

【セッションNo. 3】

Delphi/400技術セッション 開発者が知りたい 実践プログラミングテクニック

株式会社ミガロ.
RAD事業部 技術支援課
吉原 泰介

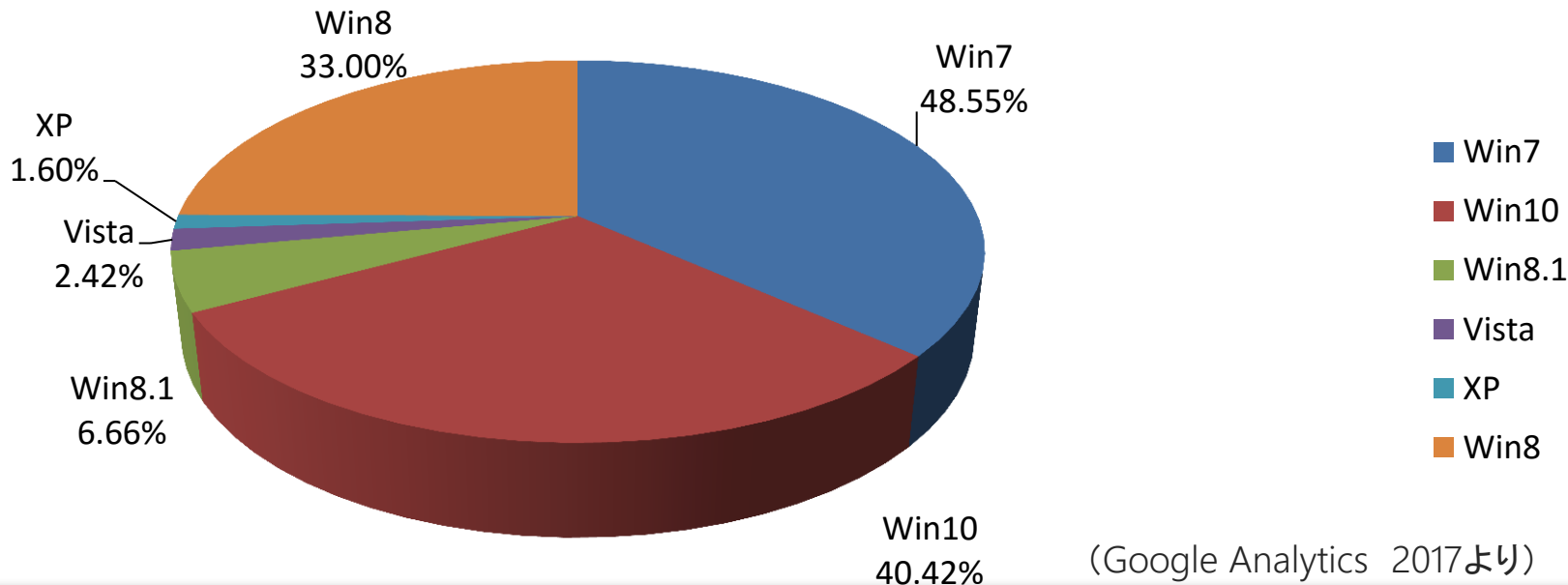
はじめに

企業で使用しているWindowsOS

2015年にWindows10がリリースされ、昨年より企業でのWindows環境入替が急速に進んできている。

現在はMicrosoftがサポートするWindows7、8(8.1)、10が企業で使用するWindows環境の主流になってきた。

【企業でのWindowsOS別使用率】



Delphi/400のWindows OS対応

Delphi/400は、10SeattleでWindows10にも正式対応！
ただし正式対応したアプリケーションでも、実行するWindows環境の違いで影響を受けてしまう場合もある。
本セッションでは、そうしたWindows環境の入替時などに役立つDelphi/400の技術ノウハウをご紹介します。

【Delphi/400のWindows OS対応表】

Delphi/400	Windows					
Version	2000	XP	Vista	7	8(8.1)	10
V5	○	—	—	—	—	—
V6	○	○	—	—	—	—
V7	○	○	—	—	—	—
V2005	○	○	—	—	—	—
V2006	○	○	—	—	—	—
V2007	○	○	○	—	—	—
V2009	○	○	○	—	—	—
V 2010	○	○	○	○	—	—
XE	—	○	○	○	—	—
XE3	—	—	○	○	○	—
XE5	—	—	○	○	○	—
XE7	—	—	—	○	○	—
10 Seattle	—	—	—	○	○	○

【アジェンダ】

Windows環境に関連したDelphi/400技術Tips

- 1.Windowsテーマのアプリケーションへの影響
- 2.解像度設定のアプリケーションへの影響
- 3.管理者権限のアプリケーションへの影響
- 4.IMEのアプリケーションへの影響
- 5.Windows64bitOSの考慮点
- 6.Windowsに関連するDelphi/400情報
- 7.まとめ

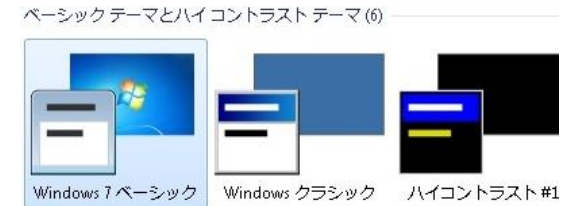
1.Windowsテーマのアプリケーションへの影響

対象OS:Windows Vista、7、8、10

1.Windowsテーマのアプリケーションへの影響

Windowsテーマとは？

Windowsのテーマとは、背景画像・ウィンドウの色・スクリーンセーバーの設定等をまとめたもの。
デスクトップを右クリックして
「個人設定」から指定できる。



特にWindows7まではクラシック（Windows2000風）の
テーマが用意されており、これを適用すると新しいWindowsの
表示が無効化されてかなり表示が変わってしまう。

通常のテーマ (Windowsメニュー)



クラシックテーマ (Windowsメニュー)



1.Windowsテーマのアプリケーションへの影響

アプリケーションへの影響

例えばGrid系の部品は新しいWindowsOSのテーマでは、グラデーションなど表示や仕様が大きく変わっているため、コンポーネントをカスタマイズしていると、ロジックによっては表示がうまくいかない場合がある。

例) CheckBOXをカスタマイズしたDBGrid



チェック	番号
☐	1001
☒	1002
☐	1003
☐	1004
☐	1005

選択時にセル値が意図せず描画されてしまう(余計な表示が出る)

1.Windowsテーマのアプリケーションへの影響

アプリケーションの対処方法①

GridのようにDrawingStyleプロパティを持っているコンポーネントでは、「gdsClassic」を設定すると部品単位でクラシック表示に切り替えることが可能。

(プログラムを直さずに表示の不備を対応する場合に便利)

	チェック	番号
	☐	1001
▶	☒ 1	1002
	☐	1003
	☐	1004
	☐	1005

クラシック形式

	チェック	番号
	☐	1001
▶	☒	1002
	☐	1003
	☐	1004

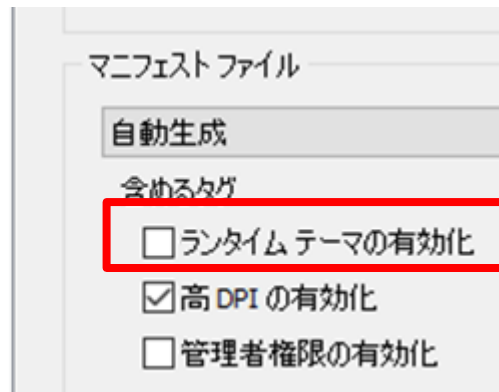
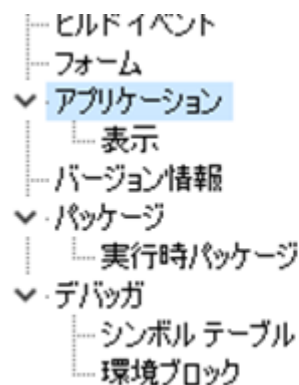
DSDBGrid1.DrawingStyle := gdsclassic;
プロパティで設定也可

1.Windowsテーマのアプリケーションへの影響

アプリケーションの対処方法②

アプリケーション単位でクラシックな見た目で表示したい場合には[プロジェクト|オプション|アプリケーション]設定で「ランタイムテーマの有効化」のチェックを外すことでアプリケーションがテーマの影響を受けなくなる。

[プロジェクト|オプション|アプリケーション]設定



※Delphiのバージョンによって設定項目名が若干異なる。

1.Windowsテーマのアプリケーションへの影響

アプリケーションの対処方法③

新WindowsテーマのGridでCheckBoxの表示を工夫する例

```
procedure TForm1.DBGrid1DrawColumnCell(Sender: TObject; const Rect: TRect;
  DataCol: Integer; Column: TColumn; State: TGridDrawState);
const
  CtrlState: Array[Boolean] of Integer =
    (DFCS_BUTTONCHECK, DFCS_BUTTONCHECK or DFCS_CHECKED);
var
  MyRect : TRect;
begin
  if Column.Field.DataType = ftInteger then //例
  begin
    Column.Alignment      := taCenter;
    TDBGrid(Sender).Canvas.FillRect(Rect);
    MyRect := Rect;
    DrawFrameControl(TDBGrid(Sender).Canvas.Handle, MyRect,
      DFC_BUTTON, CtrlState[Column.Field.AsFloat = 1]);
  end;
end;
```

チェック	番号
☐	1001
☒	1002
☐	1003
☐	1004
☐	1005

文字の位置を中央にして
描画をチェックボックスと重ねる

1.Windowsテーマのアプリケーションへの影響

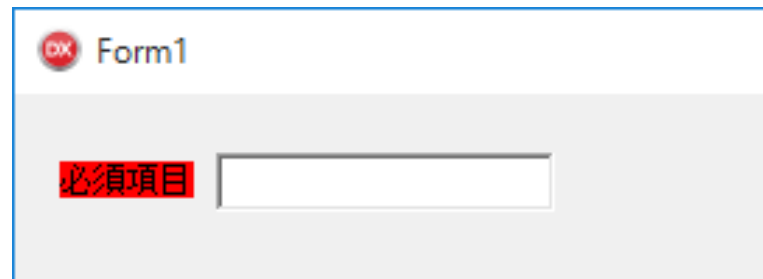
WindowsテーマのTLabelへの影響

TLabelのColorプロパティの指定はWindowsのテーマによって反映されない場合がある。

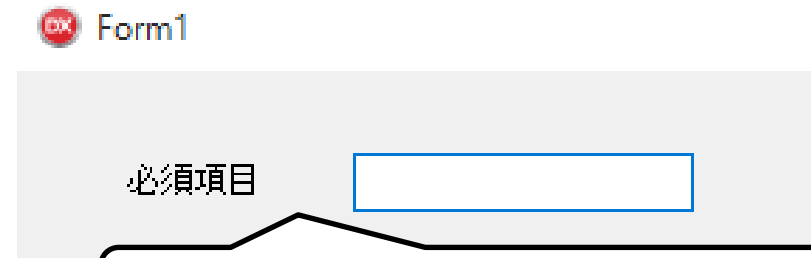
TLabelの背景色を赤に設定



クラシックテーマ環境の表示



通常のWindowsテーマ環境の表示



TLabelのColorが反映されない

1.Windowsテーマのアプリケーションへの影響

アプリケーションの対処方法

TLabelにはTransparent プロパティが用意されており、Falseにすることでテーマに影響されずにColorを確実に反映できる。

TransparentをFalseにすれば
TLabelのColorは必ず反映される

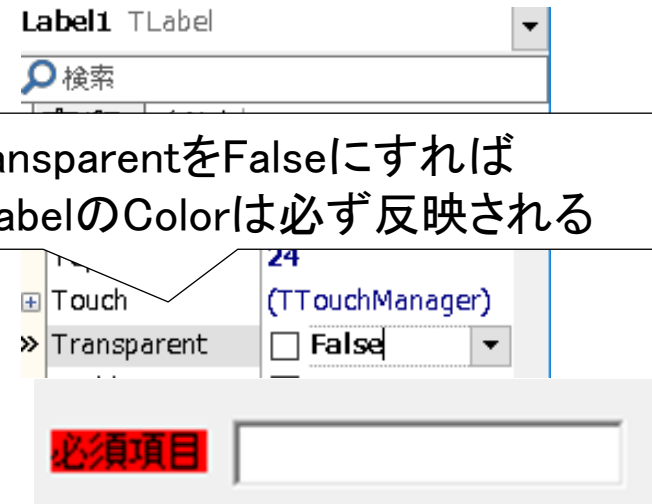
注意点

Delphi 2006 以前 : Transparentのデフォルト値はFalse

Delphi 2007 以降 : Transparentのデフォルト値はTrue

dfmにプロパティが保存されていない場合、バージョンアップ時にプロパティ値が変わってしまう危険性があるので気を付ける。

dfmの値がないと開発環境のWindowsテーマで影響を受けることもある。



2.解像度設定のアプリケーションへの影響

対象OS:Windows10

2.Windows解像度設定のアプリケーションへの影響

解像度とは？

解像度とは、画面に表示されるテキストと画像の鮮明さ（密度）を表したものの。解像度が高い場合、アイコンなどが小さく表示されますが、画面を広く使用することができます。また、表示が鮮明になる。

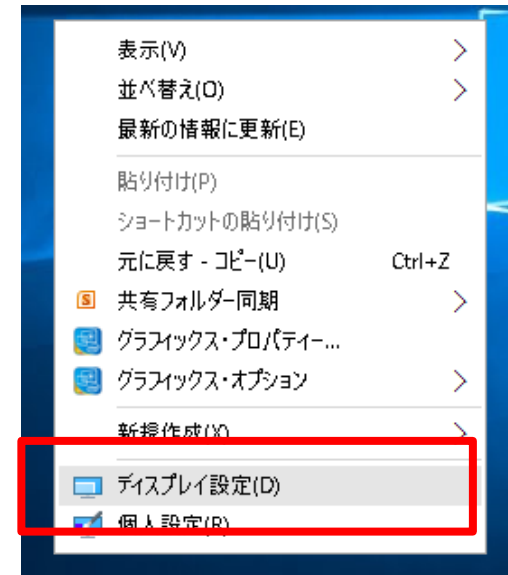


WindowsではVista以降に高解像度ディスプレイに対応するように大きく変更されており、WindowsXP以前のOSからWindows10に変更すると同じアプリケーションでも画面表示が変わってしまう場合がある。

2.Windows解像度設定のアプリケーションへの影響

Windowsの解像度設定

Windowsの解像度設定はデスクトップを右クリックして[ディスプレイ設定]から設定する。



解像度設定の中でも

「テキスト、アプリ、その他の項目サイズの変更する」はアプリケーションの表示に大きく影響する。

テキスト、アプリ、その他の項目のサイズを変更する: 125%



※Windows10では環境によって自動設定されてしまう。

2.Windows解像度設定のアプリケーションへの影響



例えば、同じアプリケーションでも
「テキスト、アプリ、その他の項目サイズの変更する」の
設定が変わると、下画面のような表示に違いが出る。

例)「テキスト、アプリ、その他の項目サイズの変更する」の違い
100% 150%



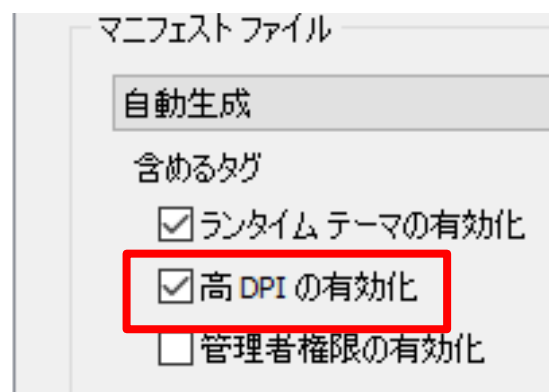
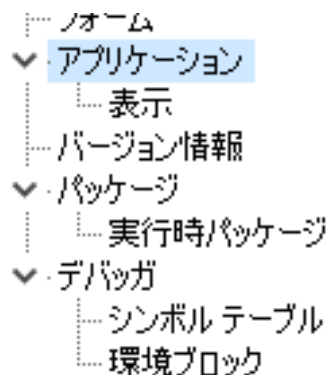
大きくて見やすくなるが、画面を占有し
デザインも狂ってしまう

2.Windows解像度設定のアプリケーションへの影響

アプリケーションの対処方法

アプリケーションレベルで高解像度に制御は
[プロジェクト|オプション|アプリケーション]で
「高DPIの有効化」のオン/オフで設定可能。
オンになっていると、先述した解像度関連の設定に対応して
しまうため、オフにすれば設定の影響を無視できる。

[プロジェクト|オプション|アプリケーション]設定



2.Windows解像度設定のアプリケーションへの影響

「高DPIの有効化」補足

「高DPIの有効化」の本来の機能目的としてはオンにしておく
くと、解像度は最適できれいな表示に対応できる。
(目的に応じて使い分ける必要がある)

「高DPI有効化」オフ

Delphi テクニカルセッション

解像度に対応できず粗い表示

「高DPI有効化」オン

Delphi テクニカルセッション

解像度に最適になり、きれいな表示

3. 管理者権限のアプリケーションへの影響

対象OS: Windows Vista、7、8、10

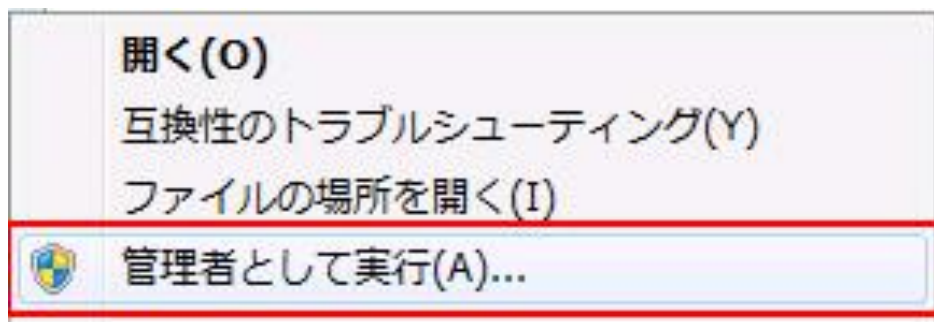
3.管理者権限のアプリケーションへの影響

管理者権限とは？

Windowsのシステム操作などに制約を受けない権限。

Windows Vista以降では、UAC（ユーザーアカウントコントロール）などの大きな変更もあり、Windows XP以前と違って管理者アカウントでログインしている場合でも、通常の操作は標準アカウントと同じ権限で実行される。

その為、Windows Vista以降で管理者実行する場合は、対象のプログラムを右クリックから「管理者として実行」で行う。



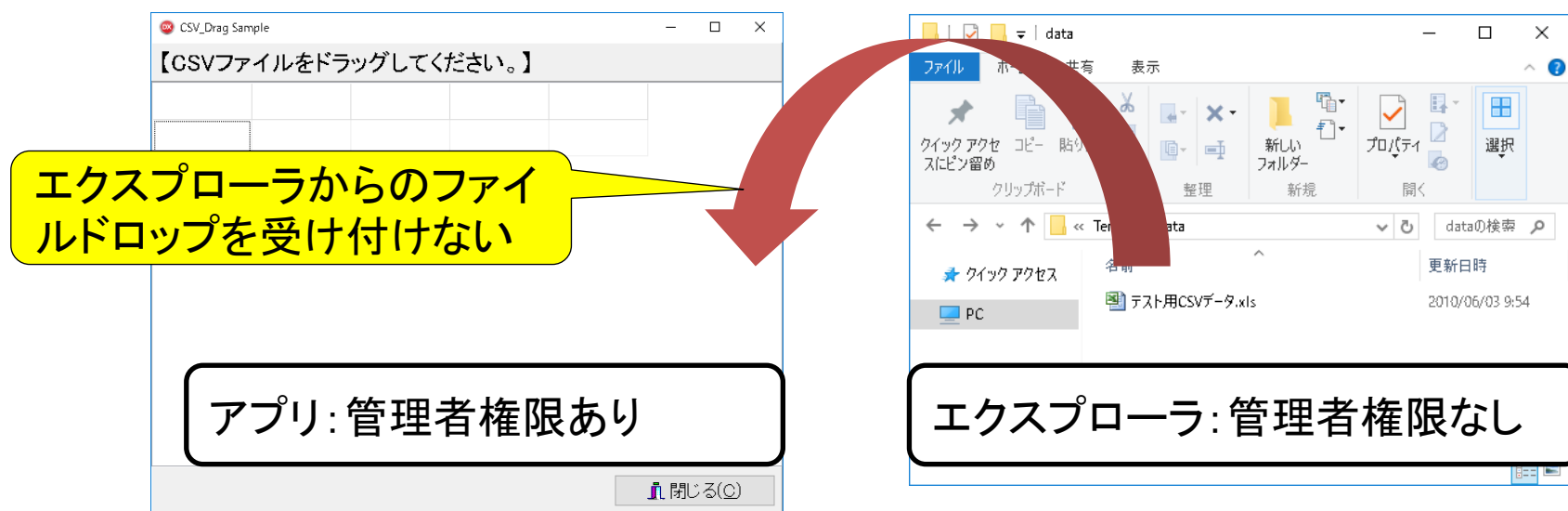
3.管理者権限のアプリケーションへの影響

管理者権限で影響を受けるアプリケーション

システムフォルダやファイルを扱うアプリケーションは管理者権限が必要になるが、把握はしやすい。注意が必要なのは、意図せず管理者権限が影響するアプリケーション。

例) ドラッグアンドドロップ機能

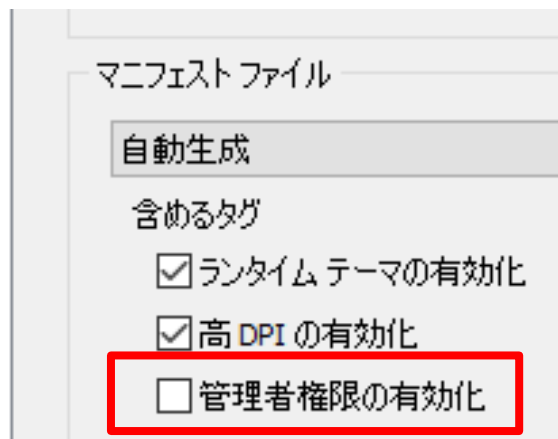
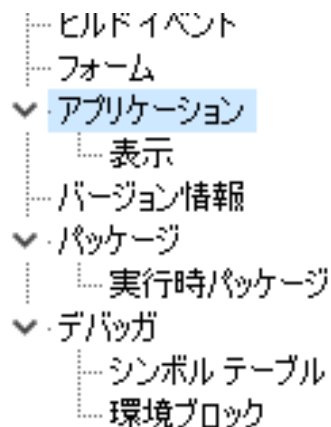
管理者権限でアプリケーションを起動している場合、エクスプローラ（非管理者権限）からのドラッグ&ドロップは受け付けない。



3.管理者権限のアプリケーションへの影響

アプリケーションの対処方法

管理者権限をアプリケーションで制御するには
[プロジェクト|オプション|アプリケーション]の
「管理者権限の有効化」チェックで設定可能。



デフォルトはオフ

補足: XE7以前のバージョンの場合では、この設定がないため、必要であればマニフェストを用意して有効にする。マニフェストファイルの生成については「第11回 Windows7に最適化したアプリ開発・運用テクニック」を参考。

4.IMEのアプリケーションへの影響

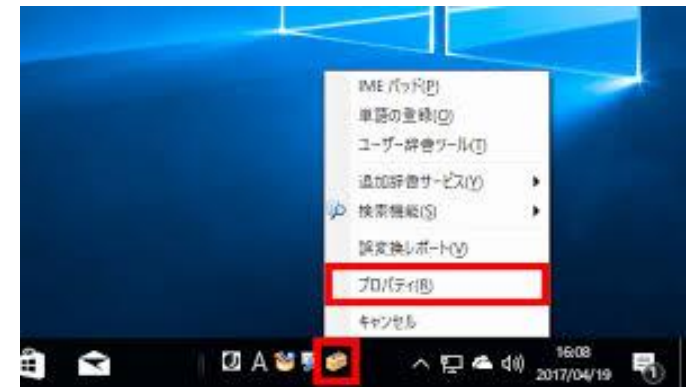
対象OS: Windows7以前のOS

4.IMEのアプリケーションへの影響

IMEとは

IMEとは文字入力をサポートするソフトウェア。
日本語入力用のIMEでは、ローマ字/ひらがな/カタカナ/漢字
などの変換機能を提供する。

Windows上のアプリケーションの入力でも通常IMEが使用されており、Delphi/400のコンポーネントではImeModeプロパティで制御を行う。



ImeModeプロパティ

ImeMode	imDontCare
ImeName	imAlpha
Left	imChinese
LiveBinding デザ	imClose
LiveBindings	imDisable
Margins	imDontCare
MaxLength	imHangul
	imHira

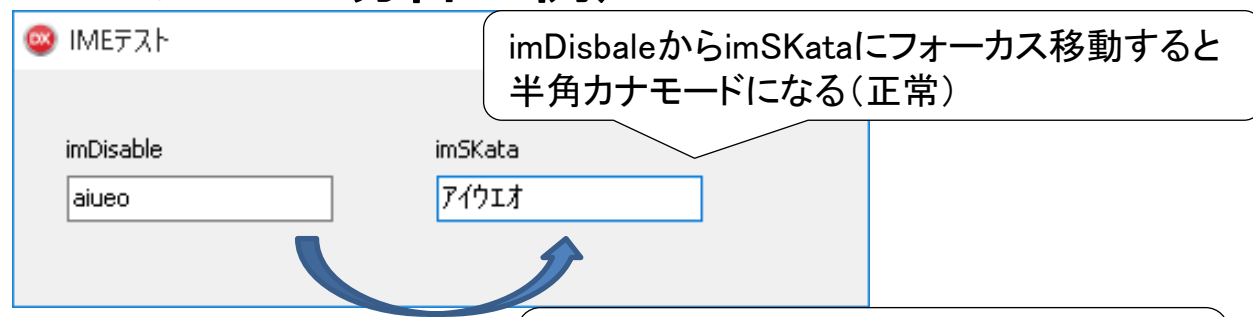
4.IMEのアプリケーションへの影響

IMEのアプリケーションへの影響

Windows8以降で、IMEModeに関連するAPIの仕様が大きく変更されており、Delphi/400のバージョンによっては、IMEの基準が異なるため、特定OSで動作が正しくない場合がある。

Delphi/400 10 SeattleでのIME切替の例)

Windows8、10



Windows7



4.IMEのアプリケーションへの影響

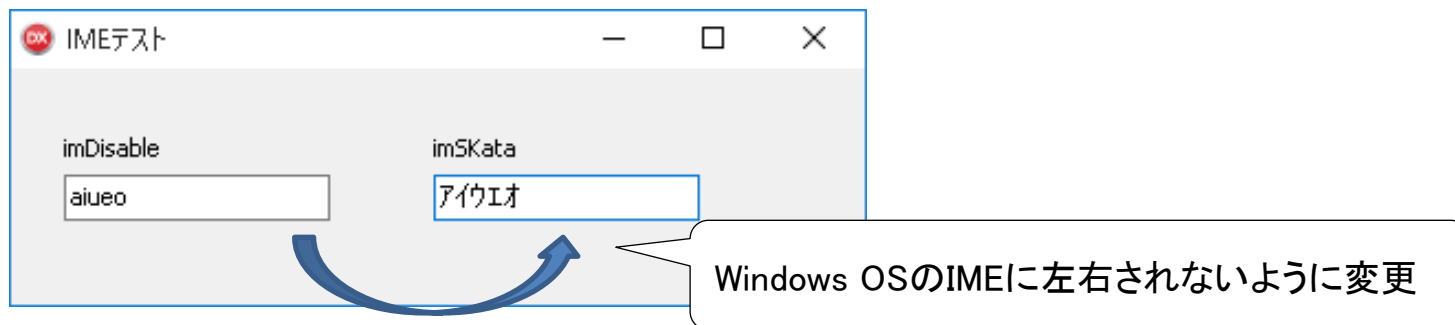
アプリケーションの対処方法

システムソースであるVcl.Controls.pasをローカルにコピーして、変更してプロジェクトに加えることで、対応可能。

例) 10 Seattleのシステムソースパス

C:\Program Files (x86)\Embarcadero\Studio\17.0\source

Delphiのシステムフォルダ内を直接変更しないように注意



4.IMEのアプリケーションへの影響

アプリケーションの対処方法

Vcl.Controls.pasの変更例(10 Seattle)

```
procedure SetImeMode(hWnd: HWND; Mode: TImeMode);  
//中略・・・ロジック最後の部分を変更・追記
```

```
if IsMSCTFAvailable and TOSVersion.Check(6,2) then  
begin  
  SetInputScope( hWnd, InputSourceModeMap[Mode] );  
  ImmReleaseContext(hWnd, IMC);  
  Exit;  
end;
```

Windowsの
OSバージョンを判断して
制御する

```
ImmSetOpenStatus(IMC, TRUE);  
ImmGetConversionStatus(IMC, Conv, Sent);  
Conv := Conv and (not(IME_CMODE_LANGUAGE or IME_CMODE_FULLSHAPE))  
           or ConversionStatusModeMap[Mode];  
ImmSetConversionStatus(IMC, Conv, Sent);  
ImmReleaseContext(hWnd, IMC);  
end;
```

5.Windows64bitOSの考慮点

対象OS: Windows Vista、7、8、10

5.Windows64bitOSの考慮点

64bitOSとは？

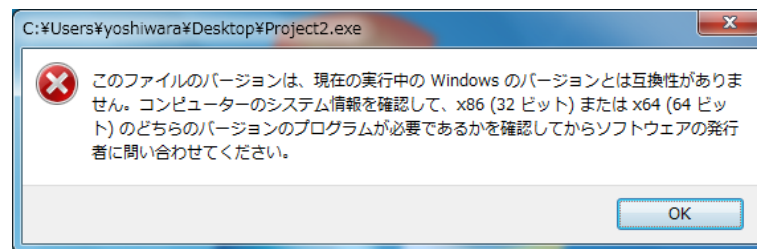
64bitOSとは、64bitマイクロプロセッサに対応したOS、または最適化されたOS。以前はサーバを中心として採用されていたが、Windows7以降はPCでも64bitのOSが増えており、新しくPCを購入する場合は64bitが主流になってきている。

【32bit /64bit アプリケーションの注意点】

64bitOSでは従来の32bitアプリも64bitアプリも動作します。しかし64bitアプリは32bitOSでは動作しません。

	32bitアプリケーション	64bitアプリケーション
32bit OS	○	×
64bit OS	○	○

【64bitアプリを32bit端末で実行】



5.Windows64bitOSの考慮点

64bitOSのシステムフォルダパス

システム関係フォルダや特殊フォルダのパスは、WindowsOSが32bitか64bitかによって異なるものがある。

アプリで絶対パスを使っている場合は見直しが必要。

同じシステムフォルダでもパスが異なる例) 32bitアプリ

32bitOS	64bitOS
C:¥Program Files	C:¥Program Files (x86)
C:¥Windows¥System32	C:¥Windows¥SysWOW64

5.Windows64bitOSの考慮点

アプリケーションの対処方法

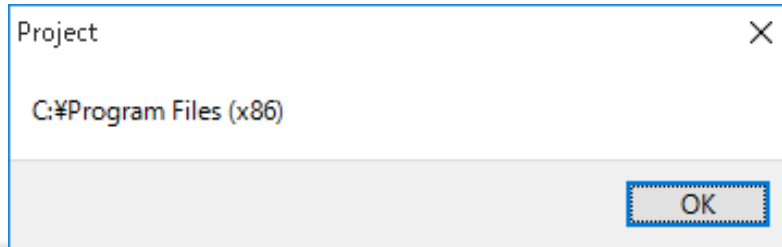
プログラムで環境に合わせたパスを動的に設定することでアプリケーションの対応が可能。

環境変数のシステムパスはGetEnvironmentVariableで取得することができる。

Program Filesパスの取得例

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
  ShowMessage(GetEnvironmentVariable('PROGRAMFILES'));  
end;
```

64bitPCで実行



5.Windows64bitOSの考慮点

GetEnvironmentVariableで扱える環境変数 (Delphiヘルプ)

環境変数名	変数値の説明
ALLUSERSPROFILE	一般的なユーザー プロファイル。
APPDATA	アプリケーション データ フォルダのパス。
CLIENTNAME	クライアント マシンの名前。
COMMONPROGRAMFILES	共通のプログラム ファイル フォルダのパス。
COMPUTERNAME	コードを実行するコンピュータの名前。
COMSPEC	cmd.exe プログラムのパス。
HOMEDRIVE	現在のホームドライブ指定(たとえば 'C:' など)。
HOMEPATH	現在のドキュメント格納場所のパス。
LOGONSERVER	ユーザー ログイン認証用のドメイン コントローラを指定します。
NUMBER_OF_PROCESSORS	現在のマシンに搭載されているプロセッサの数。
OS	オペレーティング システムの基本名。Windows XP は Windows_NT と表記されることに注意。
PATH	現在のプログラム パス。
PATHEXT	実行可能ファイルの拡張子の種類。
PROCESSOR_ARCHITECTURE	CPU アーキテクチャのタイプ。たとえば、Intel Pentium プロセッサの場合は X86。
PROCESSOR_IDENTIFIER	現在のマシンの ID 番号。
PROCESSOR_LEVEL	CPU アーキテクチャの詳細説明。
PROCESSOR_REVISION	プロセッサのリビジョン レベル。
PROGRAMFILES	プログラム ファイル フォルダのパス。
SESSIONNAME	現在の OS セッションの名前。
SYSTEMDRIVE	OS の起動ドライブ。
SYSTEMROOT	システム ディレクトリを設定します。
TEMP	一時ファイル フォルダのパス。
TMP	一時ファイルを格納するディレクトリ。
USERDOMAIN	現在のマシンのドメインを指定します。
USERNAME	現在のユーザーの名前。
USERPROFILE	現在のユーザーの情報が格納されているフォルダのパス。
WINDIR	Windows フォルダのパス

6.Windowsに関連するDelphi/400情報

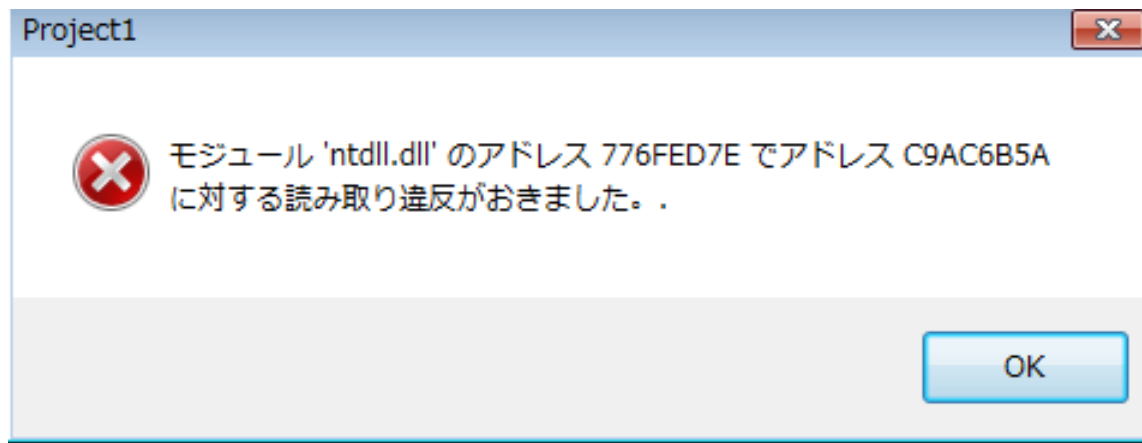
対象OS: Windows7

6.Windowsに関連するDelphi/400情報

特定のWindowsOSで発生するランタイムエラー

同じアプリケーションであっても、特定のWindowsのOSに限定してエラーが発生することがある。

特にWindows7 64bitPC上では、Windowsのランタイムモジュールであるntdll.dllから発生するエラーが多数報告されている。



6.Windowsに関連するDelphi/400情報

アプリケーション・開発環境の対処例

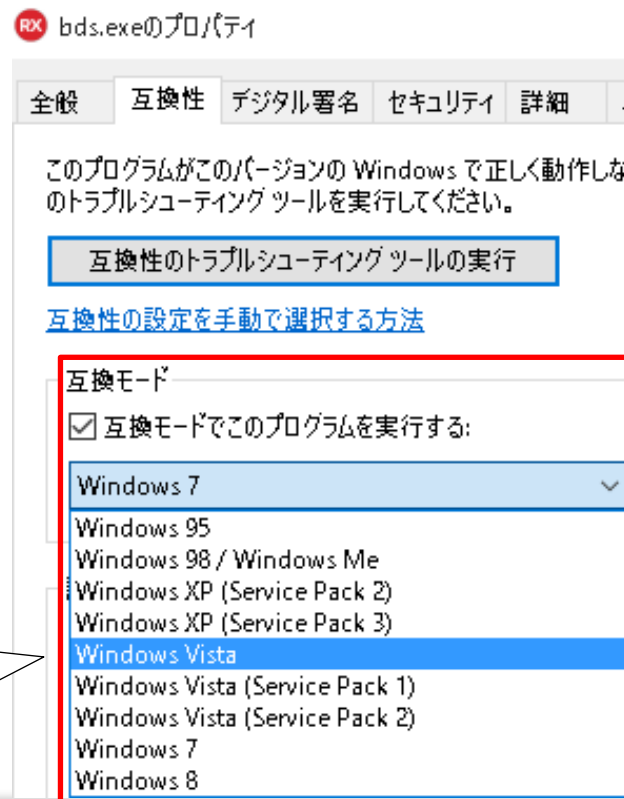
Windowsで修正される可能性はあるが、時期は分からない為、当面の対処をするのであればexeの互換チェックで他のOSを設定しておけば回避できる場合がある。

例)

アプリケーションで発生する場合：
exeの互換性プロパティ

Delphi開発環境で発生する場合：
bds.exeの互換性プロパティ

エラーにならないOSに
アプリ単位で変更する



6.Windowsに関連するDelphi/400情報

特定のWindows・PCで発生する不具合情報

CPUがSkylake（第6世代インテル® Core™ プロセッサー）でWindows7を使用している場合、PCメーカーによって「16bit互換のインストーラが起動しない」不具合報告がある。

Delphi/400でもV2005以前のインストーラでは該当する可能性
がある。メーカーによってBIOSなどの修正が提供されて解決しているので、PCメーカーへの確認が必要となる。
（PanasonicやHP等で解決報告が多数ある。）

7.まとめ

- アプリケーションの表示はWindowsテーマや解像度設定が大きく影響する。特にクラシックは違いが多いので注意が必要。
- 管理者権限がアプリケーションの動作に影響している場合は、プロジェクト設定でアプリケーション毎に制御できる。
- WindowsOSのバージョンによってIME動作に違いもあるので対処方法を把握しておく。
- OS特有のエラーは、アプリケーションの互換設定で回避できる場合もある。
- SkylakeのPCでインストーラが起動しない場合は、メーカーにパッチなどを確認する。