 Delphi でも対応できない事はありませんが、できれば Unicode 版 Delphi を使った方がいいように思います。

A系 API と W系 API の MAX_PATH

- MSDN にはこう書いてあるのですけどね...

この関数の ANSI 版では、名前は最大 MAX_PATH 文字に制限されています。この制限をほぼ 32,000 ワイド文字へ拡張するには、この関数の Unicode 版を呼び出し、パスの前に "¥¥?¥" という接頭辞を追加してください。詳細については、MSDN ライブラリの「」（ファイル名の規則）を参照してください。

※「」は脱字ではありません。

MAX_PATH を Char 型のバッファで使う

- Unicode 版 Delphi で A 系 API を呼んだり、逆に ANSI 版 Delphi で W 系 API を呼ばないのようになります。

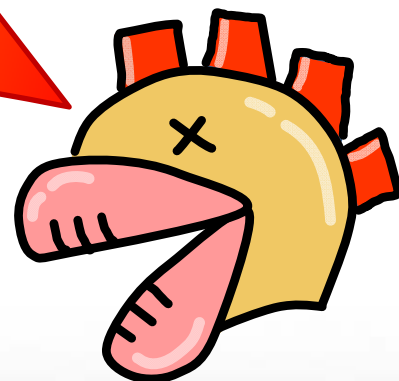
```
const
  {$IFDEF UNICODE}
    MAX_PATH_LEN = MAX_PATH;
  {$ELSE}
    MAX_PATH_LEN = MAX_PATH * 2;
  {$ENDIF}
```

```
var
  Buf: array [0..MAX_PATH_LEN + 1] of Char;
```

- Unicode の場合はそのまま。ANSI (Shift_JIS) の場合は最大でも 1文字 = 2Bytes なので、2 倍にすればいい。
(Shift_JIS は Unicode の BMP の範囲に収まるため)

安心するのは早くないですか？

SHFileOperation() は
拡張プレフィックスパスに
対応していない！！



SHFileOperation()

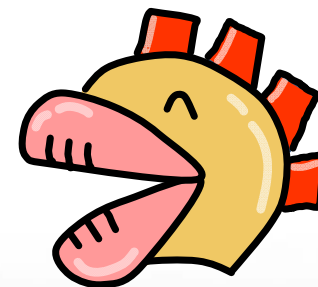
- 拡張プレフィクスを付けると動作しなくなる。
(戻り値: DE_INVALIDFILES = 0x7C)
- UNC 形式の拡張プレフィクスはエラーになる。
- もちろん MAX_PATH の制約を受ける。
- COM インターフェイス IFileOperation で置換するという手もあるが、コピー時にコピー先フォルダがあると失敗するという問題があり、かつ Vista 以降でしか使えない。
- SHFileOperation() のファイル操作は実は遅い。

マジですか…



TPath / TDirectory / TFile

- **Delphi 2010** から追加された TPath / TDirectory / TFile を使えば SHFileOperation() を代替できる。
- 名前空間は (System.)IOUtils
- TDirectory はディレクトリの操作、TFile はファイルの操作、TPath はフォルダ名やファイル名の操作を行うためのレコード型 (クラスではないので生成 / 破棄は不要)。
- 拡張プレフィクスに対応している。
- 名前空間で判るようにマルチプラットフォーム対応。



(System).IOUtils.TDirectory

- TDirectory.Copy() はディレクトリごとファイルをコピーする。
- TDirectory.Delete() はディレクトリごとファイルを削除する。
但し、サブフォルダが存在する場合には Recursive を True にしないと削除されない。
- TDirectory.Move() はディレクトリごとファイルを移動する。
但し、移動先にフォルダが存在してはならない。

(System).IOUtils.TFile

- TFile.Copy() は単一のファイルをコピーする。
「Delphi にはファイルコピー関数がない」というのは過去の話。
ファイルがコピー先に既にあっても上書きするようにするには
パラメータ Overwrite を True にする。
- TFile.Delete() は単一のファイルを削除する。
- TFile.Move() は単一のファイルを移動する。
ファイルがコピー先に既にある場合には移動できない。
- いずれもファイル属性を考慮する必要がある。

TDirectory + TFile

- ワイルドカードでのコピー / 削除 / 移動は TDirectory + TFile を組み合わせて使います。

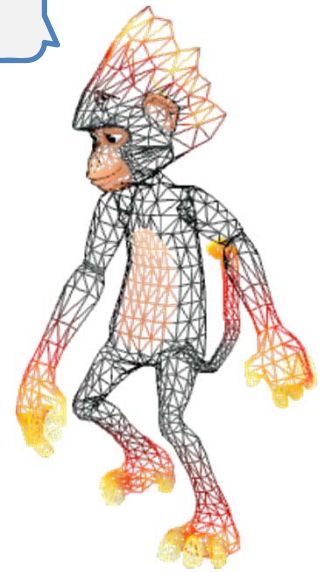
```
var
  FileName: string;
begin
  for FileName in TDirectory.GetFiles('Z:¥', '*.jpg',
                                       TSearchOption.soAllDirectories) do
    TFile.Delete(FileName);
end;
```

Z: ドライブにあるすべての階層の *.jpg ファイルを削除します。

(System).IOUtils.TPath

- パス名、フォルダ名、ファイル名を操作するのに使う。
- 旧来の ExtractFilePath() 等と同様のメソッドが
"あらかた" 用意されている。
- FireMonkey には名前空間違いの
 TPath (FMX.Objects.TPath) があるけれど
全く違うので勘違い...しないね、うん。
- 欲を言えばもうちょっとだけ機能が欲しい。

何か？

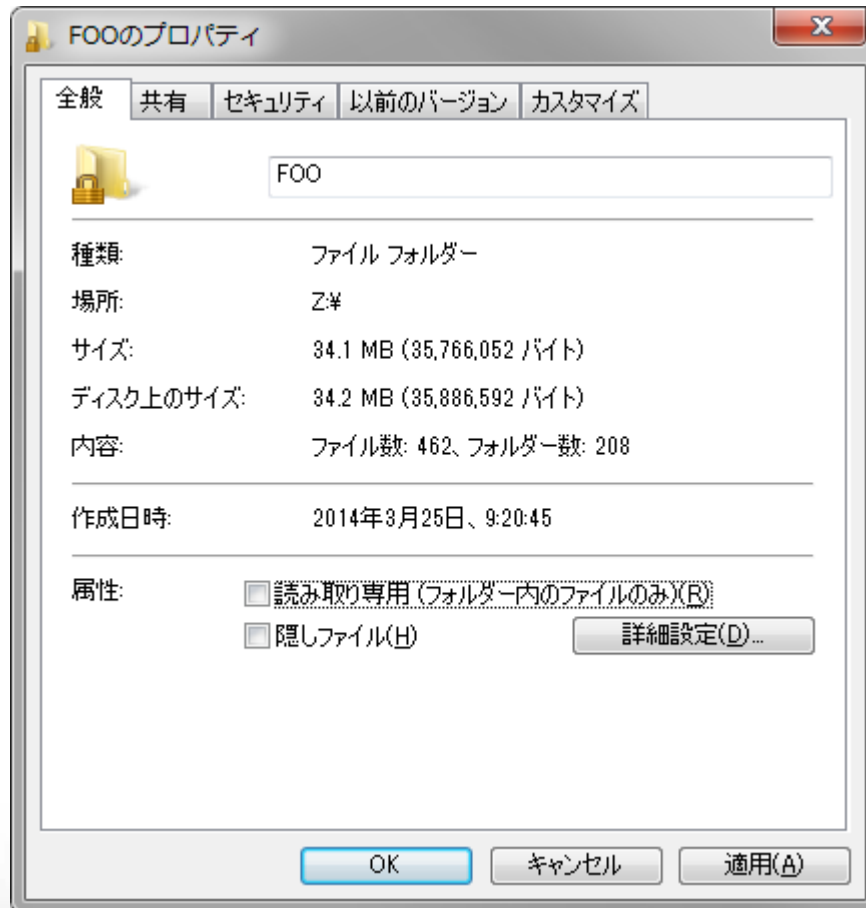


パス名あれこれ

- パス区切り文字は '~~¥~~' ではなく **PathDelim** を使うといいです。
- コードが冗長だと思えてもパス終端は `IncludeTrailingPathDelimiter()` と `ExcludeTrailingPathDelimiter()` でキッチリ処理しましょう。
- 上記 2 つを守ればマルチプラットフォーム対応の時に困った事になりません。
- `SHFileOperation()` は `C:¥foo¥¥bar` みたいな不正なパスも受け付けるのでバグの元です。パスの連結は `"+"` を使わずに **`TPath.Combine()`** を使うと不正なパスを作り出しにくくなります。

そーい、SHFileOperation() が遅いって...

- 疑り深いなあ...。



Z:¥Foo (HDD 上)

- 35MB 程度のフォルダ
- フォルダ数: 208
- ファイル数: 462

これを Z:¥Bar にコピー
するのを 10 回計測して
平均を算出。

結果:

SHFileOperation(): 3744ms

TDDirectory.Copy(): 2031ms

SHFileOperation() の実装

FileOperation()

```
function FileOperation(Mode: UINT32; const Src, Dst: string): Integer;
var
  FileOp: TSHFileOpStruct;
begin
  with FileOp do
    begin
      Wnd := 0;
      wFunc := Mode;
      fFlags := FOF_NOCONFIRMATION or FOF_SILENT;
      if (Mode <> FO_DELETE) then
        fFlags := fFlags or FOF_NOCONFIRMMKDIR;
      pFrom := PChar(Src + #$00#$00);
      pTo := PChar(Dst + #$00#$00);
      fAnyOperationsAborted := False;
      hNameMappings := nil;
    end;
    result := SHFileOperation(FileOp);
  end;
```

普通の実装だと思います。
ワザと遅くなるような
小細工はしてありません (w

※ SHFileOperation() を使うには (Winapi.)ShellAPI を uses に加える必要がある。

評価コード

SHFileOperation()

```
var
  Tick: Int64;
  Stopwatch: TStopWatch;
begin
  Stopwatch := TStopWatch.StartNew;
  Tick := Stopwatch.ElapsedMilliseconds;
  FileOperation(FO_COPY, 'Z:¥FOO', 'Z:¥BAR');
  ShowMessage(IntToStr(StopWatch.ElapsedMilliseconds - Tick));
end;
```

TDirectory.Copy()

```
var
  Tick: Int64;
  Stopwatch: TStopWatch;
begin
  Stopwatch := TStopWatch.StartNew;
  Tick := Stopwatch.ElapsedMilliseconds;
  TDirectory.Copy('Z:¥FOO', 'Z:¥BAR');
  ShowMessage(IntToStr(StopWatch.ElapsedMilliseconds - Tick));
end;
```

※ TStopWatch の名前空間は(System.)Diagnostics

SHFileOperation() を高速に！

- 長いパスはともかく、速度の件だけなら以下を試してください。
- NIC のプロパティで **QoS パケットスケジューラ** を外します。
- 同じく **インターネット プロトコル バージョン6 (TCP/IPv6)** も外します。
- コマンドプロンプトで以下を実行します (Windows 7 以降)。

```
netsh int tcp set global chimney=disabled  
netsh int tcp set global rss=disabled  
netsh int tcp set global netdma=disabled
```

- これで SHFileOperation() が速くなるはずです (Explorer も)。
- スペックの低い PC で特に有効な手段です。
- ...それでも、TFile / TDirectory の方が速いですけどね！

Delphi 2010 よりも前のバージョンでは？

- FindFirst() / FindNext() / FindClose() と再帰を使い、SysUtils にある従来のフォルダ / ファイルルーチンを組み合わせてどうにかしてください。
- SUBST や net use を使い、深いパスを仮想ドライブ / ネットワークドライブに割り当ててやるのも一つの手です。
- ...ただ、SUBST は Windows **起動時に再接続してくれない**のでスタートアップに仕込むなりする必要があります。
- 階層をまんべんなくアクセスしなくていいのなら、シンボリックリンクを使うとか (NTFS 限定)。

SUBST と net use

- **SUBST**

指定されたパスを**仮想ドライブ**として割り当てる。ネットワークパスを割り当てる事はできない。永続的にドライブを割り当てる機能はない。API で同等なのは DefineDosDevice()。

- **net use**

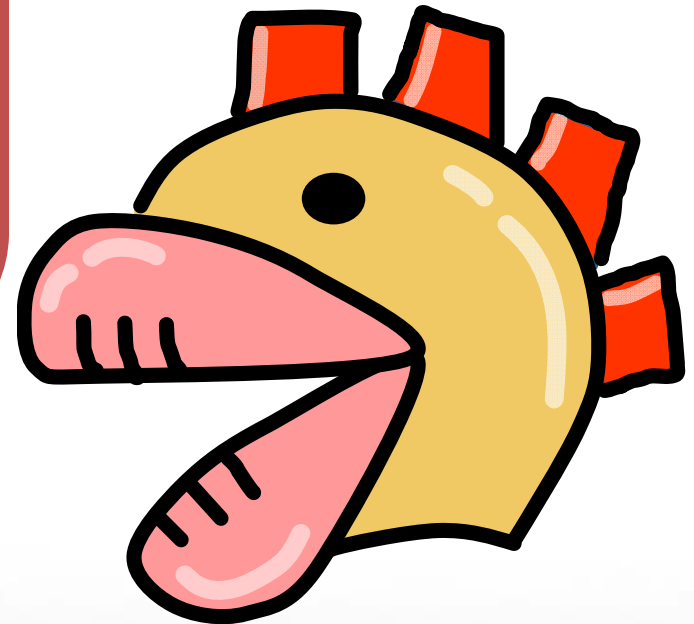
指定された UNC パスを**ネットワークドライブ**として割り当てる。ローカルパスを割り当てる事はできない。

API で同等なのは WNetAddConnection() / WNetAddConnection2() / WNetAddConnection3()。

- **ファイル / フォルダ操作のテストは仮想ドライブでやりましょう。**

念の為に確認してみる。

DB のフィールドに
FILENAME
VARCHAR(255)
とかないよネ？
(フルパス格納)



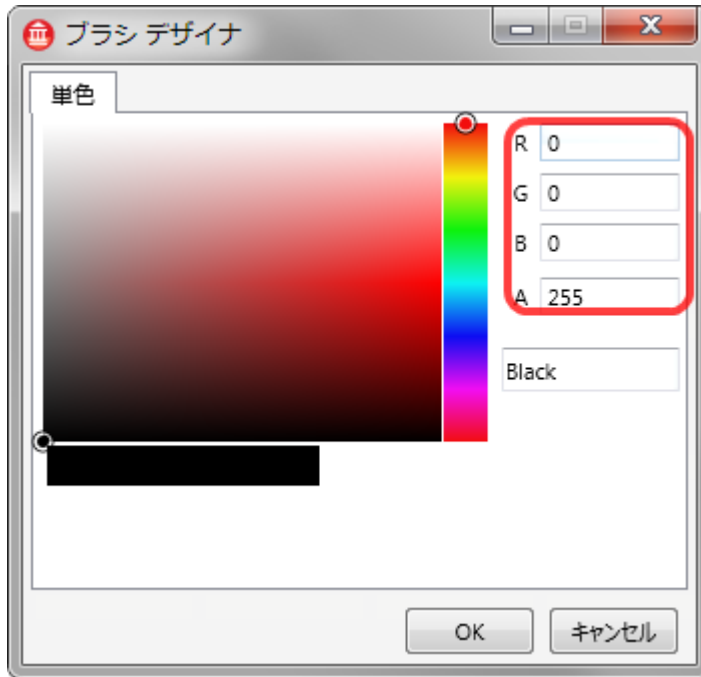
※メモ型あるいは BLOB 型にしましょう。



FireMonkey あれこれ

TNumberBox ですよ、TNumberBox

- TNumberBox ってコレです。



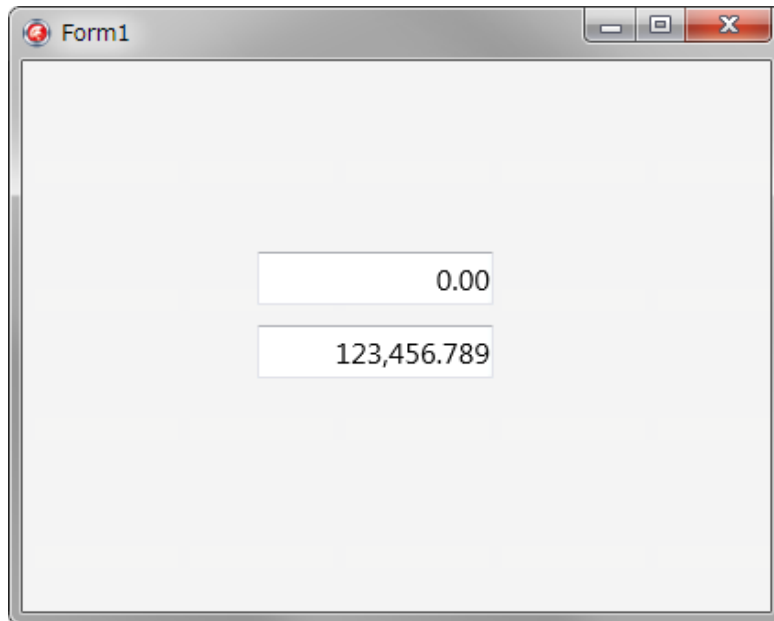
数値を入力できる…のだけど

- 格納できる値は **Single**
- DecimalDigits に小数点以下の桁を指定しても固定小数点にならない。
- デフォルトだと値をマウスで選択しようとする値がインクリメント/デクリメントされてしまう。
- デフォルトで左揃え
- 桁区切り文字は表示されない。

ビジネス用途には微妙に使えない！！

ないのなら...

- 作ってしまえ、TNumEdit



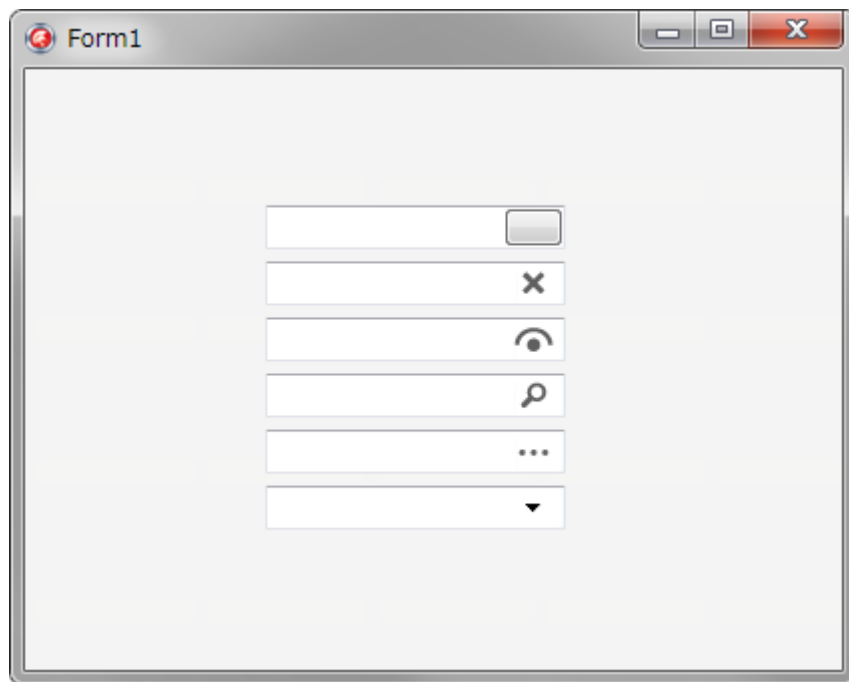
数値を入力できる！

- 格納できる値は **Extended**
- DecimalDigits に
小数点以下の桁を指定可能。
- デフォルトで右揃え
- 桁区切り文字は Thousands
プロパティで ON/OFF 可能。

ビジネス用途に使えるかも！

TEdit は...

- **ダブルクリック**すると、ボタンが付けられます。



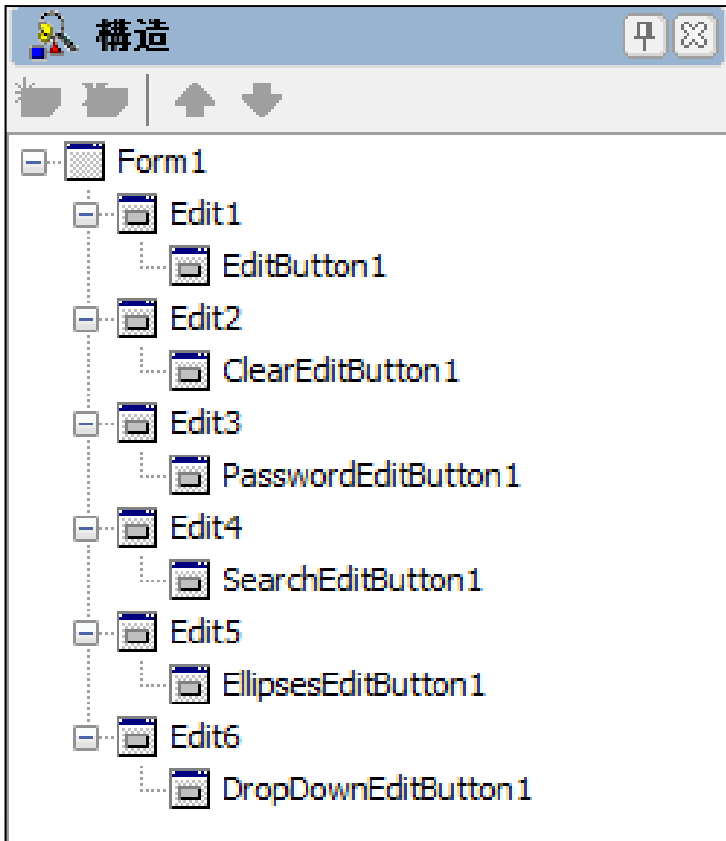
上から順に、

- TEditButton
- TClearEditButton
TClearingEdit 使ったほうが早い。
- TPasswordEditButton
- TSearchEditButton
- TEllipsesEditButton
- TDropDownEditButton
TComboEdit 使ったほうが早い。

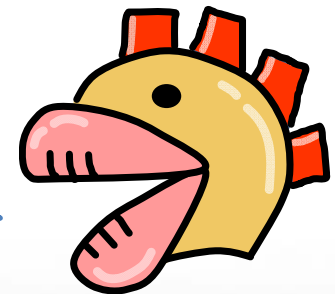
- FMX の TEdit にはコンポーネントエディタがあります。

TEditButton の派生ボタンは…

- [構造] ペインで確認するとこんな感じ。

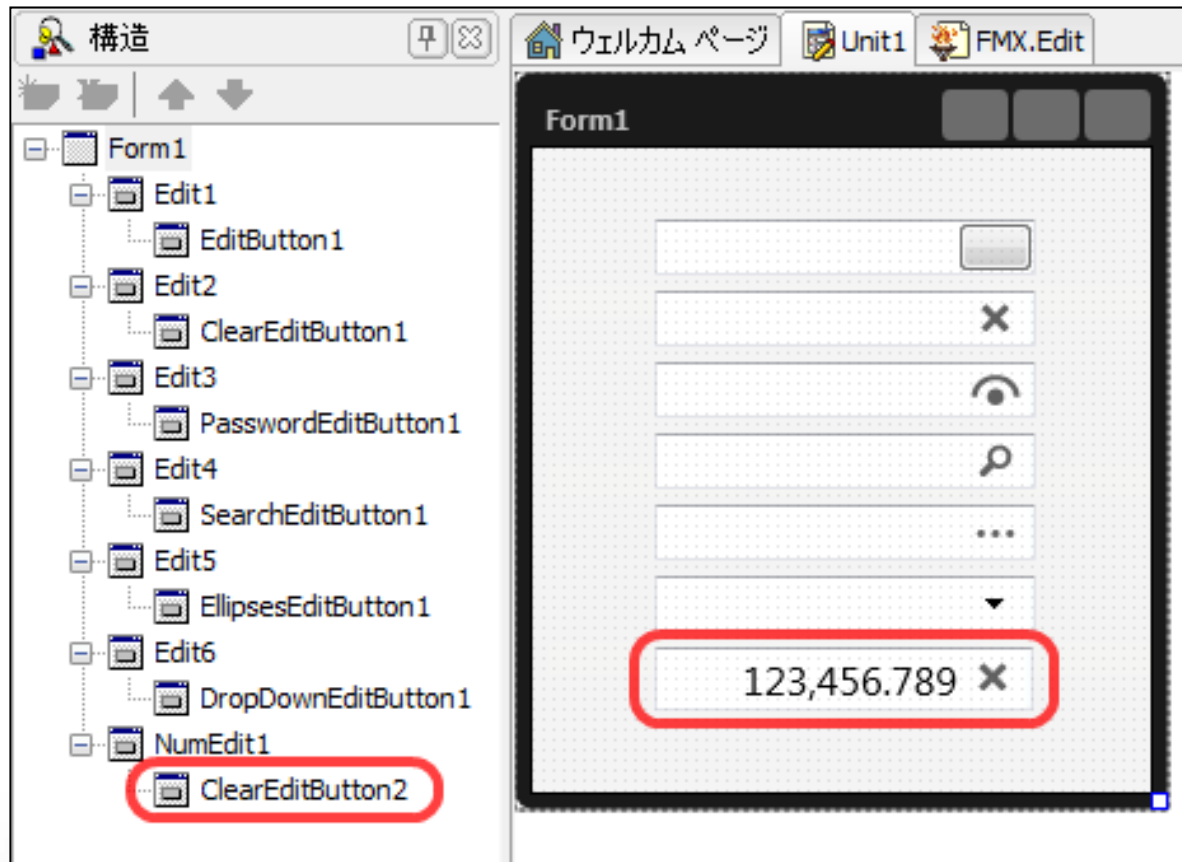


ん？
それってつまり…



TCustomEdit の派生コンポーネントに...

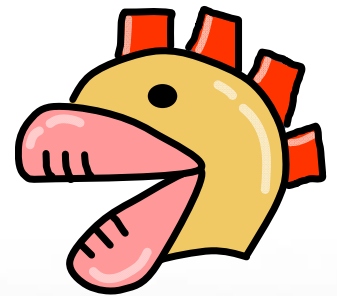
- ぶら下げる事ができます (^o^)/



ダブルクリックしてコンポーネントを追加できるコンポーネント

- TEdit
- TGrid / TStringGrid
- THeader
- TListBox / TComboBox
右クリック→[項目の追加] だと、
TListBoxHeader と TSearchBox が別途追加可能。
- TMainMenu / TPopupMenu / TMenuBar
- TTabControl
- TTreeView

とりあえず
ダブルクリックしてみ？

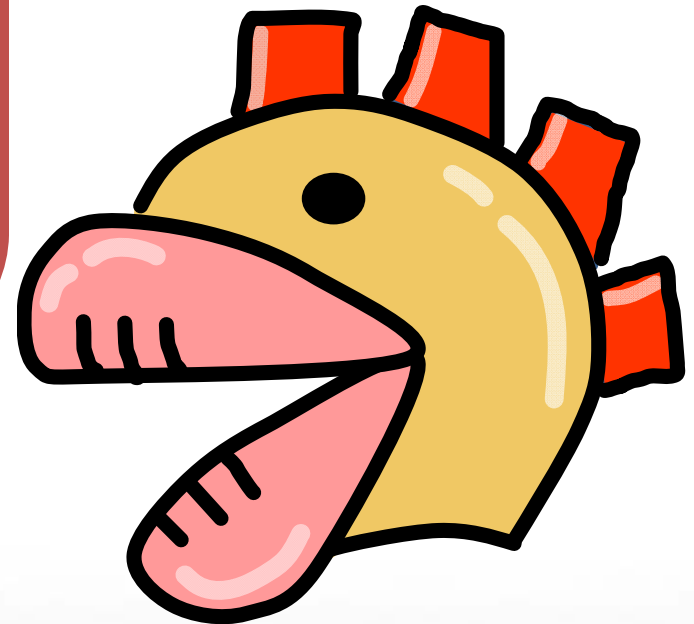




言い忘れていた VCL のアレ

言い忘れてたけど...

Vista 以降だと
ProgressBar を
使うだけで
動作が遅くなる
事があるヨ！



※検索が遅いんじゃなくて、ProgressBar の
描画が遅いだけだったりするヨ。

uProgressBarFix.pas

```
unit uProgressBarFix;

interface

uses
  Classes, Windows, StdCtrls, ComCtrls, CommCtrl;

type
  TProgressBar = class(ComCtrls.TProgressBar)
  private
    FPosition: Integer;
    procedure SetPosition(const Value: Integer);
    function GetPosition: Integer;
  public
    constructor Create(AOwner: TComponent); override;
    property Position: Integer read GetPosition write SetPosition default 0;
  end;

implementation

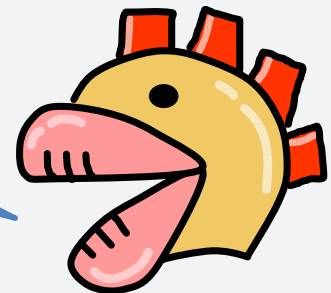
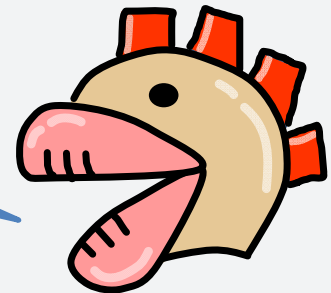
constructor TProgressBar.Create(AOwner: TComponent);
begin
  inherited;
  FPosition := 0;
end;

function TProgressBar.GetPosition: Integer;
begin
  if HandleAllocated then
    Result := SendMessage(Self.Handle, PBM_GETPOS, 0, 0)
  else
    Result := FPosition;
end;

procedure TProgressBar.SetPosition(const Value: Integer);
begin
  if FPosition <> Value then
  begin
    FPosition := Value;
    if HandleAllocated then
      SendMessage(Self.Handle, PBM_SETPOS, Value, 0);
  end;
end;
end.
```

実測して
遅かったら
試してみてネ

uProgressBarFix
を uses するだけ
で OK ダヨ。

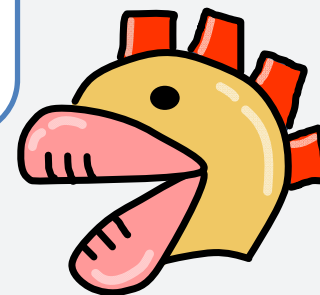


描画を正常にした上で高速化するには？

- SetPosition() を書き換えてください。

```
procedure TProgressBar.SetPosition(const Value: Integer);
var
  IncMax: Boolean;
begin
  if FPosition = Value then
    Exit;
  if HandleAllocated then
    begin
      if FPosition < Value then
        begin
          IncMax := (Value = Self.Max);
          if IncMax then
            Self.Max := Self.Max + 1;
          SendMessage(Self.Handle, PBM_SETPOS, Value + 1, 0);
          SendMessage(Self.Handle, PBM_SETPOS, Value, 0);
          if IncMax then
            Self.Max := Self.Max - 1;
        end
      else
        SendMessage(Self.Handle, PBM_SETPOS, Value, 0);
      end;
    FPosition := Value;
  end;
```

ProgressBar の
速度はコードの
書き方によっても
左右されるから
一概には語れない
のヨネー



描画速度は逆に落ちるかもしれないけれど、ゲージは正しく描画されるコードです。

そもそも...

- ProgressBar.Max は 100 のままにして Position を百分率で指定すれば **最大でも 100 回しか描画されない**ので高速に動作する...ハズ？

```
var
  i, RecCnt: Integer;
begin
  RecCnt := 12345;
  for i := 1 to RecCnt do
    begin
      ...
      // (処理)
      ...
      ProgressBar1.Position := i * 100 div RecCnt; // 整数演算で%を出す
      Application.ProcessMessages;
    end;
  end;
```

実は uProgressBarFix を
uses していないと、
遅くなるコードの代表例です。

ProgressBar の描画にはコストが掛かる事をお忘れなく！



まとめ？

脱線しまくりでしたネ

- XP のメンテが必要なら、仮想化を視野に入れておきましょう。
- 古い Delphi アプリケーションを最近の OS で動かすのなら、最低限の修正は行いましょう (マニフェストなど)。
- 単純作業が苦痛なら、最初に面倒な作業をやっときましょう。
(「急がばまわれ」です)
- Excel 2013 には気をつけましょう。
- パスの扱いには注意しましょう。
- SHFileOperation() はなるべく使わないようにしましょう。
- DB にパスを格納する時も注意しましょう。



Extras

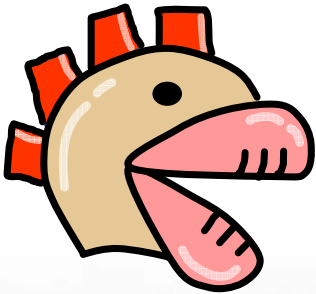
Microsoft Hyper-V Server

- Hyper-V Server は無償で利用可能です。
ゲスト OS は **XP 以降**をサポートしています。
- Microsoft Hyper-V Server 2008 R2
<http://www.microsoft.com/ja-jp/download/details.aspx?id=3512>
※ 比較的古いハードウェアを用いる場合。
※ 2012 でサポートされなくなったゲスト OS のドライバが必要な場合に。
- Microsoft Hyper-V Server 2012
<http://www.microsoft.com/ja-jp/server-cloud/hyper-v-server/>
※ 比較的古いハードウェアを用いる場合。
- Microsoft Hyper-V Server 2012 R2
<http://technet.microsoft.com/ja-jp/evalcenter/dn205299.aspx>
※ 比較的新しいハードウェアを用いる場合。

セッション中に出てきたイロイロなもの

- 評価コード等は後日まとめて閲覧できるようにしておきます。

<http://ht-deko.minim.ne.jp>



今日のうちにやれよ！

ひい～！！

