embarcadero®

Developer Camp

【B2】Delphiテクニカルセッション

「Delphiのマイグレーションを行うのであれば、これだけは知っておこう！」

株式会社　フルネス
シニアマネージャー　石井　智

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。

はじめに

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。

はじめに

本テクニカルセッションは、旧Delphi（特にDelphi 5～7）で作成されたアプリケーションを、最新版のDelphi XE5でマイグレーションすることに主眼を置いております。

今までの資産（ソース等）をそのまま利用し、最小限の労力でWindows7/8に対応することを目指しています。

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。

アジェンダ

- Delphi 7以前で作成されたアプリケーションの現状
- マイグレーションに際して
- XE5（VCL）への移行
- データベースアクセスの移行
- 帳票の移行
- クロスプラットフォーム対応への移行
- まとめ

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。

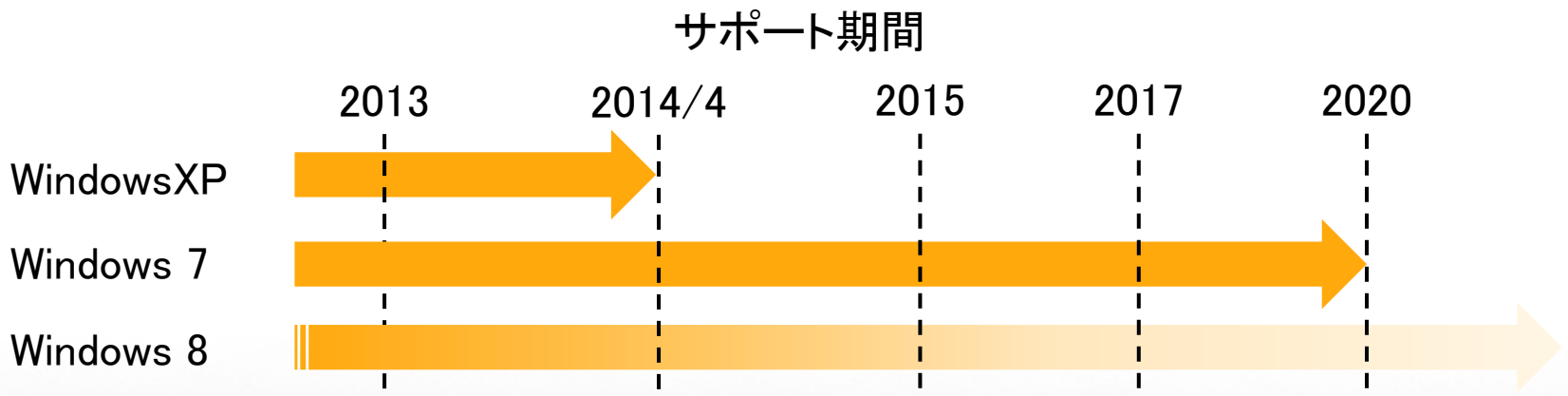
Delphi 7以前で作成されたアプリケーションの現状

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。

Delphi 7以前で作成されたアプリケーションの現状

過去のDelphiで作成されたアプリケーションが、約10年間程度使用されており、今後もアプリケーションを使用し続けるユーザが多数存在します。

その一方で、MicrosoftのWindowsX Pサポートが来年4月に終了となるため、すべてのユーザにおいて、Windows 7/8への移行が迫られています。

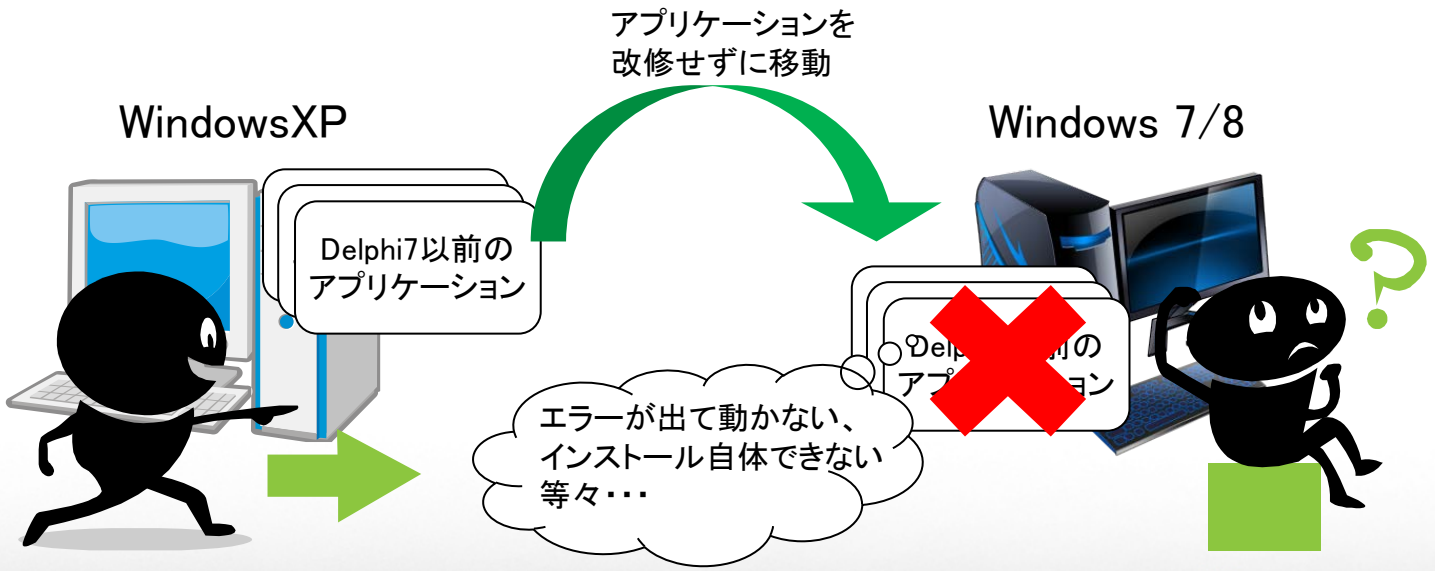


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

Delphi7以前で作成されたアプリケーションの現状

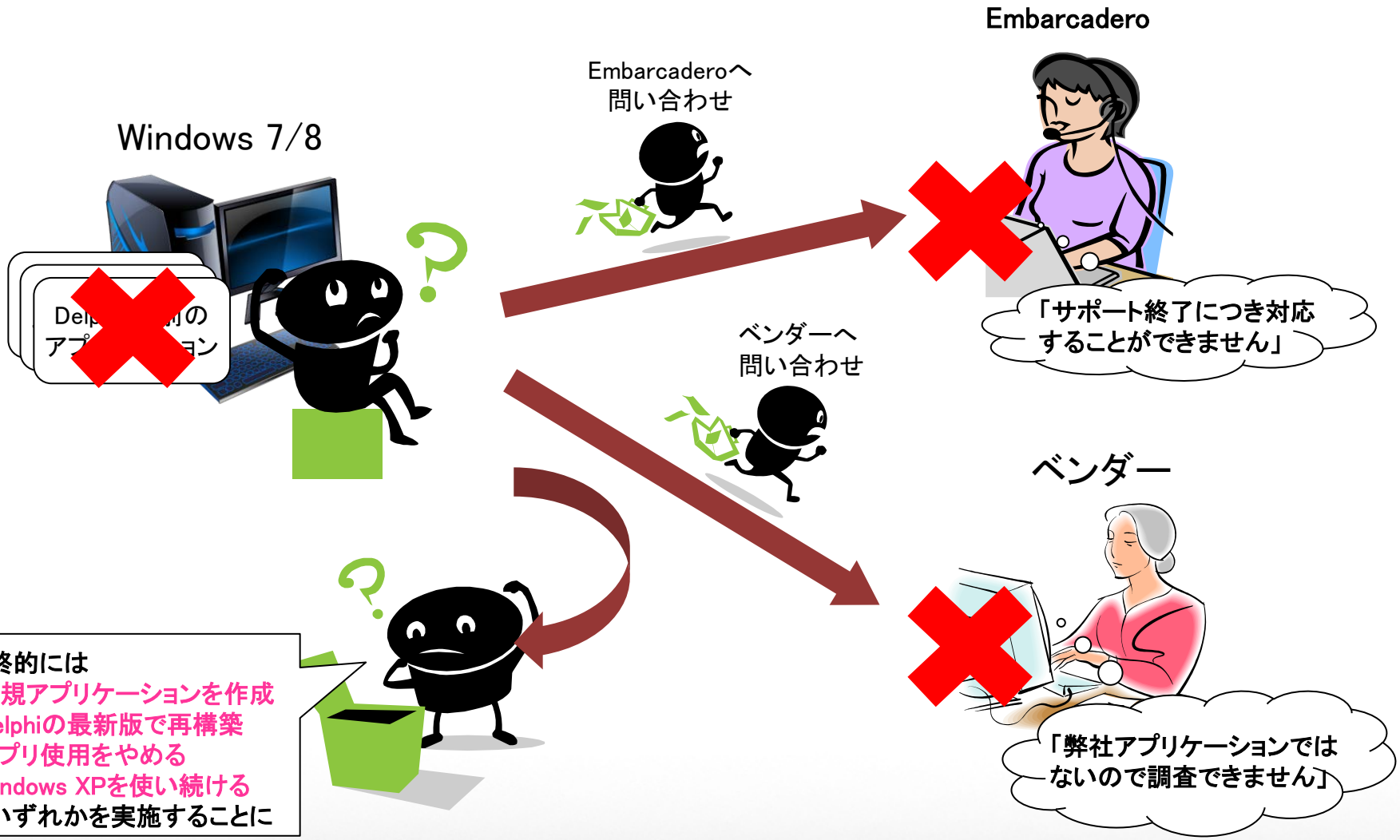
しかし、Delphi 7以前に作成されたアプリケーションは、一部またはすべてにおいてWindows 7以降では動作しない状態となっています。
また、ベンダー（現在はエンバガデロ）においても、Delphi 7以前は既にサポートが終了しています。

※現在、Embarcaderoでサポートされているのは、Delphi 2007およびXE3以降です。



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。

Delphi 7以前で作成されたアプリケーションの現状



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。

Delphi 7以前で作成されたアプリケーションの現状

現行のアプリケーションがWindows 7や8で動作しない場合、幾つかの選択肢があります。

セキュリティの危険性を承知でWindows XPを使い続ける

別途開発ツールを使用し新規のアプリケーションとして作成する（別言語）

現行のアプリケーションの使用をやめる

Delphiの最新版（XE5）で再構築する

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。

マイグレーションに際して

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。

マイグレーションに際して

Delphi XE5でマイグレーションするにあたり、以下の流れで現行アプリケーションを調査します。

- アプリケーション全体のソースが存在するかチェック
- データベースを使用しているかチェック
- 帳票を使用しているかチェック
- サードパーティ製のコンポーネントを使用しているかチェック
- Windows API等を直接呼び出しているかチェック

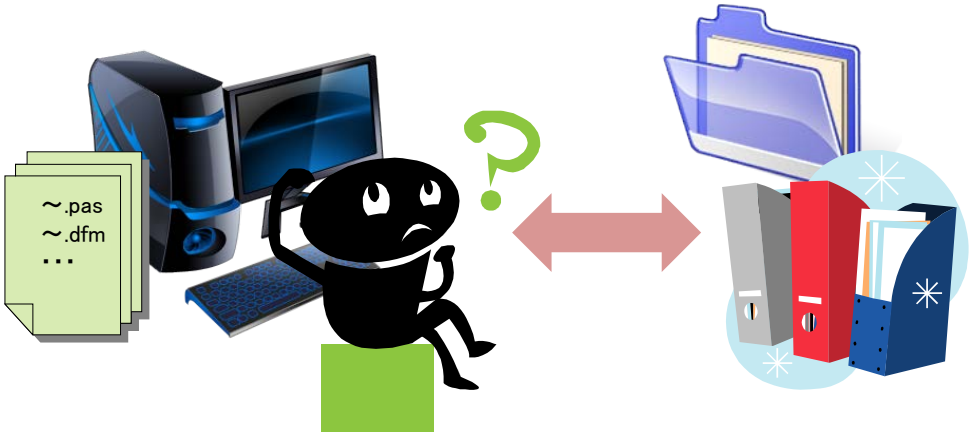
これらのチェックを行ったうえで、最良のマイグレーション方法を選択します。

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。

マイグレーションに際して

●ソースの有無チェック

マイグレーションに際して、一番重要な確認事項として、アプリケーション全体のソースがすべて確保できているか、確認します。
※設計書等が存在せず、ソースのみある場合は、旧アプリケーションのプロジェクトを開いて、プロジェクト内のファイルを確認します。



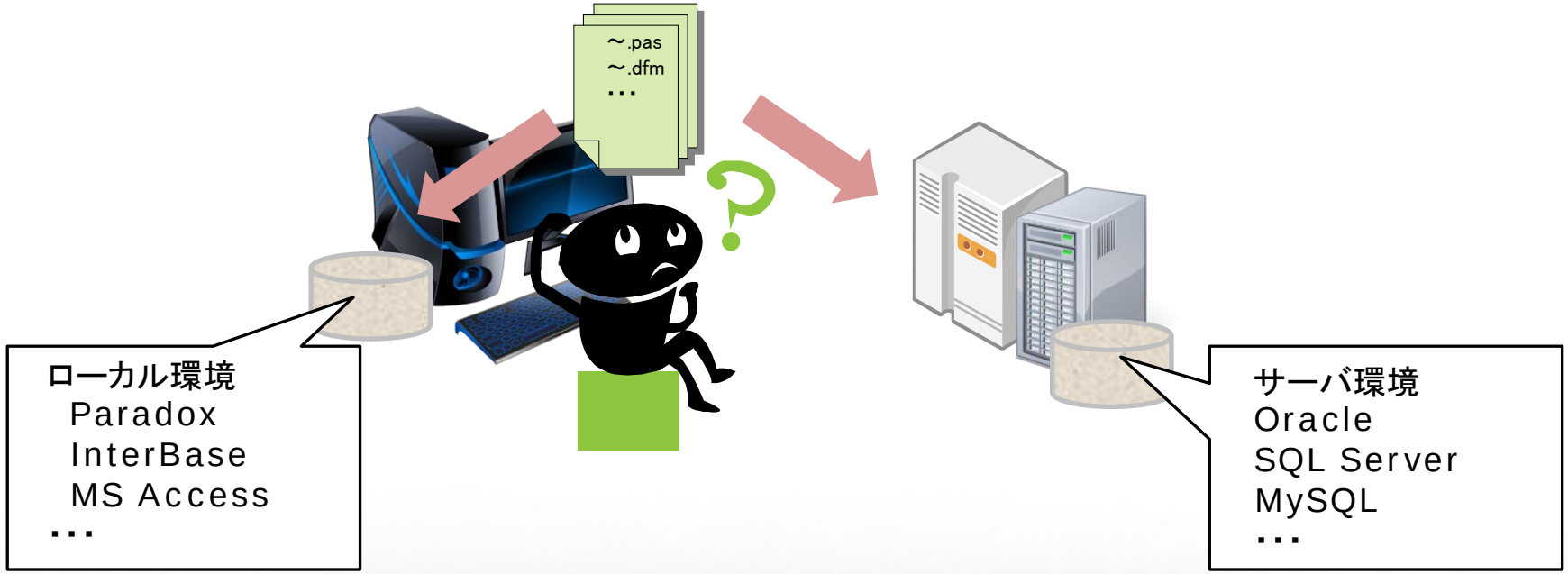
ソースが一部またはすべてが存在しない場合、XE5上でマイグレーションすることができなくなり、ソースが無い部分は新規に作成する必要が出てきます。
アプリケーション全体を新規に作成することに・・・・なるかも知れません。

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。

マイグレーションに際して

●データベースアクセスの確認

アプリケーションがデータベースにアクセスしているか確認します。確認する際、接続に使用しているコンポーネントも把握します。
※データベースをアクセスしていなければ、確認の必要はありません。

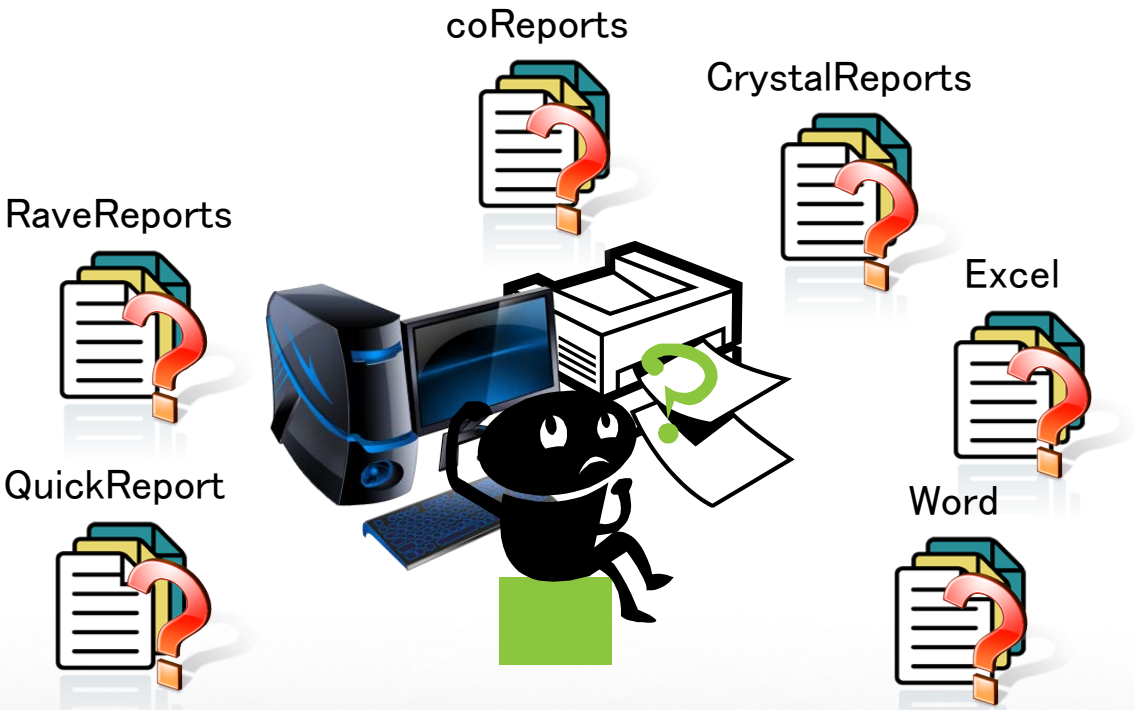


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

マイグレーションに際して

●帳票の確認

アプリケーションで、帳票を出力しているか確認します。
帳票出力で、ExcelやWordのテンプレートを使用して印刷するアプリケーションもありますので、合わせて確認します。

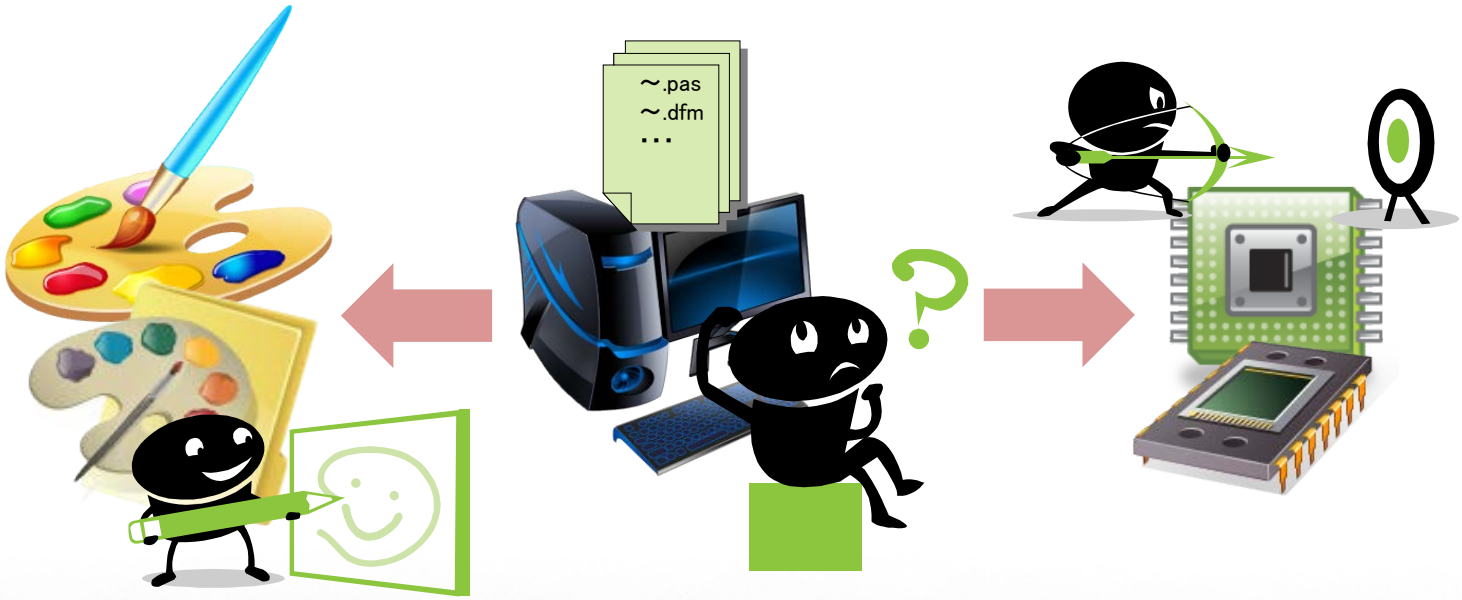


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

マイグレーションに際して

●サードパーティ製コンポーネントの有無

サードパーティ製のコンポーネントを使用しているか確認します。画像処理コンポーネントや制御系で使用するコンポーネントなどが該当します。
※サードパーティ製のコンポーネントは、DLLやOCX形式で提供されているものです。

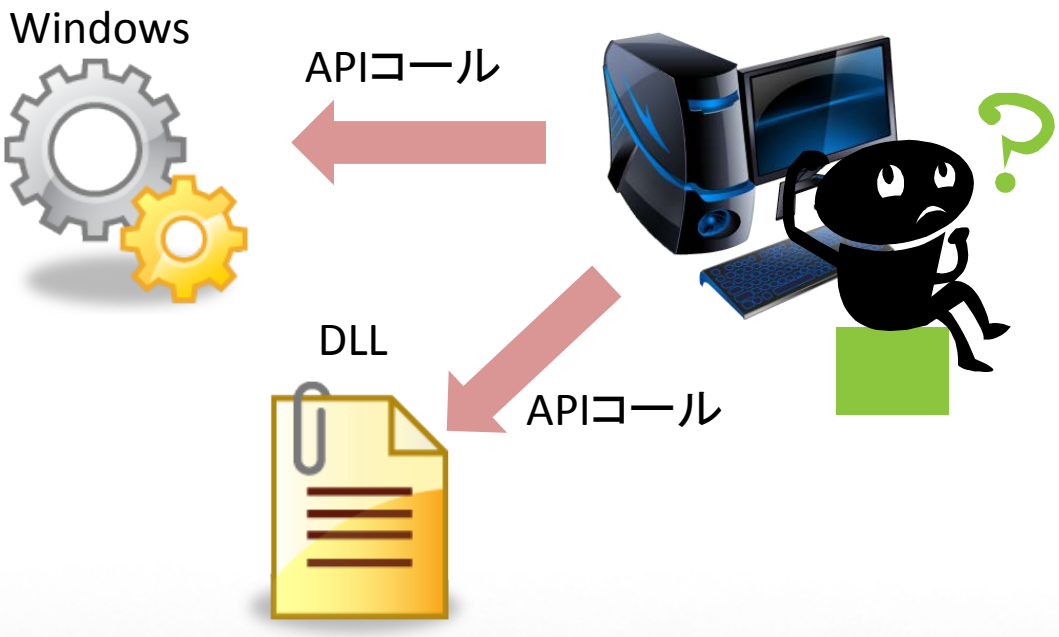


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

マイグレーションに際して

●API呼び出し等の確認

DLLの呼び出しやWindows APIの呼び出しを直接行っているか確認します。
API呼び出しで使用しているパラメータに設定されている変数の型が、XE5では変更になっている可能性があるためです。



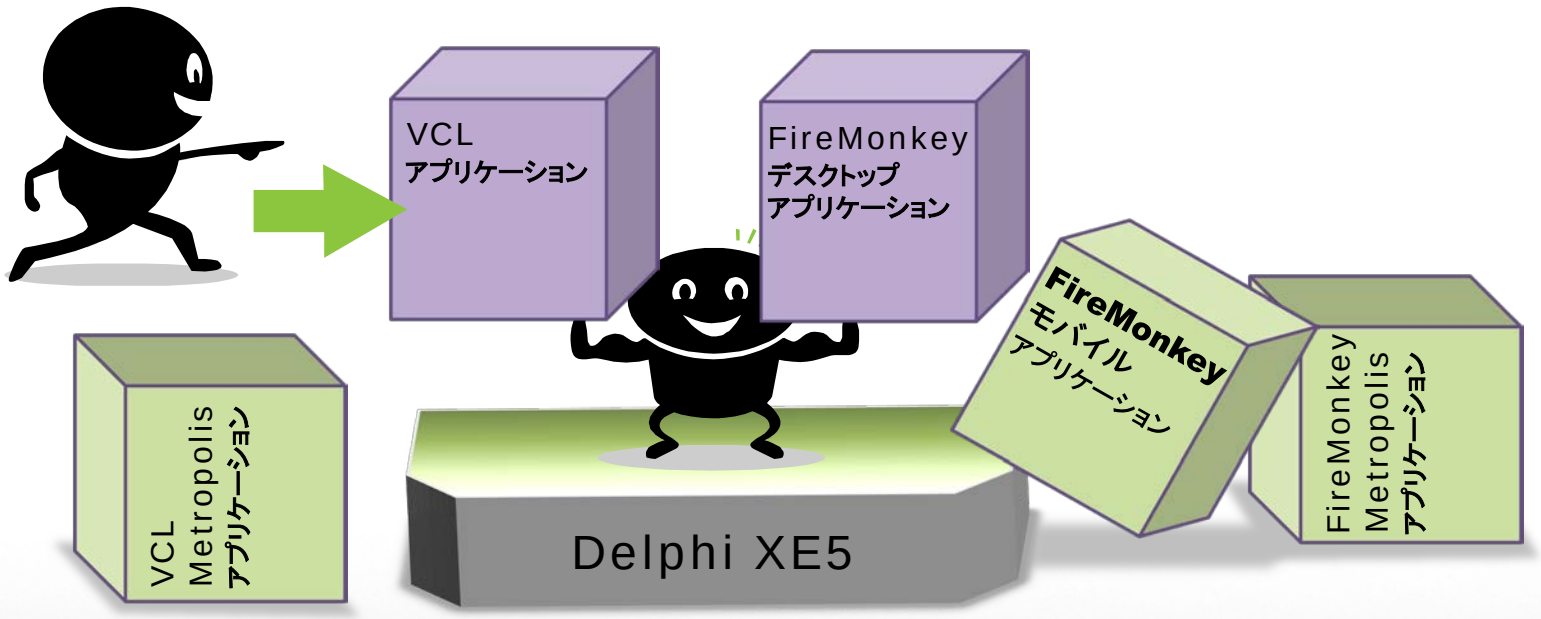
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

XE5（VCL）への移行

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。

XE5（VCL）への移行

Delphi XE5では、以前のDelphiから受け継がれるコンポーネントライブラリ（VCL）と、クロスプラットフォーム開発を目的にしたコンポーネントライブラリ（FireMonkey）が存在します。
まずは、既存アプリケーションをDelphi XE5のVCL版へマイグレーションすることを考えてみます。



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。

XE5（VCL）への移行

Delphi XE5のVCLフォームアプリケーションへ移行する場合は、次頁以降の①～④の手順で移行を行います。

この時、dBASE/Paradoxを使用しているデータベースアクセスの部分はそのままで移行します。 Delphi XE5には、dBASE/Paradoxを使用したアプリケーションをメンテナンスできるように、BDEコンポーネントが搭載されています。（SQL Link を使用している場合は修正が必要です）。

新規開発を行うには、推奨されていませんので、移行時の一時的な動作確保のみに使用します。

※BDE以外のデータベースコンポーネントは、そのまま使用できます。

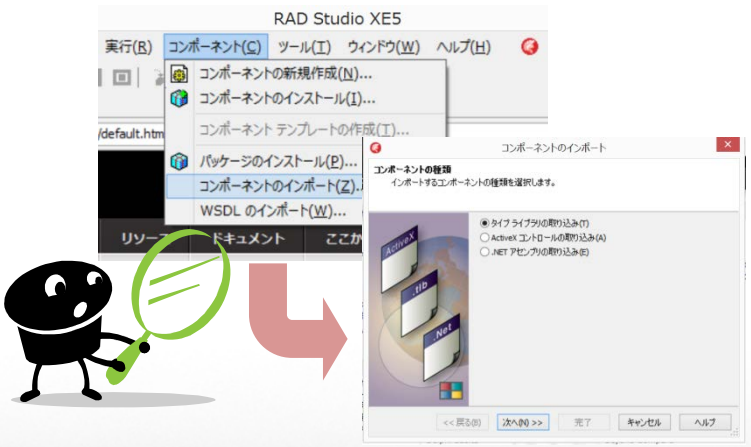
帳票コンポーネントについては、この移行作業時に同時に進める必要があります。

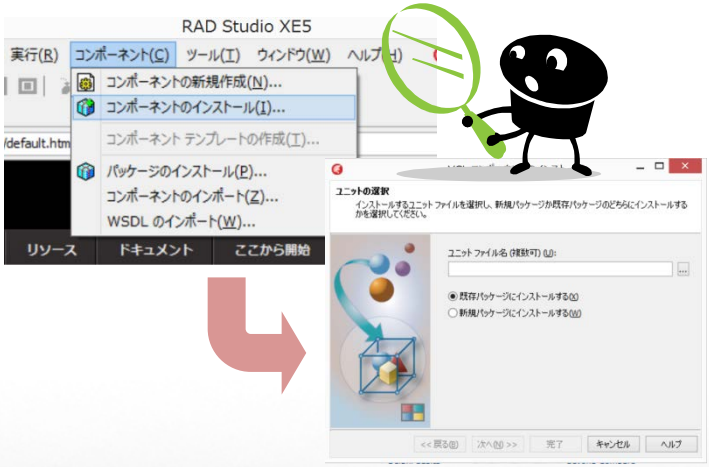
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

XE5（VCL）への移行

① プロジェクトに必要なコンポーネントを登録します

サードパーティ製のコンポーネントや、ユーザが作成したコンポーネントを登録します。
現行のアプリケーションで使用しているサードパーティ製コンポーネントを登録します。この際、ベンダから最新版（XE5対応）が提供されている場合は最新版にアップグレードしておきます。
ユーザコンポーネントについては、コンポーネントのインストールから、用意してあるソースを登録します。



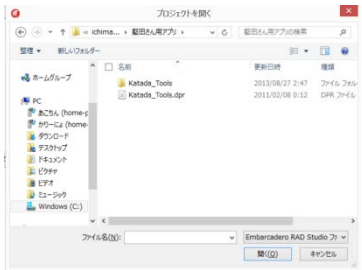


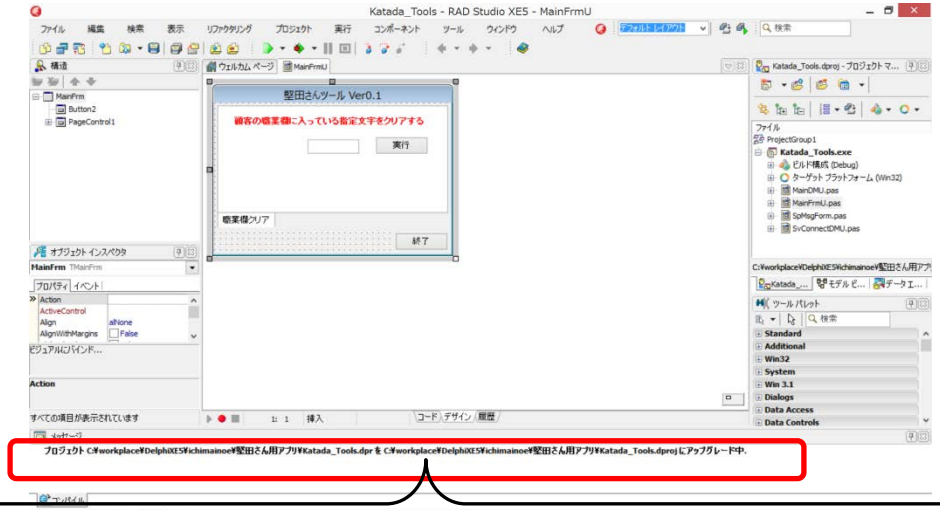
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

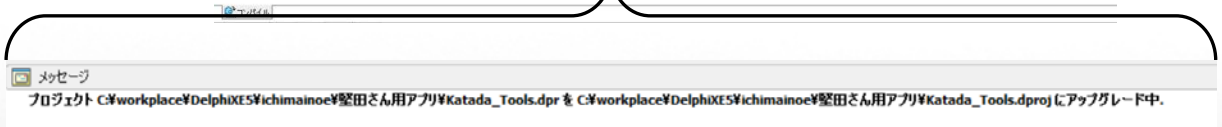
XE5（VCL）への移行

② 移行するアプリケーションのプロジェクトをDelphi XE5で開きます

開く際、Delphi XE5がプロジェクトの自動変換を行ってくれます。
この段階で、アプリケーションの各ソースが表示できるか確認します。
この時、すべてのフォームについて、表示するようにしてください。
※コンポーネントが不足している場合には、フォームが開けなくなります。





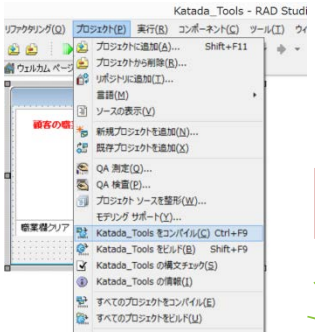


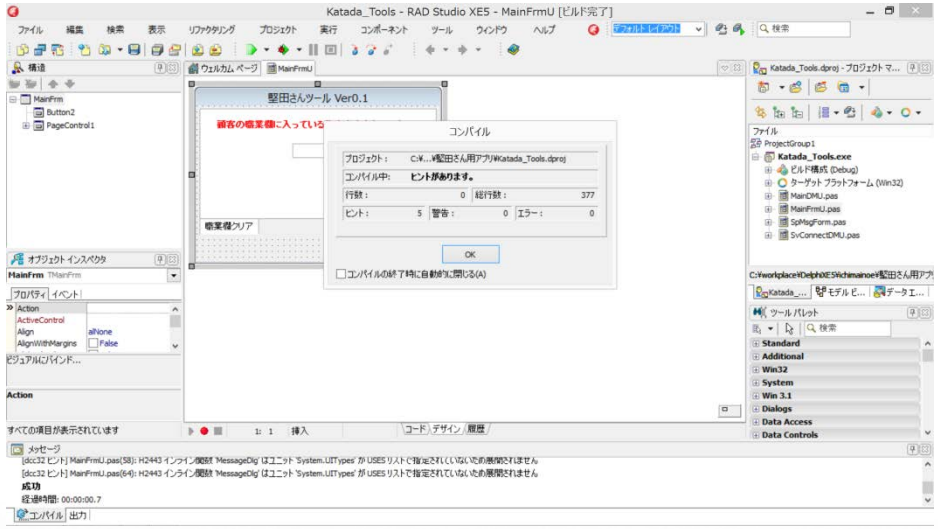
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

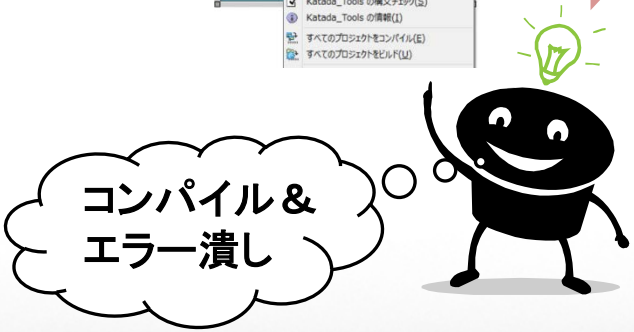
XE5（VCL）への移行

③ コンパイルします

エラーが表示された場合は、エラー箇所を修正します。
ほぼ修正無にコンパイルすることができますが、一部プログラムを修正する必要がある場合があります。
特にコンポーネントの登録忘れなどが多いです。





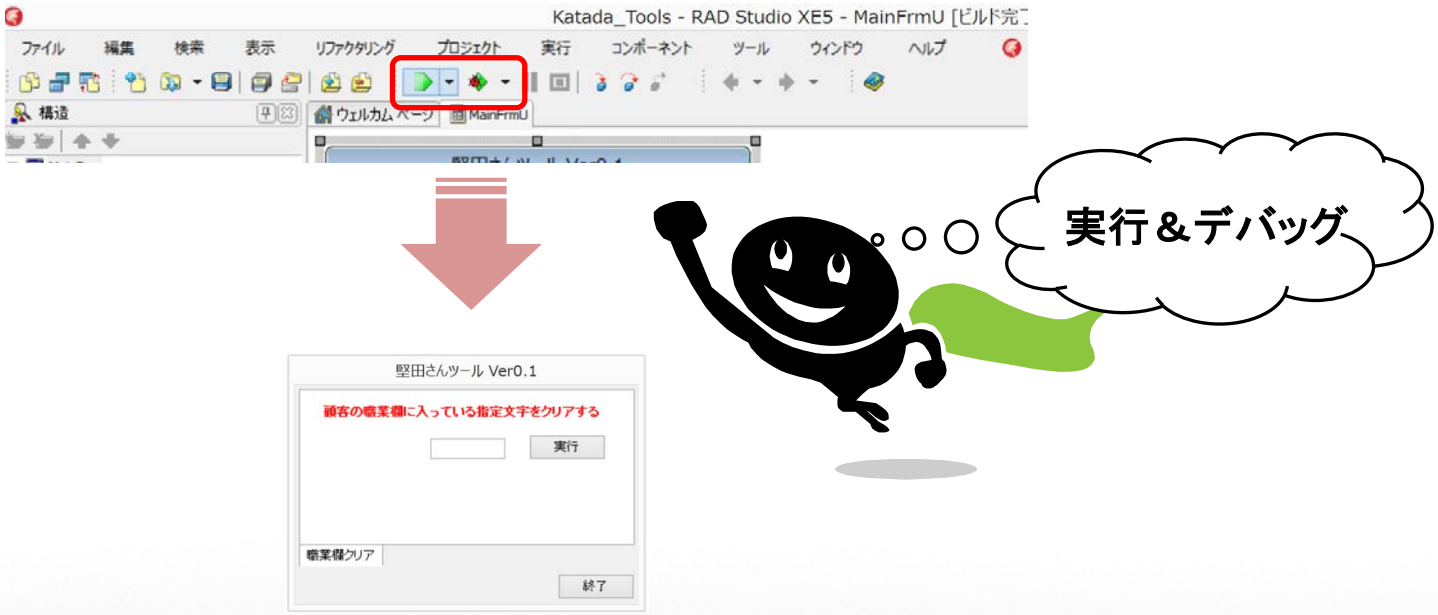


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

XE5（VCL）への移行

④ 動作確認テストを行います

コンパイルしたアプリケーションを実行して、各機能のテストを行います。
各画面を一通り表示するようなテストを行うようにしてください。



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。

XE5（VCL）への移行

前述の手順にて、Delphi XE5（VCL）へのマイグレーションは完了します。

ただし、データベースアクセスはそのままですので、BDEを使用しているアプリケーションでは、他のデータベースアクセスを使用する必要があります。
※BDEはメンテナンス用として提供されているものであり、そのまま使用することは推奨されておりません。（使用し続ける場合は、個人の責任において行ってください）

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。

5

データベースアクセスの移行

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。

データベースアクセスの移行

現在のアプリケーションが、dbExpressやIBExpressなどBDE以外を使用している場合は、Delphi XE5でもそのまま使用することができます。
しかし、BDEを使用してデータベースにアクセスしている場合には、他のデータアクセスコンポーネントへ移行する必要があります。

BDEは、既にサポート・開発が中止され、今後ベンダーによるアップグレードが期待できないからです。また、既にいくつかの問題も抱えています。

- 4Gを超えたHDDの空き容量が正しく認識されない
- マルチコアプロセッサ搭載PC上で動作に不安がある
- WindowsVISTA移行のOS機能に未対応
- 対応しているデータベースのバージョンが古い
- Unicodeに対応していない

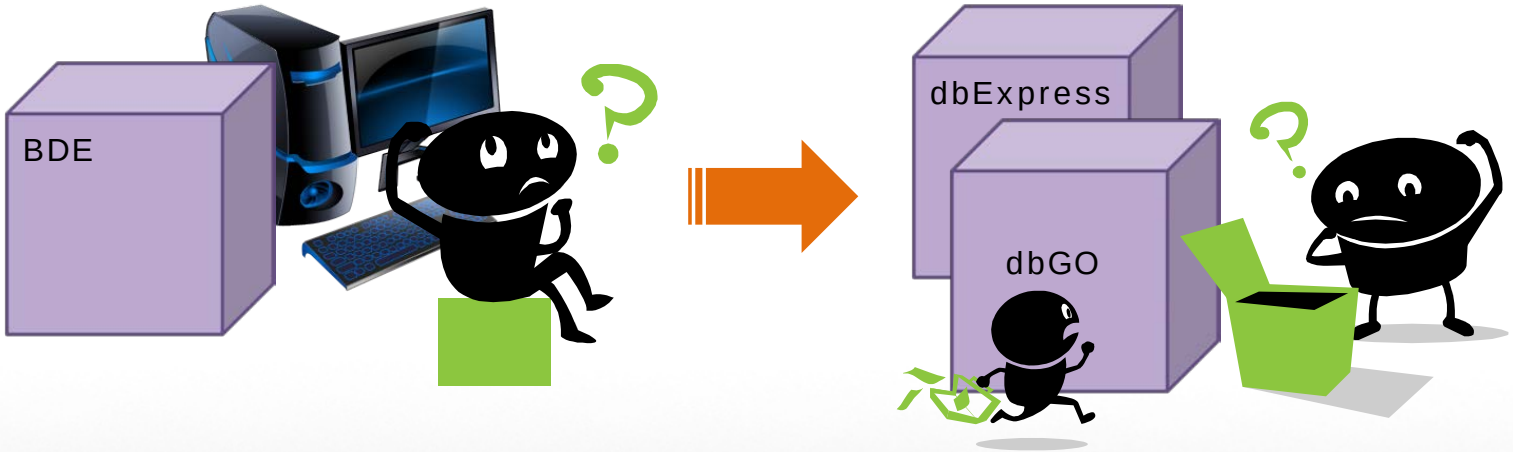
・・・等々

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。

データベースアクセスの移行

まずは、BDEをdbExpressやdbGO、ParadoxをInterBaseに置き換えることを考えます。今までに、これらの移行を考えた方も多いと思います。

しかし、データアクセスアーキテクチャが根本的に異なるため、非互換でありメソッドやプロパティが違います。そのため、そのままでは移行することができず、移行するには、BDEでアクセスしている部分のプログラムをすべて変更する必要があります。また、アーキテクチャが異なるため、パフォーマンスが発揮できない可能性もあります。



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。

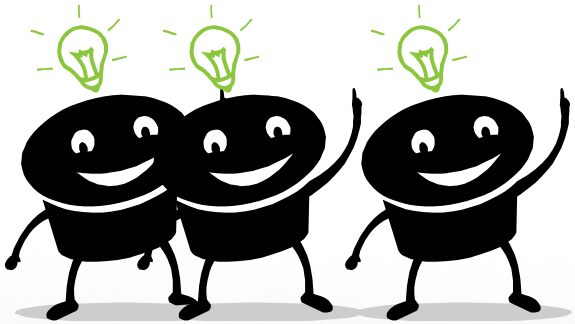
データベースアクセスの移行

前述以外の選択肢として、Delphi XE5に搭載されているFireDACコンポーネントを使用する方法があります。

■ FireDACを使用するメリットとして

- データアクセスアーキテクチャが類似している
- BDEのDataSetと互換性を備えたDataSet
  ※完全互換ではないため、プロパティアクセスやイベント部分で修正が必要
- 専用のデータベースドライバが不要で、インストーラが不要になる
- BDEと同等以上のパフォーマンスが出せる

・・・等々

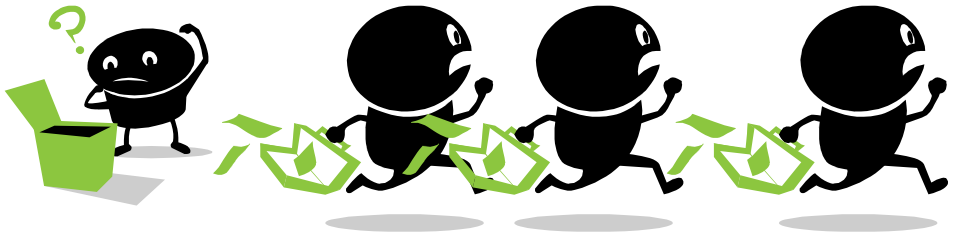


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

データベースアクセスの移行

- BDEからFireDACへ移行する場合のデメリットとして
  - BDEのAPIを直接呼び出しているプログラムは、動作しない
  - BDEとFireDACでは、異種クエリの構文や機能が異なる
  - 互換性のないプロパティは完全に削除しなければいけない
  - Windows7や8にBDEをインストールできるインストーラが必要
  - BDEでParadoxを使用している場合は、FireDAC以外のDBを用意する必要がある
    ※FireDACは、ローカルDBをサポートしていない

・・・等々



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。

embarcadero Developer Camp

データベースアクセスの移行

BDEとFireDACでは、以下のような対応が成り立ちます。
一部対応するコンポーネントが存在しても、挙動が異なるコンポーネントがあります。

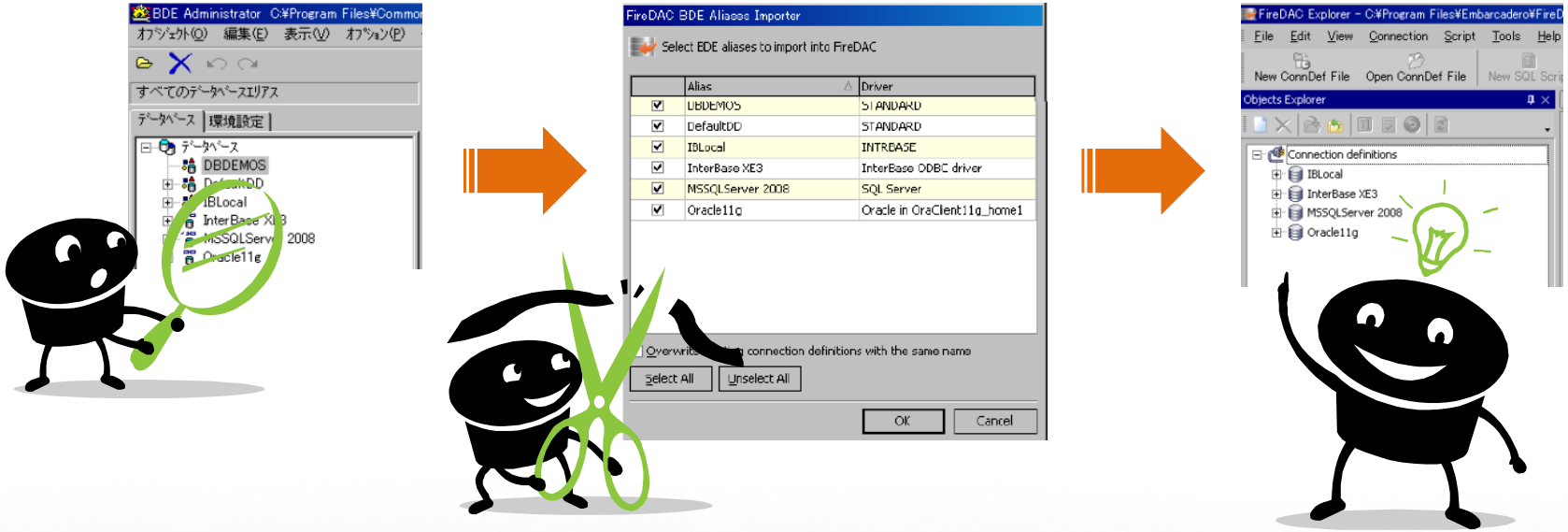
BDE	FireDAC
TDatabase	TADConnection
TSession	TADManager
TTable	TADTable
TQuery	TADQuery
TStoredProc	TADStoredProc
TUpdateSQL	TADUpdateSQL
TBatchMove	TDataMove
・・・等々	

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。
embarcadero Developer Camp

データベースアクセスの移行

FireDACでは、BDEのエイリアス設定をインポートすることが可能です。
ただし、Paradoxのエイリアスはインポートできません。
※FireDACでは、ローカルDBがサポートされていないためです。

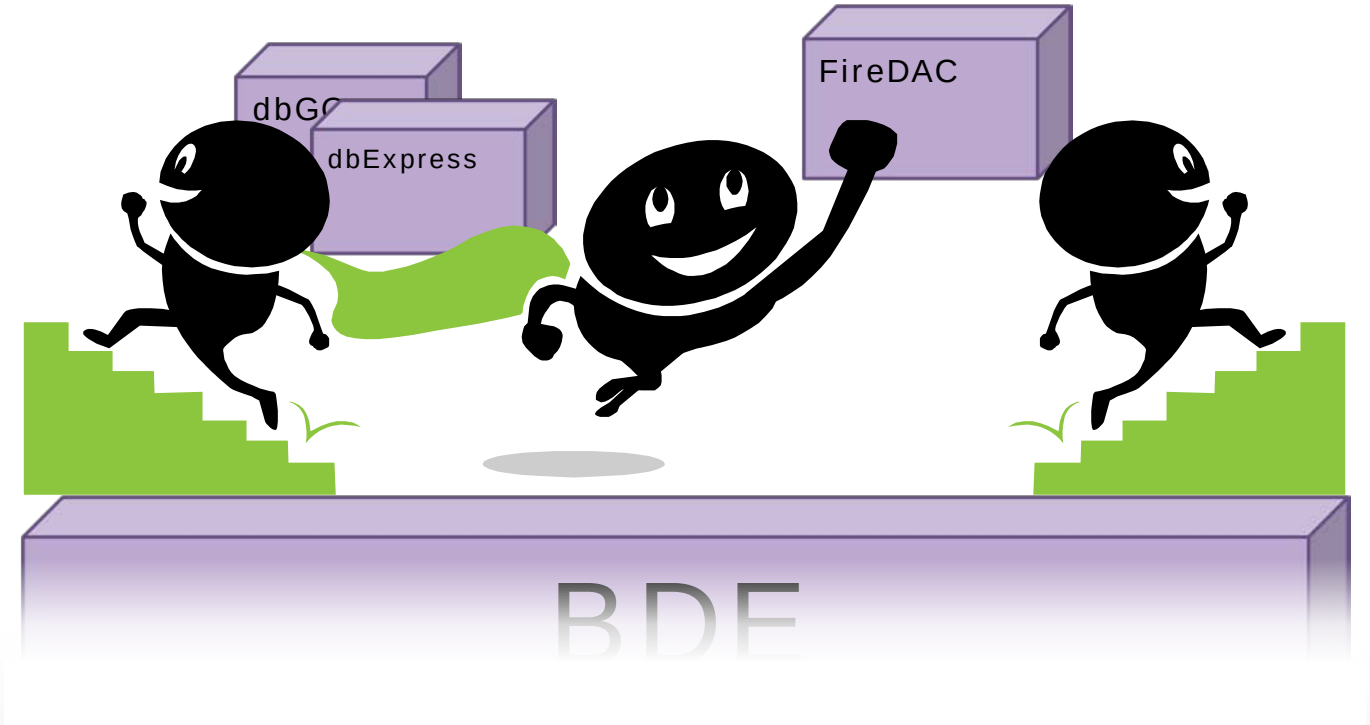
また、インポート後にエイリアスの各パラメータを編集する必要があります。



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。

データベースアクセスの移行

BDEを他のデータアクセスコンポーネントに移行するには、かなりの時間と労力が必要となってきます。
ただ、BDEは今後同梱される保証はありませんので、アプリケーション全体をマイグレーションする機会に合わせて、BDEから飛び立ちましょう！



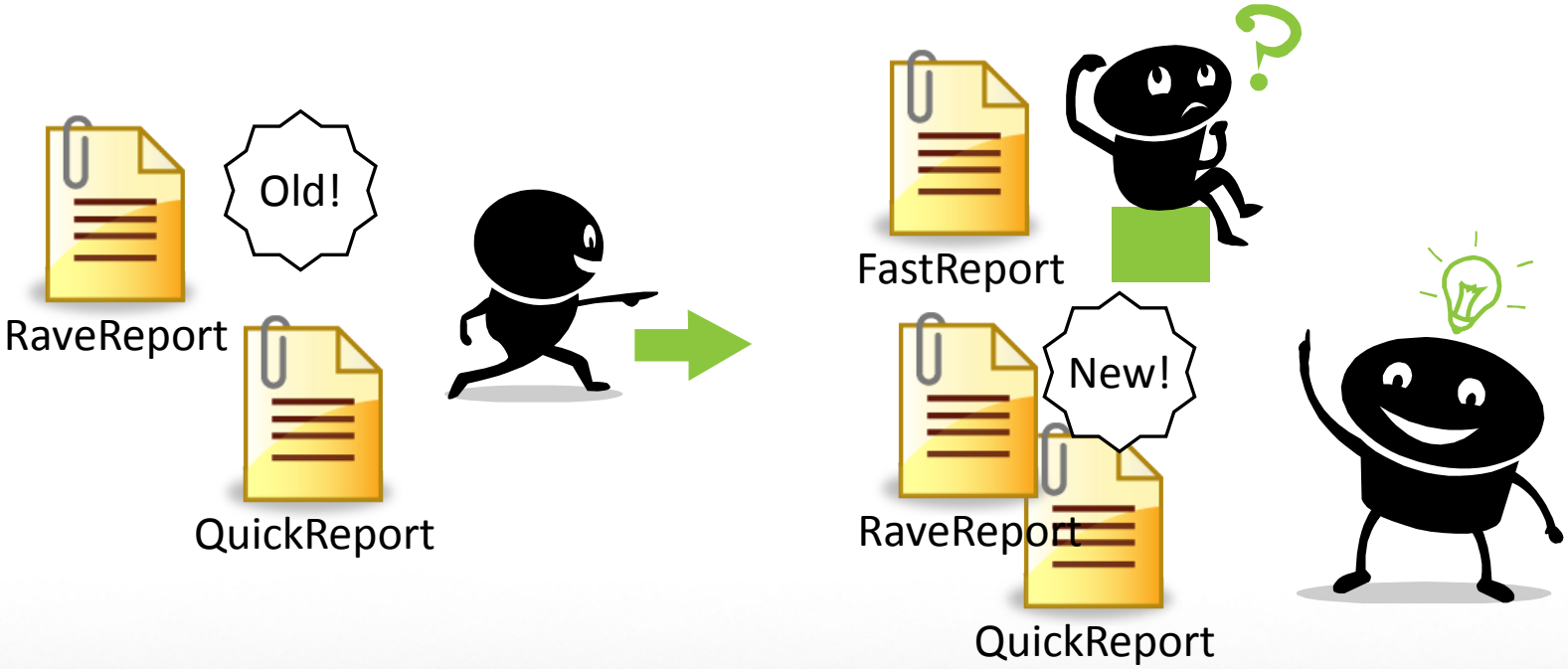
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

帳票の移行

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。

帳票の移行

Delphi 7以前のアプリケーションでは、Delphi付属の帳票コンポーネント「QuickReport」「RaveReport」を使用して、帳票を作成されている場合が多く見受けられます。
これらの帳票をDelphi XE5に付属するFastReportに移行する場合と、QuickReport・RaveReportの最新版に移行する場合の手順を考えます。



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。

帳票の移行

● QuickReport、RaveReportからFastReportへのコンバート

QuickReportやRaveReportで作成された帳票レイアウトは、そのままでは移行できず、コンバートする必要があります。
このコンバートには次の手順を行う必要があります。

- 既存のDelphi開発環境（アプリケーションが作成された環境）にFastReportコンポーネントをインストールします
  ※同一環境上にFastReportを使用する環境が複数ある場合は動作上の問題が発生する可能性があります
- コンバートプログラムを作成し、プログラムを実行します
- 保存された帳票レイアウトファイルをマイグレーションするアプリケーションに組み込みます
- 帳票を操作するプログラムを書き換えます
  ※プログラムは手動コンバートであるため、帳票コンポーネントを操作しているプログラムはすべてFastReport用のプログラムに書き換える必要があります。

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。

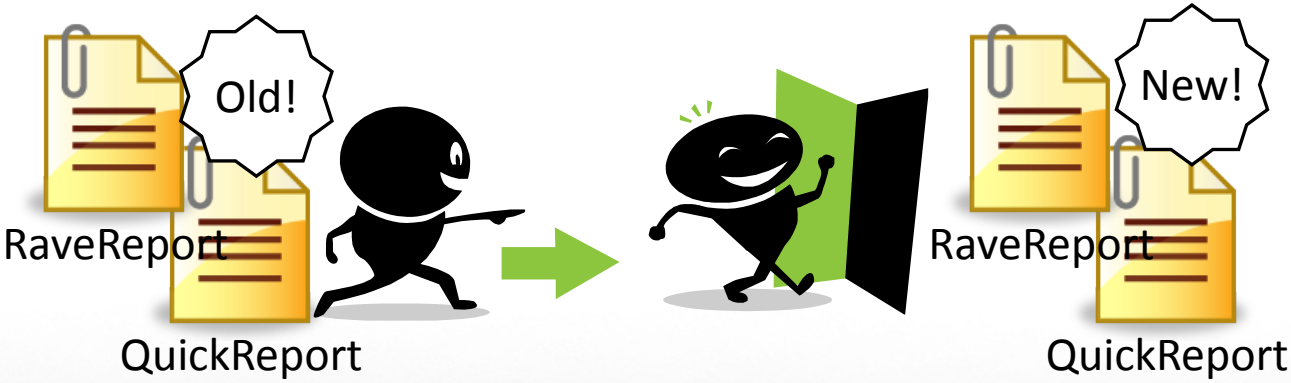
帳票の移行

● QuickReport、RaveReportの最新版へ

Delphi XE5に付属しているFastReportsへ移行せず、QuickReportやRaveReportの最新版を入手して、アプリケーションの帳票をそのまま使用する方法もあります。

この方法では、プログラムをほとんど改修せずに帳票を移行することができますが、有償ですのでマイグレーションを行う際は、熟慮が必要です。

現時点において、QuickReportはDelphi XE5に対応したバージョンがありますが、RaveReportはXE4対応版までしか確認できていません。



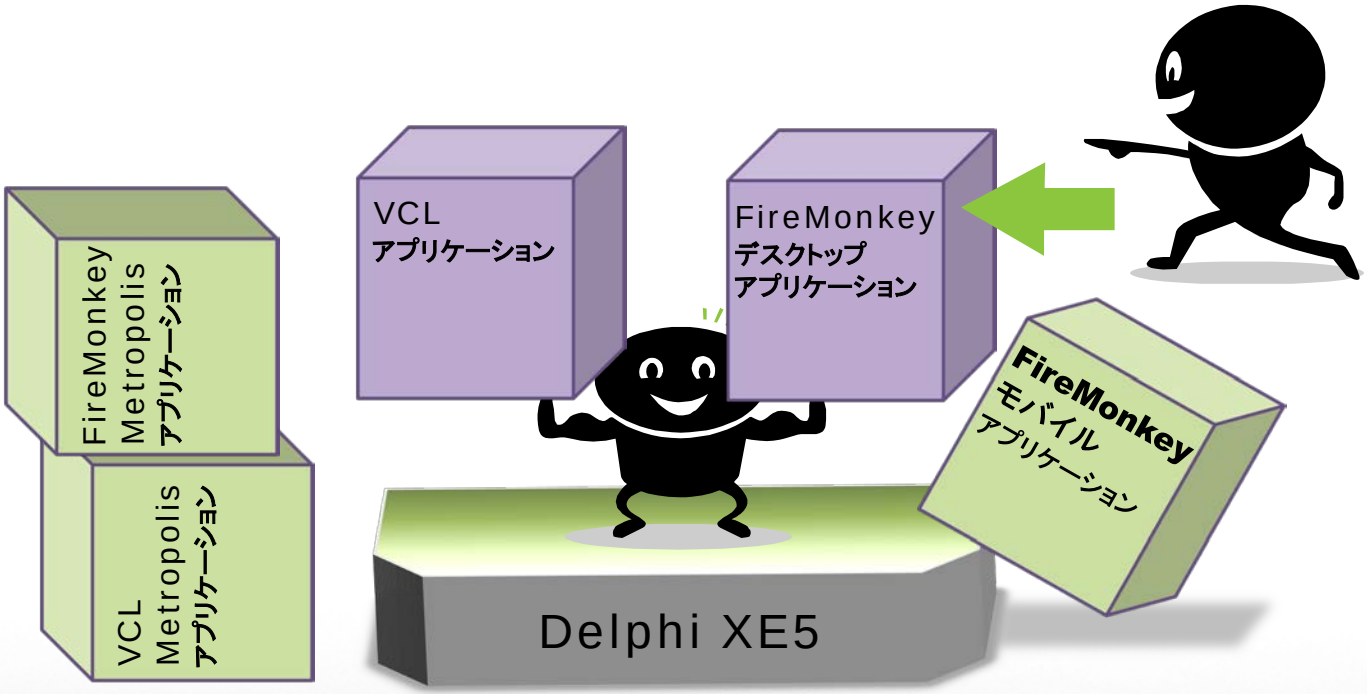
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

クロスプラットフォーム対応への移行

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。

クロスプラットフォーム対応への移行

先に述べたように、コンポーネントライブラリにはVCLと、FireMonkeyが存在します。クロスプラットフォーム対応するには、FireMonkeyを使用してアプリケーションを作成する必要がありますので、既存アプリケーションをFireMonkeyアプリケーションへ移行する方法を考えます。

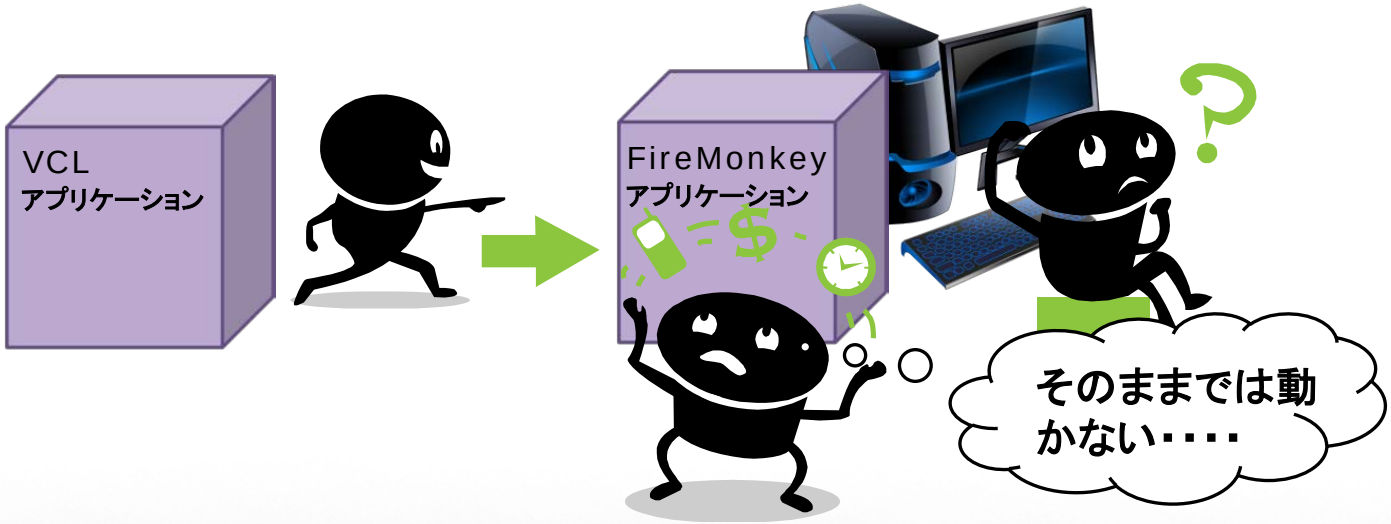


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

クロスプラットフォーム対応への移行

通常のVCLアプリケーションをFireMonkeyアプリケーションへ移行する場合、VCLを使用したコンポーネントやフォームはそのままでは使用できません。
FireMonkeyでは、VCLにあったプロパティが無くなっていたり、異なる名称のプロパティになっていたりします。当然新しいプロパティも存在します。

※FireMonkeyは、別物と考えた方がいいと思います。



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。

クロスプラットフォーム対応への移行

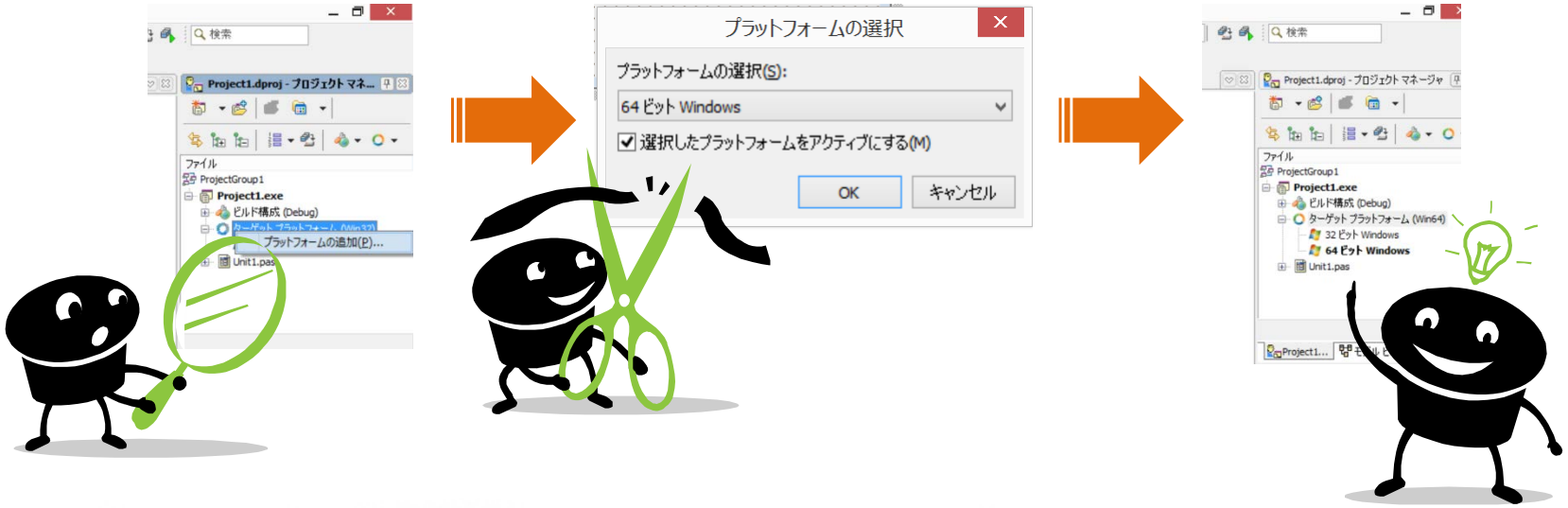
FireMonkeyへVCLのアプリケーションを移行する場合、以下のような手順でアプリケーションを移行します。
※現時点では、直接移行する手段がありません
（サードパーティ製の変換プログラムは考慮していません）

- 新規にFireMonkeyアプリケーションを作成する
- 既存アプリケーションを修正する
  既存のVCLアプリケーションプロジェクト上で、FireMonkeyに対応できるよう修正しておきます
- 既存アプリケーションのプロジェクトからコピー
  修正を行った既存プロジェクトから、FireMonkeyプロジェクト上にコピー＆貼り付けを行います。
- 動作テストを行う

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。

クロスプラットフォーム対応への移行

64ビットWindowsへ対応する場合は、プロジェクトのターゲットプラットフォームに「64ビットWindows」を追加し選択することで、簡単に対応することができます。
ただし、開発環境が32ビットOSの場合では、64ビットWindowsの環境が別に必要になります。

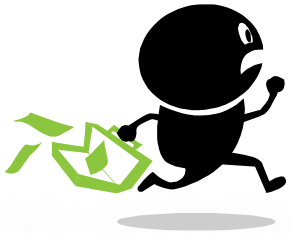


本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

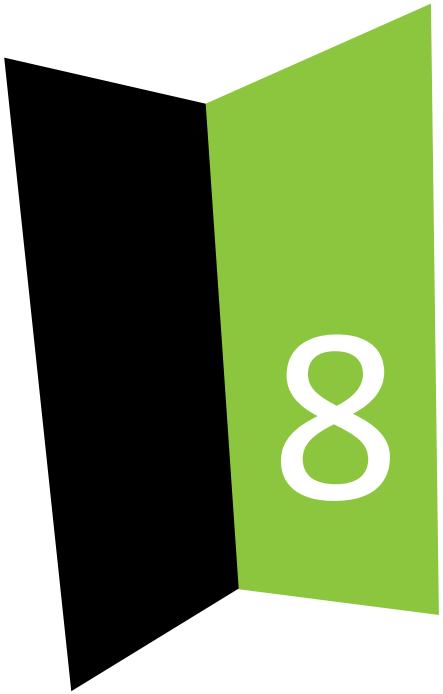
クロスプラットフォーム対応への移行

FireMonkeyに移行する手間を考えると、新規アプリケーションとして作成したほうがきれいに作成することができるかもしれません。
ただ、コンポーネントを使用しないクラスや関数などは、ある程度そのままで使用できますので、画面系だけ新規に作成するのが、最良かと考えます。

また、デスクトップアプリケーションをモバイルアプリケーション対応にする場合、モバイルコンパイラでサポートされていないデータ型などは、すべて変更する必要が出てきます。



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。



まとめ

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。

まとめ

改修するにあたり、新規にアプリケーションを開発するより、Delphi XE5でマイグレーションする方が、期間的・予算的に余裕をもって取り組むことができます。

また、マイグレーションを機にアプリケーションをクロスプラットフォーム対応化することも可能ですので、Delphi XE5を開発ツールとしてご検討して頂ければ幸いです。
Delphiのマイグレーションをお考えの方は、是非「株式会社フルネス」までご相談ください。

以上、ご清聴有難うございました。

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。