CodeGear from Borland
Developer Camp

Delphi 2007 for Win32によるWindows Vista対応

ボーランド株式会社 Developer Tools Group 高橋智宏

アジェンダ

- Windows Vista による影響
- Delphi for Vista の新機能
- Vistaに対応するには

注:このプレゼンテーションは以下の資料を参考にしています

[MSDN: Windows Vista Application Development Requirements for User Account Control]
http://msdn2.microsoft.com/en-us/library/aa905330.aspx

[Developer Best Practices and Guidelines for Application in a Least Privileged Environment]
http://msdn2.microsoft.com/en-us/library/aa480150.aspx

[Microsoft Windows Vista: 互換性に関するドキュメント]
http://www.microsoft.com/japan/msdn/windowsvista/general/AppComp.aspx

CodeGear from Borland

Developer Camp

Windows Vista

Developer Camp

イノベーションと互換性

- Windows Vista によるユーザー・エクスペリエンスの向上
  - セキュリティ
  - 信頼性
  - 使い易さ
- 多くのアプリケーションはそのまま動作する
- Windows Vista による改良点が互換性を損なうこともある

典型的な互換性の問題

- 管理者権限での動作を想定している (セキュリティ上重要なリソースへのアクセス)
- 古いOSの機能を利用している
- OSのバージョンに強く依存している
- 内部的なシステムコールやデータ構造を使用している
- 潜在的な不具合
- UIの描画に関する問題

ユーザーアカウント制御 (UAC)

- アプリケーションは管理者権限無しで動作する
- 脆弱性が入り込む余地を軽減する
  - マルウェア, トロイ, ウィルス, rootキット
- 問題点
  - インストーラが正しく動作しないものがある
  - 管理者権限での実行を想定して設計されているアプリケーションがある
  - 管理者としての資格を調べるアプリケーションがある
  - 症状 (Silent failure、プロンプト、モーダルメッセージ)

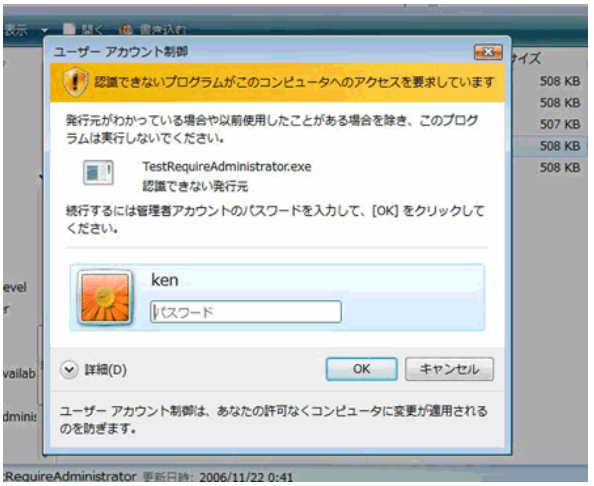
ユーザーアカウント制御 (UAC)

- UACの下でも動作するようアプリケーションを修正する
  - 標準ユーザー権限で動作させる
  - 管理タスクはCOMオブジェクトまたは別の実行可能EXEを使用して行う
  - アプリケーションマニフェストでUAC権限レベルを指定する
  - リダイレクトが必要無いようにする
- 実行時における対処法
  - アプリケーションが権限を昇格できるようにする
  - ファイルやフォルダに対するアクセス制御リスト(ACL)を外す

ユーザーアカウント制御 (UAC)

- セキュリティトークンは、ログイン中分離されています
  - 「**ユーザートークン**」と「**管理者トークン**」
- シェルは「**標準ユーザーのトークン**」で動作しています (管理者であっても)
- プロセスを生成する際には、毎回、明示的に「管理者のトークン」が必要となります。
  - これが「昇格(elevation)」と呼ばれるものです

標準ユーザー から 管理者 への昇格



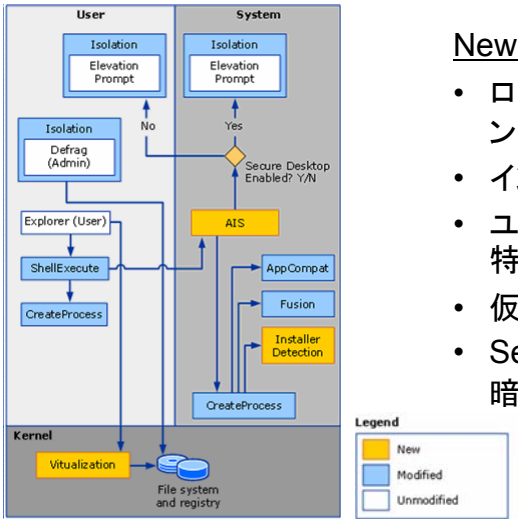
Windows Vista

- デフォルトで、UAC が有効になっている
- 新規に作成されるユーザーアカウントはすべて「**標準ユーザー**」として作成される
- デフォルトで、「**Secure Desktop**(画面の暗転)」上に、「昇格のプロンプト(Elevation Prompt)」が表示される
- バックグラウンドアプリケーションに対する「**昇格のプロンプト**(Elevation Prompt) 」は、タスクバー上に最小化されて表示される
- デフォルトで、新規インストール時、ビルトインの管理者アカウントは無効になっている
- 新しいアクセス制御リスト(ACL)のデフォルト設定

標準ユーザー

- デフォルトでは、プロセスは標準ユーザー権限で起動する
- 標準ユーザーが行えないこと
  - 「**Program Files**」フォルダ内のファイルに対する変更
  - 「**Windows**」または「**System32**」フォルダ内のファイルに対する変更
  - 「**HKLM¥Software**」以下にあるレジストリの変更
  - ローカルコンピュータの日時の変更
  - 「**サービスアプリ**」のインストールとアンインストール
  - …
- これまで推奨されてきたことが、ついに強制されるようになった！

UACのアーキテクチャ



New!

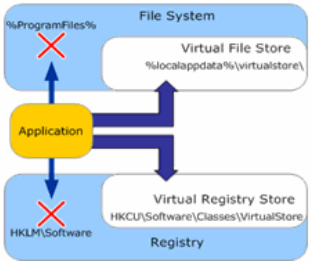
- ログイン中のアクセストークンの分離
- インストーラの検出
- ユーザーインターフェース特権の分離(UIPI)
- 仮想化
- Secure Desktop（画面の暗転）

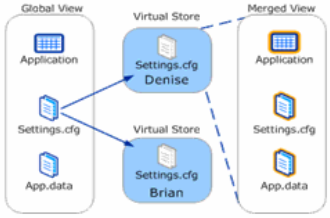
ユーザーインターフェース特権の分離 (UIPI)

- **一般的なガイドライン – 「低い権限」では「高い権限」にアクセスできない**
- 低い権限のプロセスで出来ないこと
  - ウィンドウハンドルの検証
  - SendMessage や PostMessage
  - スレッドフックを利用したアタッチ
  - ジャーナルフックを利用したモニタリング
  - ダイナミックリンクライブラリ(DLL)を使ったインジェクション
- いくつかのリソースは依然としてプロセス間で共有される
  - デスクトップウィンドウ（実際にデスクトップ表面を覆っているもの）
  - 読み込み専用の共有メモリとしてのデスクトップヒープ
  - グローバル atom テーブル
  - クリップボード

仮想化／リダイレクト

- 仮想化は互換性のためにある – 機能ではありません
- アプリケーションマニフェストにUAC情報を設定することで無効化できる！

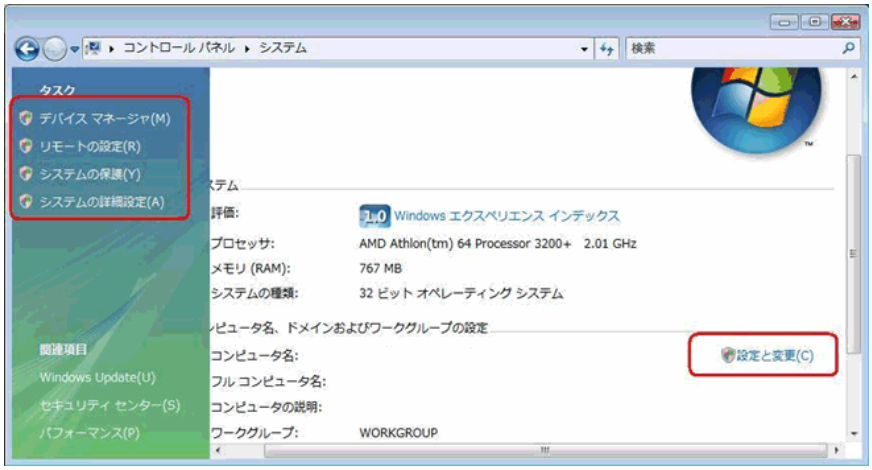




「Shield」- アイコン

- クリックして次の手順に移行するのに「昇格(elevation)」を必要とする場合に、コントロールに付加する
- 1つのステータスしか持たない(例：ホバリングやDisable状態などは無い)

「Shield」の例



Windows リソース保護(WRP)

- システムを保護する: ファイル／フォルダ／レジストリキー
- OSだけが保護されたリソースを更新できる
- 問題点
  - システムのバイナリリソースの置き換え
  - システムのレジストリキーへの書き込み

- 絶対にWindowsシステムのファイルやレジストリキーを置き換えない
- 絶対にMicrosoftの再配布パッケージをリパッケージしない

アプリケーションの更新

- アプリケーションの更新には管理者権限が必要
- 問題点:
  - 標準ユーザーは管理者権限を持たない
  - アプリ更新プログラムとして、別のEXEを使用する
- 対処法:
  - 更新プログラムは、マニフェスト付きの別プロセスにすべき
  - MSIを使ったパッチ適用
  - ClickOnce技術を使用する

サービスアプリの分離

- サービスアプリは独自のセッション(セッション0)で動作します
- ユーザーアプリはサービスアプリとセッションを共有することは出来なくなりました
- 問題点
  - サービスアプリはユーザーのデスクトップやアプリケーションと直接対話することはできない
  - ユーザーとの対話を行うサービスは、UIに表示されないためハングアップするかもしれません
  - セッションやグローバルな名前付きのオブジェクトの作成

サービスアプリの分離 (続き)

- アプリケーションを修正する
  - サービスアプリはUIを使用しないようにすべき
  - ユーザーアプリはセッション0での動作を想定すべきではない
  - ユーザーアプリは「Fast User Switching」をサポートするか否か
  - サービスアプリとオブジェクトを共有するユーザーアプリは、グローバルなオブジェクトを使用すべき
  - グローバルな名前付きのオブジェクトはセッション0で作成する
- OSによる対処法
  - セッション0でのUI表示がある場合には、現在のユーザーに対して通知が行われる

Internet Explorer の保護モード

- Internet Explorer は、極めて低い権限で動作します
  - IE は、ユーザーのファイルやレジストリキーを変更できません！
  - ファイルやレジストリへの書き込みは、IEだけに見える場所へとリダイレクトされます
  - ウィンドウメッセージはブロックされます
- 問題点
  - 外部プロセスとデータを共有しているコントロールは動作しない
  - ユーザーの許可を求める新しいプロンプト

Internet Explorer の保護モード(続き)

- 対処法
  - マニフェスト付きのOCXのインストーラを使用する
  - 外部プロセスと通信する場合はCOMオブジェクトを使用する
  - サイトを信頼済みサイトの一覧に追加する
  - IEの保護モードを無効にする(低MIC)

バージョンチェック

- アプリケーションがOSの特定のバージョンをチェックしている
  - 依存関係のチェックのため
  - 製品のマーケティング上の制約のため
- 症状はアプリケーションごとに異なる
  - インストールや起動が失敗する(Silent failure)
  - 「サポートされないOSです」等のメッセージが表示される
- 対処法
  - アプリケーションは、サポートOSの最小バージョンを決定しておくべき

USER／GDI／DPI

- 新しいディスプレイドライバモデルと、Desktop Window Manager(DWM)
- テストが必要な新しい領域
  - サムネイル
  - Flip 3D
  - ドラッグアンドドロップ
  - アプリケーションがポップアップ表示するもの
  - 高DPI
  - スクリーンの回転
  - マルチモニタのサポート
  - 新しいUIテーマ

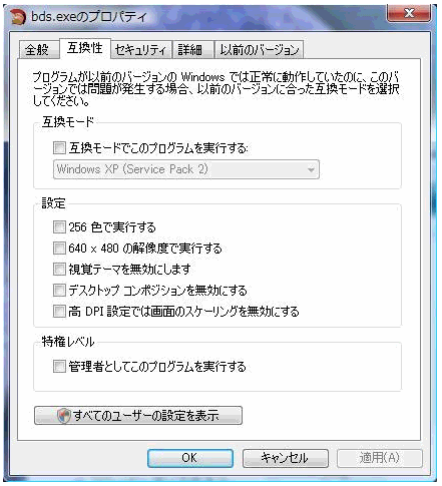
その他

- シェルの変更
  - 「マイドキュメント」フォルダの構造が変更された – そのパスをハードコードしているアプリケーションは動作しないかもしれません
  - Fast User Switching(高速ユーザー切り替え)
- 「Full text search API」の変更
- 「IIS 7」の変更
- バックアップソフト向け「ボリュームシャドウコピーサービス(VSS)」のWriterの変更
- 新しい「Server core edition」はほとんどUIを必要としなくなりました

削除されたコンポーネント

- 以下の機能は利用できなくなりました
  - FrontPageサーバーの拡張
  - POP3 サーバー
  - Macintosh向けのサービス
  - カーネルモードプリンタドライバ
  - **WinHelp** (HTMLヘルプ と .CHMファイル はサポートされます)
  - ビデオオーバーレイ

プログラムの「互換性」タブ



設計の指針

- ユーザー固有のデータは、ユーザーのフォルダ／ファイル／レジストリキーに格納する
  - アプリケーションを標準ユーザーで動作するように設計する
  - 「C:¥ProgramData」か「%UserProfile%」に書き込む
  - レジストリの値は「HKCU」に書き込む
  - アプリケーションのインストールには「MSI」を使用する
- 必要があるまで不用意にファイルやレジストリを書き込みモードでオープンしない

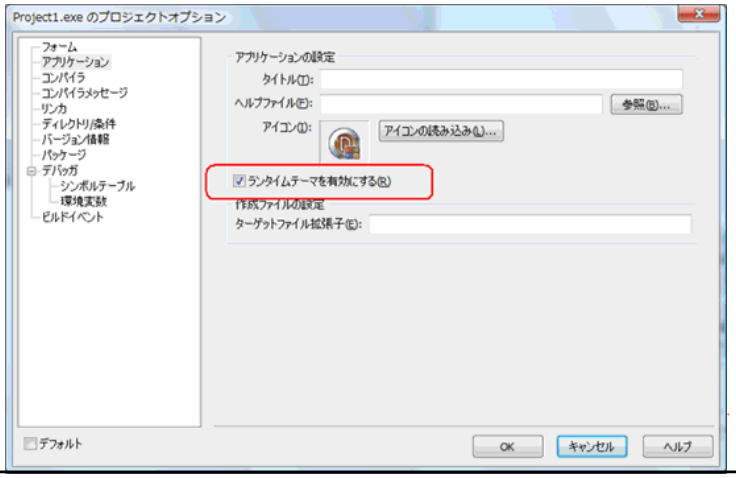
参考情報

- ドキュメント
  - Windows Vista Application Development Requirements for User Account Control Compatibility
  - Developer Best Practices and Guidelines for Application in a Least Privileged Environment
  - http://www.microsoft.com/japan/msdn/windowsvista/general/AppComp.aspx
- ツール
  - Microsoft Standard User Analyzer(SUA)
- Windows Vista Logo Program
  - http://microsoft.mrmpslc.com/InnovateOnWindowsVista/

Delphi 2007 の新機能

テーマ

- XPManユニットはプロジェクトオプションに置き換わりました



新しいコンポーネント

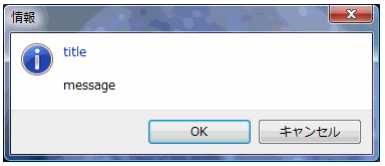
- **TTaskDialog**
  - 新しい TaskDialogIndirect APIをラップ
- **TFileOpenDialog**
  - 新しい IFileOpenDialogインターフェースをラップ
- **TFileSaveDialog**
  - 新しい IFileSaveDialogインターフェースをラップ
- フォームデザイナ上での「ダイアログのテスト」メニュー(プレビュー機能)が追加

新しいプロパティ

- TCustomForm.**GlassFrame**
  - グラス効果のウィンドウフレームを、フォームのクライアント領域へも拡張して使用できる
  - 「**SheetOfGlass**」プロパティを**True**に設定する必要がある(デフォルトはFalse)

新しい関数

- **TaskMessageDlg**
  - MessageDlgに相当するもの。Vista以外のOS上で呼び出された場合は、従来の*MessageDlg*が呼び出されます。



Vista上

XP上

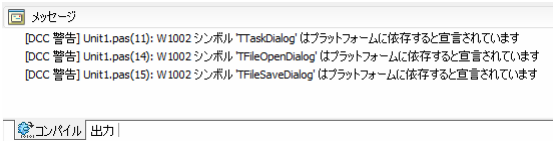
新しいグローバル変数

- **UseLatestCommonDialogs**
  - Trueに設定されていた場合、TOpenDialog／TSaveDialog／TOpenPictureDialog／TSavePictureDialog／MessageDlgを使用すると、最新のVista環境に合ったダイアログが表示される (アプリケーションがVistaで動作している場合)
  - Falseに設定すると、これまでの動作と同じになる
  - デフォルトはFalse(OFF)

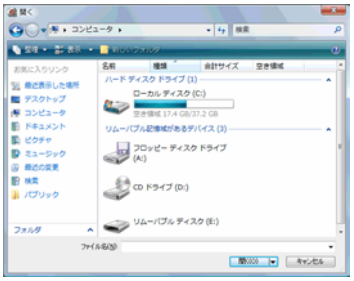
新しい例外

- EPlatformVersionException

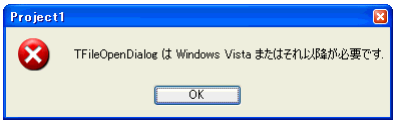
ビルド時



Vista上



XP上(例外が発生)



Windows API に関する変更

- UxThemes – 新規
- DwnApi – 新規
- ActiveX – アップデート
- Windows – アップデート
- Messages – アップデート
- CommCtrl – アップデート
- ShlObj – アップデート

DBX4(dbExpress Ver4)

- 新しいDelphiデータベースドライバフレームワーク
- DBX4の目標
  - BDPとDBXのドライバテクノロジーを1つに統合する
  - ADO.NET 2.0 のサポート
  - UNICODEのサポート
  - DBX VCL コンポーネントを単一ソースにする
  - DBXドライバの「共通」の実装部分のソースコードを提供
  - DBXドライバの開発をアウトソースすることを可能にする

Developer Camp

TDBXフレームワーク

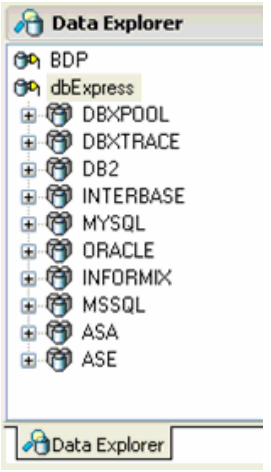
- ドライバフレームワークは完全にDelphiだけで記述されている
- ポインタ(Pointer)型ではなく、強く型付けされたデータ型によるアクセス
- TDBXドライバフレームワークは**単一ソース**である。単一のソースコードにより、ネイティブな*dcc32コンパイラ*と マネージド*dccilコンパイラ*によるコンパイルが可能
- TDBXのドライバ、コネクション、コマンド、リーダーなどは、抽象基本クラスを使用している
- 例外ベースのエラー処理

CodeGear
Where developers matter

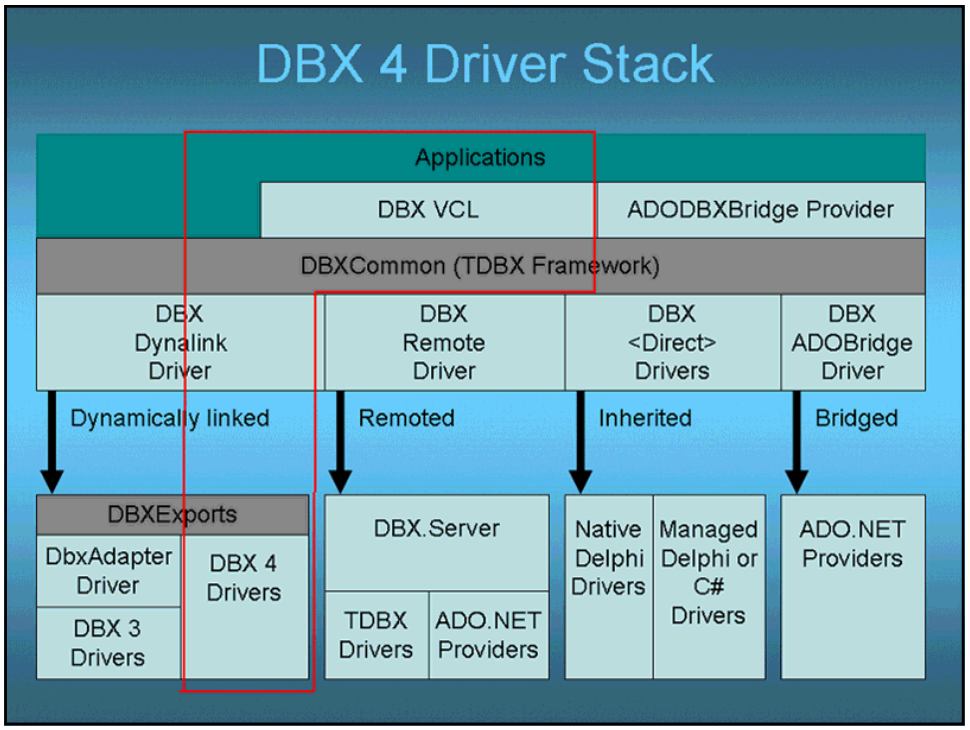
Developer Camp

TDBX と Delphi for Win32

- 既存のDBXアプリケーションは、最小限の変更で動作する
- UNICODEのサポート
- ソースコードが付属
- 100% Delphiで記述
- DBXドライバの仕様をサードパーティーベンダーに公開



CodeGear
Where developers matter

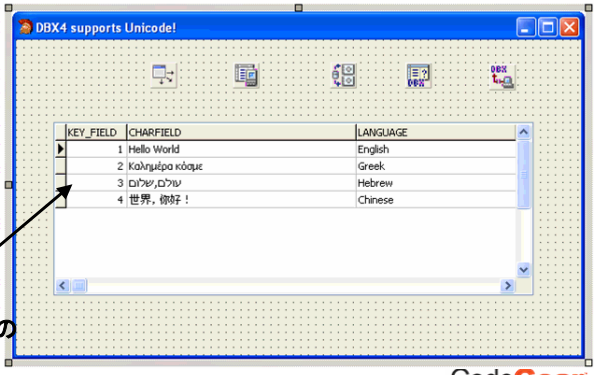


互換性

- dbExpressを使用しているほとんどのVCLアプリケーションは、変更せずにコンパイル&実行ができる
- デュアルインターフェースを実装したドライバ
- ドライバアダプター
- パッケージの利用
- ユニットのインターフェースの互換性
- MySQL用の新しいDBXドライバ

DBX4のUNICODEサポート

- InterBase 2007
- Oracle 10g
- SQL Server 2000/2005
- MySQL 4.1/5.0



TntWare Delphi Controls の TntDBGridコンポーネント

http://www.tntware.com/delphicontrols/unicode/

デリゲートドライバ

- TDBXTrace
  - フレームワークレベルのトレース機能
  - フレームワークの利用状況を示すDelphiコードを出力する
- TDBXPool
  - コネクションプール機能
- デリゲートドライバのチェイン(連結)

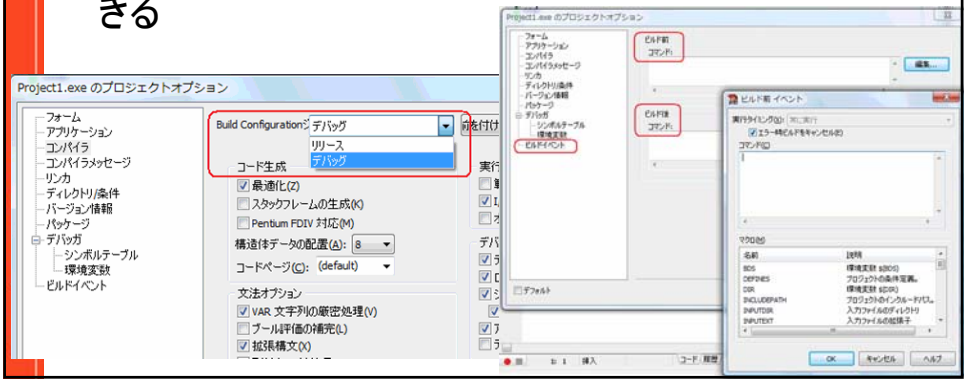
Developer Camp

新しいビルドエンジンの概観

- プロジェクトファイルの変更
  - .bdsproj から .dproj へ
- MSBuildがベース
- ビルドを記述するための完全にオープンで一般的なXML ファイル形式
- カスタマイズと拡張が可能
- 開発者は(タスクやロガーの)マネージドコードを記述することによりビルドの拡張が可能
- 拡張された Delphi 2007 for Win32のビルドは、マルチプロダクトおよびマルチプラットフォームを同時にサポート

新しいビルドエンジンは

- プロジェクト(ビルド)設定を複数持てる
  - リリース用, デバッグ用 など
- ビルド処理の前後のイベントで、コマンドを実行できる



CodeGear from Borland

Vistaに対応する

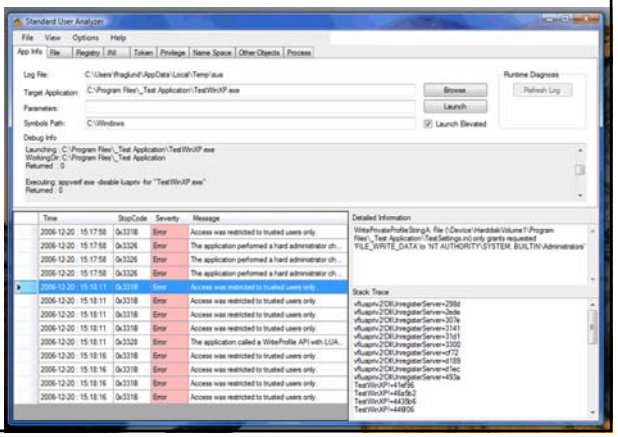
Delphi – 行うこと…

- アプリケーションをテストする – 問題を確認する
- アプリケーションを、標準ユーザー／管理者／その複合形態で検証する
- **アプリケーションマニフェストを追加する**
- 機能を再設計
  - **ユーザーアプリは、正しい場所にデータを書き込むべき**
  - 管理者の機能は、別の実行ファイルに分離する
- ユーザーインターフェースの再設計
  - **ボタンに「Shield」アイコンを追加する**
- インストーラの再設計
- 繰り返しテストする
- 必要があればアプリケーションに署名する (Authenticode)

Developer Camp

「Standard User Analyzer」ツールを使ってテストする

- SUAは、アプリケーション内で行っている問題のある処理を発見するのに役立つ

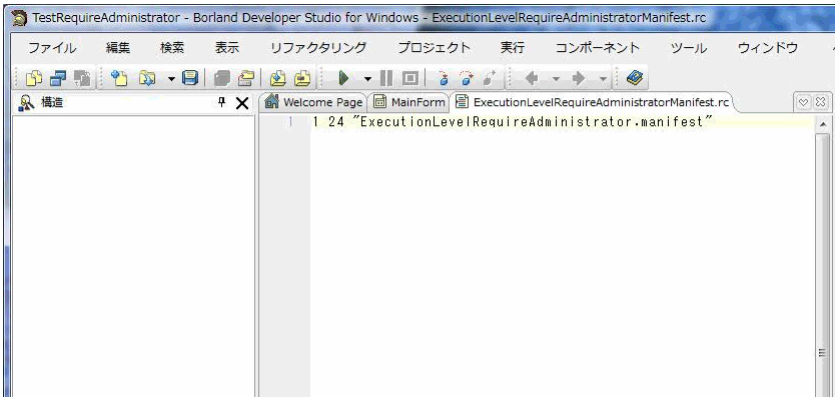


アプリケーションマニフェスト

マニフェストファイルの例

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
  <dependency>
    <dependentAssembly>
      <assemblyIdentity
        type="win32"
        name="Microsoft.Windows.Common-Controls"
        version="6.0.0.0"
        publicKeyToken="6595b64144ccf1df"
        language="*"
        processorArchitecture="x86" />
    </dependentAssembly>
  </dependency>

  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v2">
    <security>
      <requestedPrivileges>
        <requestedExecutionLevel
          level="asInvoker"/>
      </requestedPrivileges>
    </security>
  </trustInfo>
</assembly>
```

RCファイルを RESファイルにコンパイル



Requested Execution Level

- XPManユニットへの参照をすべて削除する
- 別のレベルを指定した独自のマニフェストを使用する

```
program TestAsInvoker;

{$R 'ExecutionLevelAsInvokerManifest.res' 'ExecutionLevelAsInvokerManifest.rc'}

uses
  Forms,
  MainForm in '..\Common\MainForm.pas' {Form2};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TForm2, Form2);
  Application.Run;
end.
```

<requestedExecutionLevel /> の説明

- level=”asInvoker”
  - プロセスを作成するユーザーと同じトークンを使用してプロセスを起動する
- level=”highestAvailable”
  - 管理者は、昇格への同意を求められる。しかし、もしユーザーに管理者権限が無い場合は、標準ユーザーとして起動される
- level=”requireAdministrator”
  - 管理者は、昇格への同意を求められる
  - 標準ユーザーは、「管理者への昇格」のためのログイン・ダイアログが表示される
  - 管理者権限で起動する

Windows XP での注意！

- 不正なフォーマットのマニフェストは、Windows XP上で「**ブルースクリーン**」を引き起こすことがある
- KB921337 を参照してください
  - http://support.microsoft.com/kb/921337

再設計

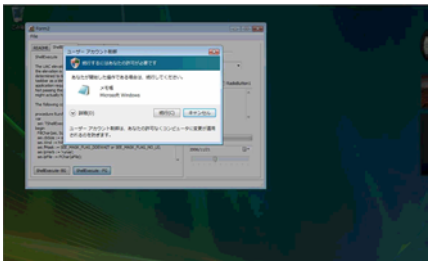
- ファイルやレジストリキーは、書き込みフラグ付きでオープンしない
- データやログファイルは、SHGetFolderPath関数を使用して正しい場所に保存する
  - CSIDL_PERSONAL { My Documents }
  - CSIDL_APPDATA { Application Data, new for NT4 }
  - CSIDL_LOCAL_APPDATA { non roaming, user¥Local Settings¥Application Data }
  - CSIDL_COMMON_APPDATA { All Users¥Application Data }
  - CSIDL_MYPICTURES { My Pictures, new for Win2K }
  - CSIDL_COMMON_DOCUMENTS { All Users¥Documents }
  - …

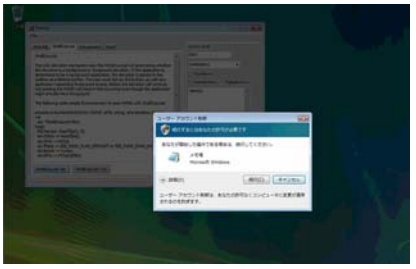
SHGetFolderPath関数の利用

```
uses
  SHFolder;

function GetFolder(csidl: Integer; ForceFolder: Boolean = False): string;
var
  i: Integer;
begin
  SetLength(Result, MAX_PATH);
  if ForceFolder then
    SHGetFolderPath(0, csidl or CSIDL_FLAG_CREATE, 0, 0, PChar(Result))
  else
    SHGetFolderPath(0, csidl, 0, 0, PChar(Result));
  i:= pos(#0, Result);
  if i > 0 then
    SetLength(Result, Pred(i));
end;

function GetLocalAppDataFolder(ForceFolder: Boolean = False): string;
begin
  Result:= GetFolder(CSIDL_LOCAL_APPDATA, ForceFolder);
end;
```

RunAsAdminの実装

- 別プロセスを「プロンプト」付きで起動する
- 「lpVerb」に「runas」を使用する
- ShellExecuteEx関数に渡す親ウィンドウハンドルで、「プロンプト(elevation)」の表示場所が変わる





RunAsAdminの実装

```
// Vista Utilities

procedure RunAsAdmin(hWnd: HWND; aFile: string; aParameters: string);
var
  sei: TShellExecuteInfoA;
begin
  FillChar(sei, SizeOf(sei), 0);
  sei.cbSize := sizeof(sei);
  sei.Wnd := hWnd;
  sei.fMask := SEE_MASK_FLAG_DDEWAIT or SEE_MASK_FLAG_NO_UI;
  sei.lpVerb := 'runas';
  sei.lpFile := PChar(aFile);
  sei.lpParameters := PChar(aParameters);
  sei.nShow := SW_SHOWNORMAL;
  if not ShellExecuteEx(@sei) then
    RaiseLastOSError;
end;
```

「Shield」- SetElevationRequiredState関数の実装例

- ボタンコントロールをパラメータに渡して呼び出すと、「Shield」アイコンが追加／削除される
- BCM_SETSHIELDメッセージまたは、Button_SetElevationRequiredStateマクロを使用

```
const
  BCM_FIRST = $1600; // Button control messages
  BCM_SETSHIELD = BCM_FIRST + $000C;

procedure SetElevationRequiredState(aControl: TWinControl; Requiered: Boolean);
var
  lRequired: Integer;
begin
  lRequired := Integer(Requiered);
  SendMessage(aControl.Handle, BCM_SETSHIELD, 0, lRequired);
end;
```

Authenticodeを使って署名する

- それほど重大ではない感じのプロンプトが表示される
- http://winqual.microsoft.com/ にて登録する
- 証明書を購入する (ベリサイン社などから)
- 実行ファイルに署名する (Makecert.exe と Signtool.exe)
- クラッシュした際のログ情報にアクセスするために、アプリケーションを「winqual」に登録する

CodeGear from Borland

Thank you