

【セッションNo. 3】

## 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 8.0

株式会社ミガロ.  
RAD事業部 技術支援課  
吉原 泰介

## 【アジェンダ】 -サポートからの問合せフィードバック-

Q1. ExcelファイルをPDFに変換するには？

Q2. EXEのサイズを小さくするには？

Q3. C/Sアプリで3階層接続を行うには？

## ■ Q1. ExcelファイルをPDFに変換するには？

### ❗【質問】

Acrobat等のツールを使わずに、Delphi/400でExcelファイルをPDFに変換することはできますか？

### ✉【回答】

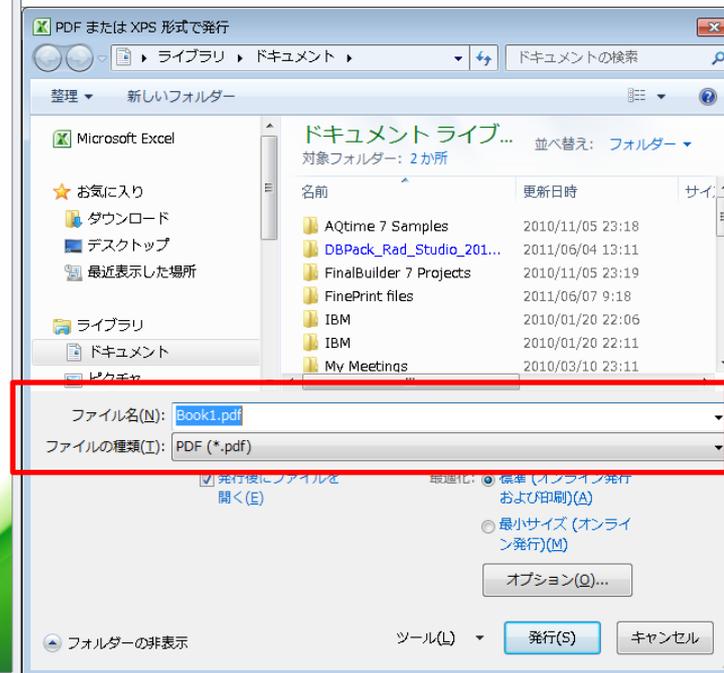
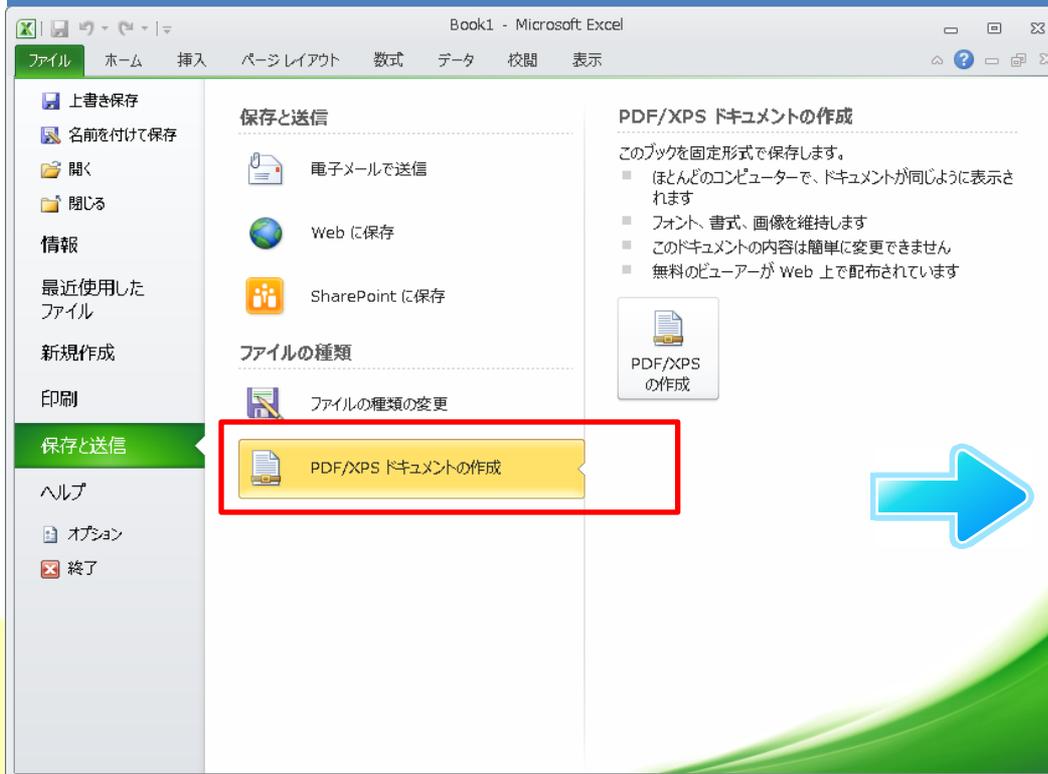
Excel2007以降では、実はExcel自身にPDFエクスポートの機能が実装されています。

Delphi/400から、この機能をOLEで利用することでAcrobat等のツールを使わなくても、ExcelファイルをPDFに変換することができます。

## ■ Q1. ExcelファイルをPDFに変換するには？

- Excel2007以降に提供されているPDF変換機能

### Excel2010



## ■ Q1. ExcelファイルをPDFに変換するには？

- ExcelをPDF変換するサンプルプログラム

ExcelをPDF変換する方法

変換元Excelファイル名  
C:\Users\yoshiwara\Desktop\エクセルファイル.xls

変換先PDFファイル名  
C:\Users\yoshiwara\Desktop\エクセルファイル.pdf

ダイアログで  
ファイルを設定

Excelの機能を  
利用してPDF  
に変換

変換

エクセル.xls	2011/06/15 21:08
エクセル.pdf	2011/06/15 21:09

## ■ Q1. ExcelファイルをPDFに変換するには？

- OLEからExportAsFixedFormat関数でPDF機能を利用する

### 呼出パラメータ

- ① **XlFixedFormatType** (XlFixedFormatType)  
PDF 形式または XPS 形式のどちらで保存するかを指定する。
- ② **Object Filename** (OleVariant)  
保存ファイルの完全パス。省略した場合、現在のフォルダーに保存される。
- ③ **Object Quality** (OleVariant)  
エクスポートされるファイルの品質を指定する。
- ④ **Object IncludeDocProperties** (OleVariant)  
文書のプロパティを含める場合は true。それ以外は false。

## ■ Q1. ExcelファイルをPDFに変換するには？

### 呼出パラメータ

- ⑤ **Object IgnorePrintAreas (OleVariant)**  
印刷領域を無視する場合は true。それ以外は false。
- ⑥ **Object From (OleVariant)**  
エクスポートを開始するページ番号。
- ⑦ **Object To (OleVariant)**  
エクスポートする最終ページ番号。
- ⑧ **Object OpenAfterPublish (OleVariant)**  
直ちにビューアーで表示する場合は true。それ以外は false。
- ⑨ **Object FixedFormatExtClassPtr (OleVariant)**  
IMsoDocExporter インターフェイスの実装へのポインタ。

## ■ Q1. ExcelファイルをPDFに変換するには？

### 変換ボタンの処理(ソース)

```
procedure TForm1.BitBtn1Click(Sender: TObject);
var
  Excel      : Variant;
  WBook      : Variant;
  Wsheet     : Variant;
  ver        : Currency;
begin
  try
    Excel := CreateOleObject( 'Excel.Application' );
    Excel.Application.Visible := True;           //Excelを表示
    Ver:=StrToFloatdef(Excel.Application.Version, 0.0); //Excelバージョンの取得
    if Ver<12.0 then                               //Excel2007以前はエラー
    begin
      ShowMessage( 'Excel2007以降でしか使えません' );
      Excel.Quit;
      Exit;
    end;
  end;
```

## ■ Q1. ExcelファイルをPDFに変換するには？

### 変換ボタンの処理(ソース)

```
WBook:=Excel.Application.WorkBooks;
WBook.Open(edtxlsFile.Text); //画面で設定したExcelを読み込む
WSheet:=Excel.Application.ActiveSheet;
WSheet.ExportAsFixedFormat( //Excel → PDF 変換処理
    $00000000, // Type : PDF $00000000 XPS $00000001
    edtPDFFile.Text, // Filename : 変換するファイル名
    $00000000, // Quality : 通常$00000000 最小$00000000
    True, // IncludeDocProperties: ドキュメント有効
    True, // IgnorePrintAreas : 印刷範囲有効
    EmptyParam, // From : 開始ページ
    EmptyParam, // To : 終了ページ
    True, // OpenAfterPublish : 作成後表示
    EmptyParam // FixedFormatExtClassPtr: ポインタ
);
finally
    WBook.Close; //Book終了
    Excel.Quit; //Excel終了
end;
end;
```

既存のExcel処理も  
簡単にPDF出力を機能  
拡張できます！



エクセル.xls

エクセル.pdf

## ■ Q2. EXEのサイズを小さくするには？

### ❗【質問】

アプリケーションのEXEのサイズが大きくなってしまったのですが、サイズを小さくすることはできますか？

### ✉【回答】

Delphi/400のアプリケーションは、基本1つのEXEに全て必要なモジュールが組み込まれているので、配布が簡単なことが利点ですが、サイズも大きくなります。しかし、EXEに組み込まれたランタイムのパッケージを分離するとEXEのサイズを小さくすることができます。

## ■ Q2. EXEのサイズを小さくするには？

- Delphi/400 アプリケーションのEXEサイズ  
Delphi/400はプログラムをコンパイルすると通常、  
全てを含んだ1つのEXEアプリケーションを生成。



クライアントPC



ドキュメントライブラリ

Projects

名前	更新日時	サイズ
 Project1.exe	2011/06/13 20:24	795 KB

たくさんの画面や機能を作りこむと  
サイズが比例して大きくなります。

## ■ Q2. EXEのサイズを小さくするには？

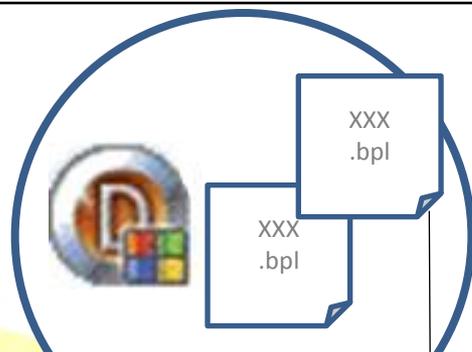
### • Delphi/400 アプリケーションのEXEサイズ

EXEの中には必要なものが全て含まれていますが、コンポーネント情報などが含まれるパッケージ(.bplファイル)を分離することで、EXEのサイズを小さくすることもできます。

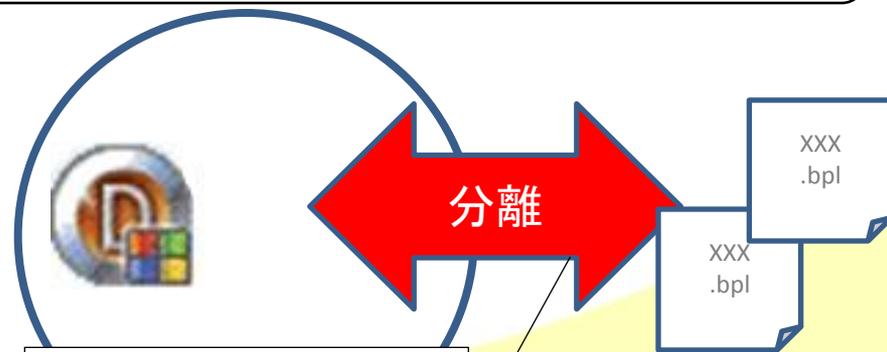


メモ情報

bplファイルとはBorland Package Libraryが略された拡張子のファイルになります。コンポーネントなどの標準で部品化されたプログラム情報が含まれており、Delphi開発PC内にあります。自作することもできますが、基本的にはbplは完成されたものなので、変更されたりはしません。



通常はEXE内にパッケージ情報を含めている



分離するとパッケージ分サイズが小さくなる

## ■ Q2. EXEのサイズを小さくするには？

- EXEからパッケージを分離する  
[プロジェクト|オプション]からパッケージを選択し、  
『実行時パッケージを使って構築』のチェックを入れてコンパイル。

The image shows the Delphi IDE interface. On the left, the 'Project' menu is open, and the 'Options...' item is highlighted with a red box. A blue arrow points from this menu item to the 'Project1.exe のプロジェクト オプション - Base 構成' dialog box. In the dialog box, the 'パッケージ' (Packages) section is highlighted with a red box, and the '実行時パッケージ(L)' (Runtime Packages) section is also highlighted with a red box. The checkbox '実行時パッケージを使って構築(B)' (Build using runtime packages) is checked. The '実行時パッケージ(L)' section also shows a list of packages: 'vclx;vcl;vclimg;dbrtl;Rave77VCL;bdertl;rtl;vclactnband;vclddb;vcldbx'.

## ■ Q2. EXEのサイズを小さくするには？

- EXEからパッケージを分離する  
パッケージを分離すると必要最低限のEXEサイズになります。  
(パッケージは別途配布が必要です。)

クライアントPC

ドキュメント ライブラリ  
Projects

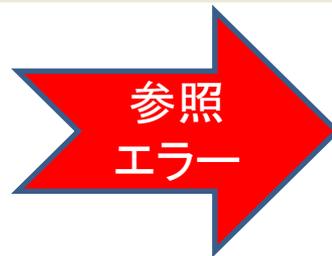
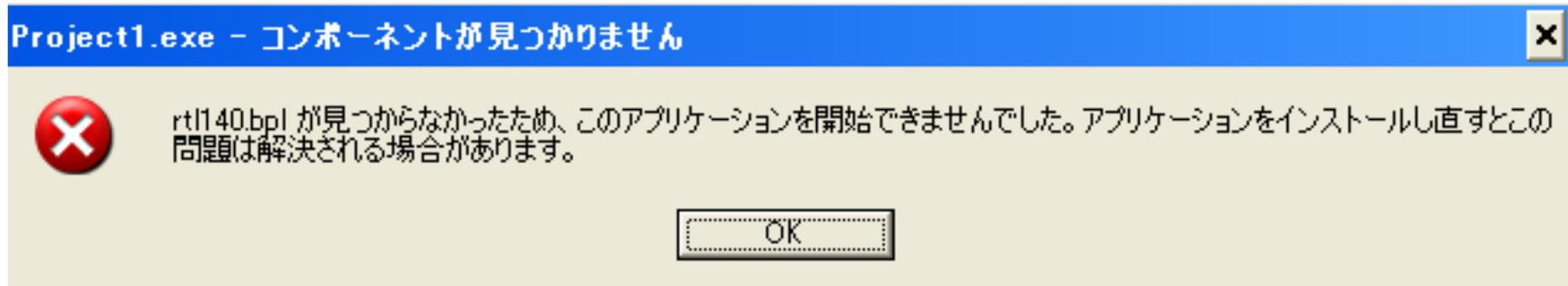
名前	更新日時	サイズ
Project1.exe	2011/06/13 20:27	21 KB

XXX .bpl    XXX .bpl

パッケージ分サイズが大幅に小さくなる

## ■ Q2. EXEのサイズを小さくするには？

- パッケージを分離する場合の注意点  
小さくなったEXEだけを配布しても、実行すると以下のようなエラーになるので注意



EXEに含まなくてもPC内に  
パッケージは必要です。

## ■ Q2. EXEのサイズを小さくするには？

- 配布が必要なパッケージを調べる  
EXEで使用している(配布が必要な)パッケージは  
[プロジェクト|プロジェクト名の情報]で確認

The screenshot shows the 'Project1.dproj の情報' (Project1.dproj Information) dialog box. The '使用するパッケージ' (Used Packages) section is highlighted with a red box and contains the following text:

```
使用するパッケージ  
rtl140.bpl  
vcl140.bpl
```

Below the dialog box, two callout boxes show 'XXX .bpl'. A text box explains that the packages used by the EXE are listed in the project information, and that bpl files are not created or changed by the user, so they are basically OK for distribution after one distribution.

プロジェクトの情報  
すべてのプロジェクトをコンパイル  
すべてのプロジェクトをビルド  
リソース...  
構成マネージャ...  
依存関係...  
オプション... Shift+Ctrl+F11

Project1.dproj の情報

プログラム	
コンパイル対象:	102 行
コード領域:	2596 バイト
データ領域:	72 バイト
初期スタックサイズ:	16384 バイト
ファイルの大きさ:	20992 バイト

状態  
Project1.dproj : コンパイル済み

使用するパッケージ  
rtl140.bpl  
vcl140.bpl

ヘルプ

プロジェクトの情報  
すべてのプロジェクトをコンパイル  
すべてのプロジェクトをビルド  
リソース...  
構成マネージャ...  
依存関係...  
オプション... Shift+Ctrl+F11

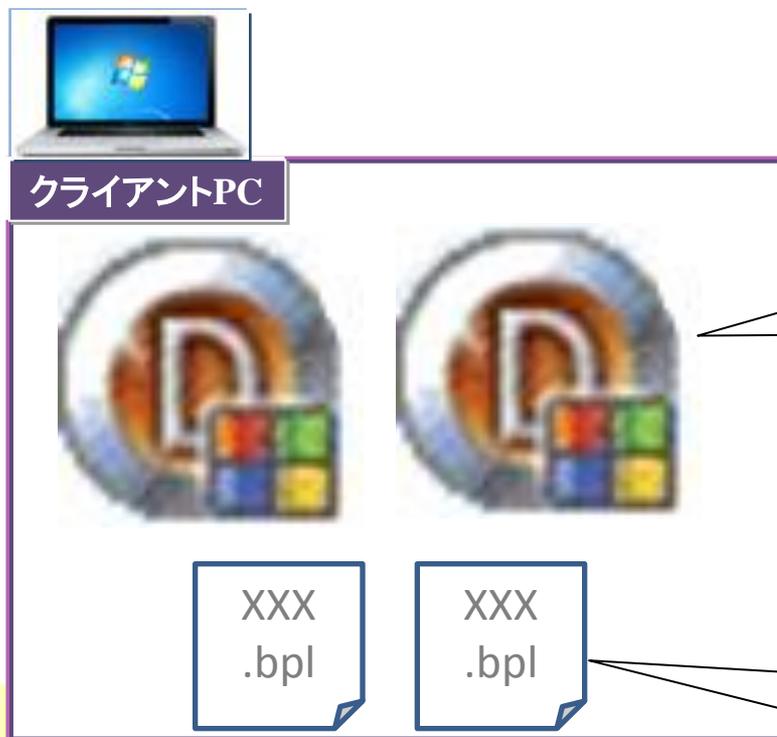
XXX .bpl

XXX .bpl

プロジェクトの情報で、このEXEが使用しているパッケージが表示される。  
bplは自分で作成しないかぎり、変更はないので  
**基本的に1度配布すればOK!**

## ■ Q2. EXEのサイズを小さくするには？

- こんな場合にサイズが小さいことが有効



### EXEの入れ替えが多い場合



サイズが小さければ  
EXEの入れ替え(上書き)  
がスムーズにできる！

### EXEがたくさんある場合



ランタイムのパッケージは  
どのEXEにも組み込まれる  
ので、分離しておく  
とEXE毎に無駄な重複が  
なくなる！

## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### ! 【質問】

C/SアプリケーションでWebのように3階層方式で接続を行う仕組みは開発できますか？

### ✉ 【回答】

Delphi/400にはDataSnapという機能がありますが、これを使用することで、dbExpressでの3階層方式での接続アプリケーション開発も可能です。特にV2010以降は大きく機能追加されており、簡単に仕組みを構築することができます。

## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

- 3階層の接続の仕組みでは、どんな効果があるか？



PC毎の環境構築や管理が手間になっている。



各PCはEXEを配布するだけで環境構築なしで使える！



クライアントアプリケーションは『画面』  
サーバアプリケーションは『機能』  
という作り分けをすれば、機能の変更時に  
サーバだけで対応(モジュール入れ替え)できる！



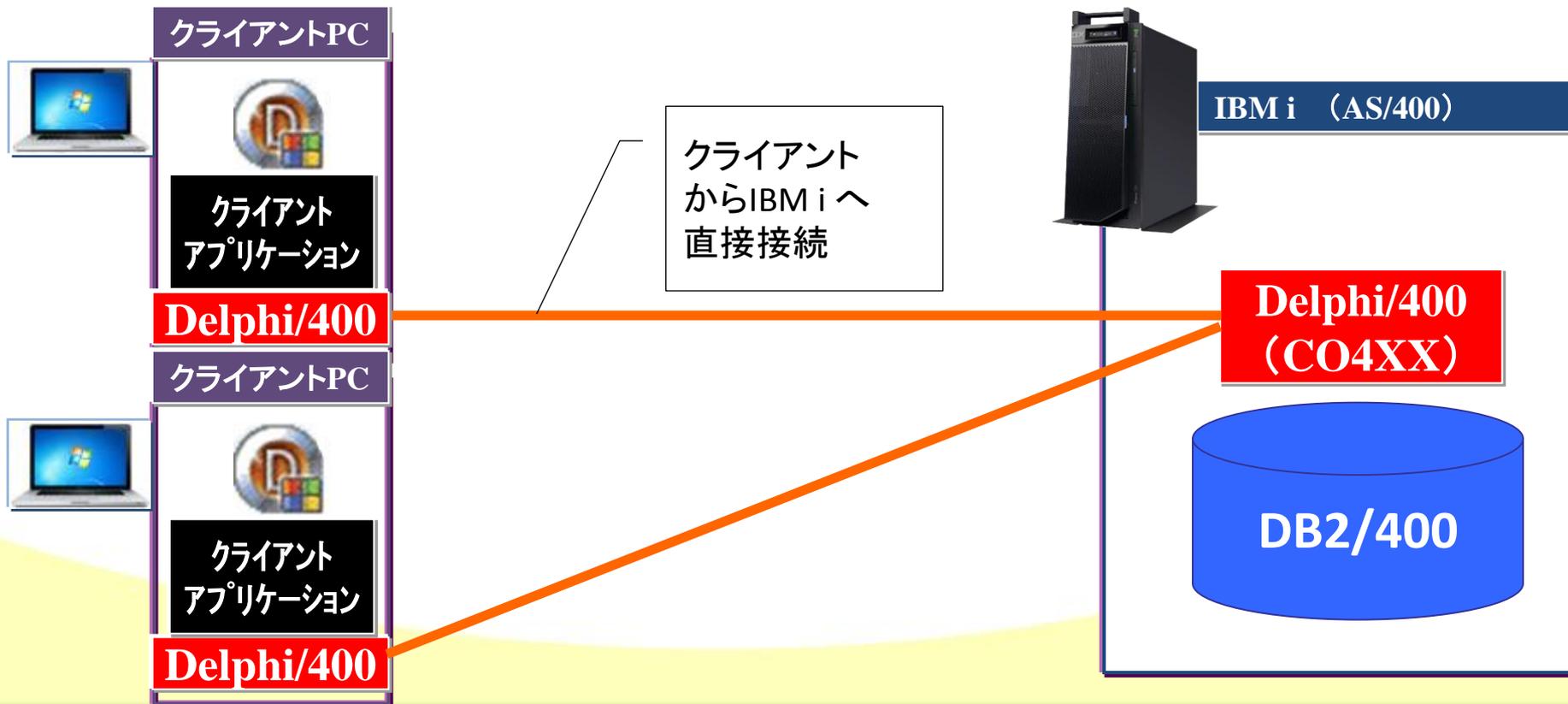
PC上でのクライアントアプリケーションの動作負荷が大きい。



クライアントの負荷をサーバ経由で軽減することができる！

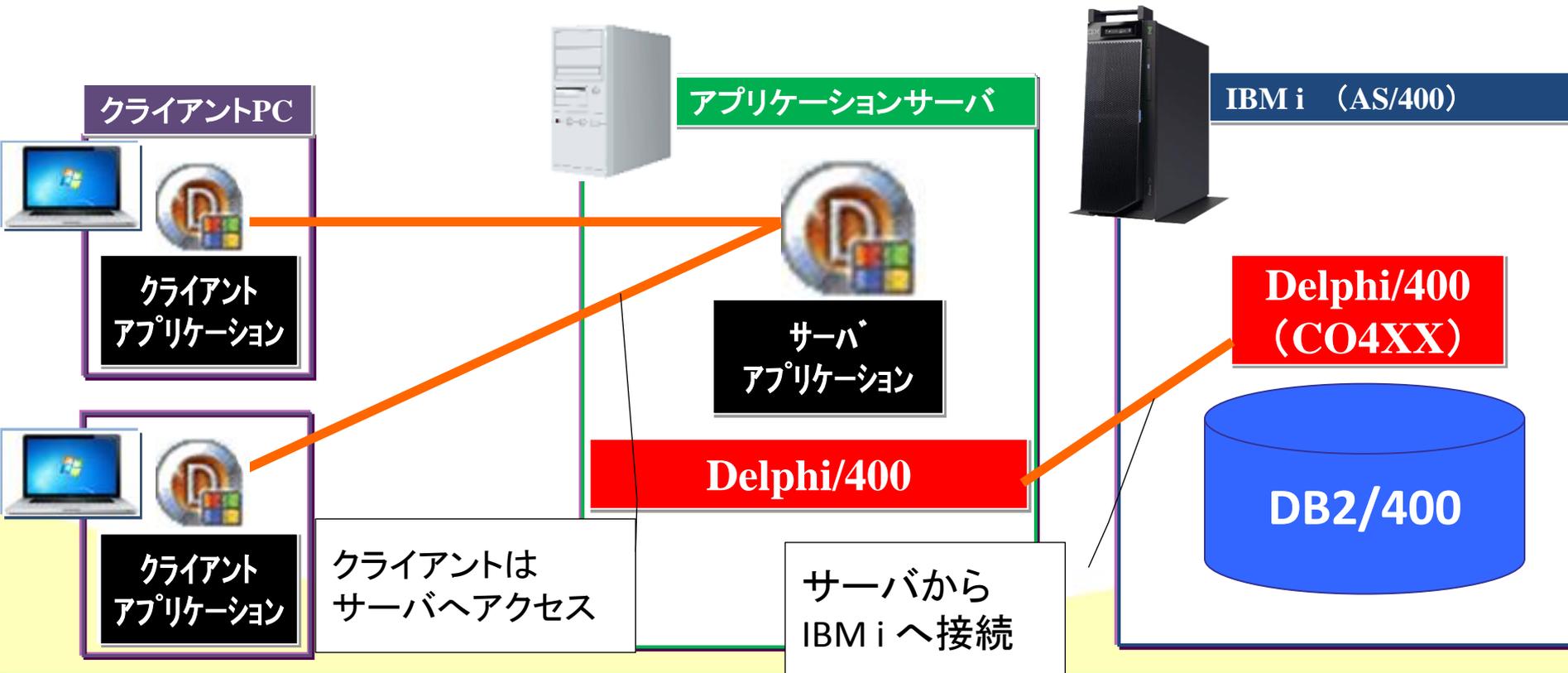
## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

- C/Sアプリケーションでの2階層接続の仕組み



## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

- C/Sアプリケーションでの3階層接続の仕組み  
(アプリケーションの仕組みによって実現する方法)



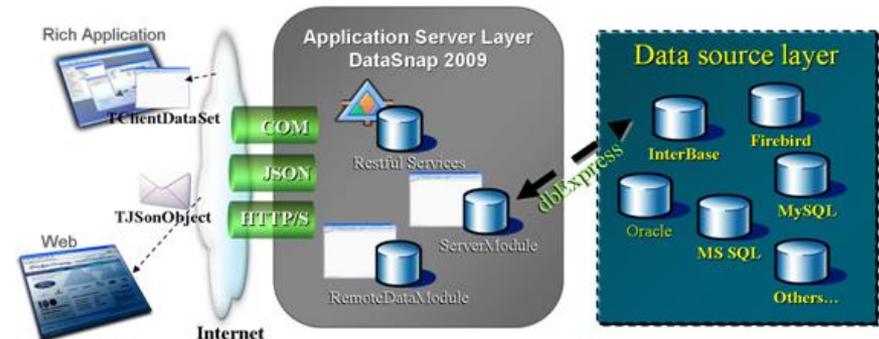
## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### DataSnapとは

- DataSnapは、多層アプリケーション、とりわけ多層データベースアプリケーションの開発を可能にする Delphi/400の技術です。

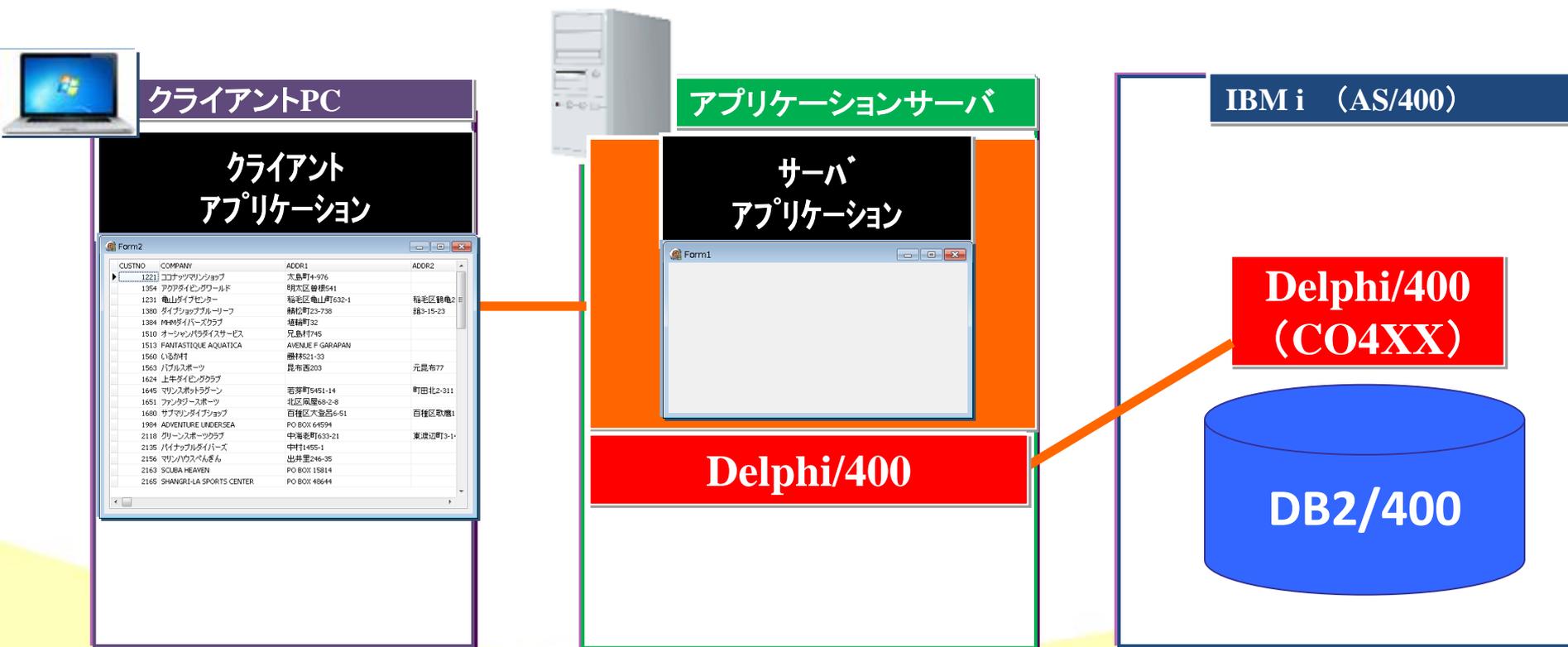
DataSnapはV2009で大幅に機能が再構築され、V2010ではより使いやすく、機能も拡張されました。多層アプリケーションを作成する使いやすいウィザードも用意され、簡単に開発ができるようになりました。

### V2009,V2010と強化されたDataSnap



## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

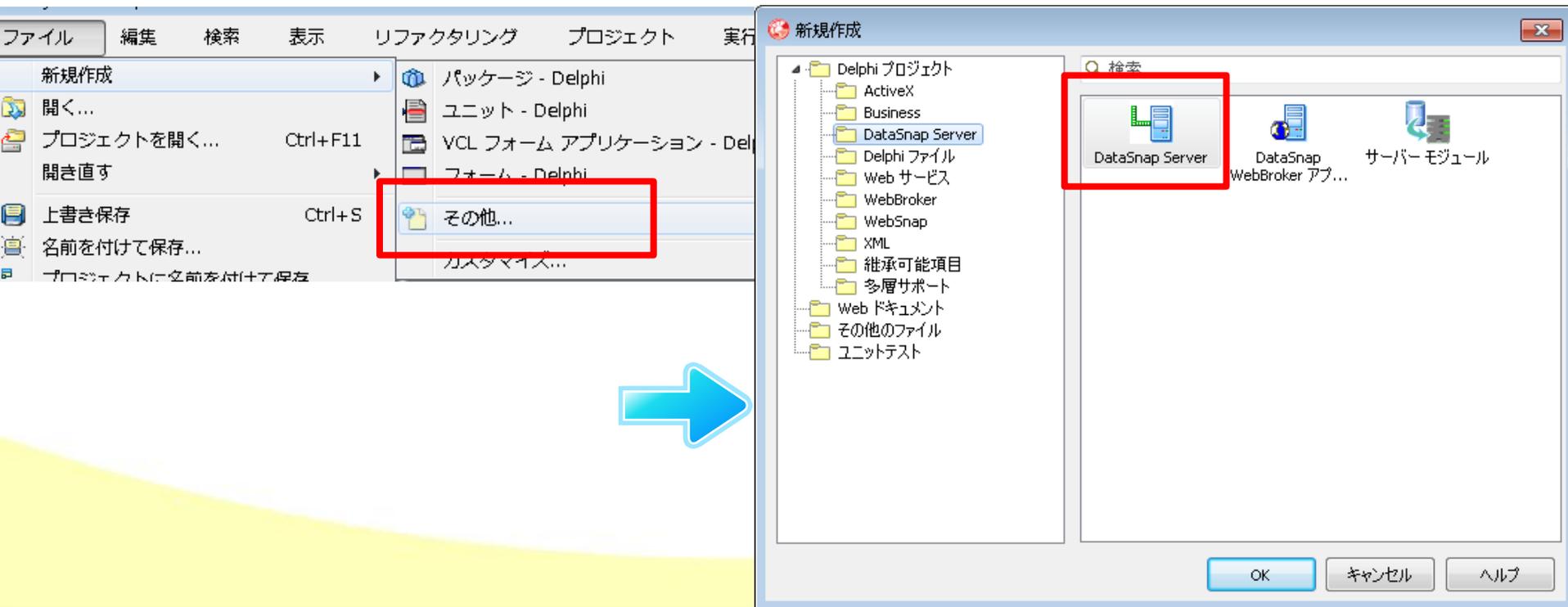
### • サーバアプリケーションの作り方



## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### • サーバアプリケーションの作り方①

メニューの[ファイル|新規作成|その他]から『DataSnapServer』を選択



## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### • サーバアプリケーションの作り方②

DataSnapServerの構成を選択してOK

DataSnap サーバーの新規作成

アプリケーションの種類を次の中から1つ選択できます

VCL フォーム アプリケーション  
 コンソール アプリケーション(C)  
 サービス アプリケーション(S)

通信プロトコルを1つ以上選択できます

TCP/IP(I)  
 HTTP(H)  
 認証

サーバー メソッド クラスを追加(A)

上位クラス: TDSServerModule

サンプル メソッドを含める

OK キャンセル ヘルプ(H)

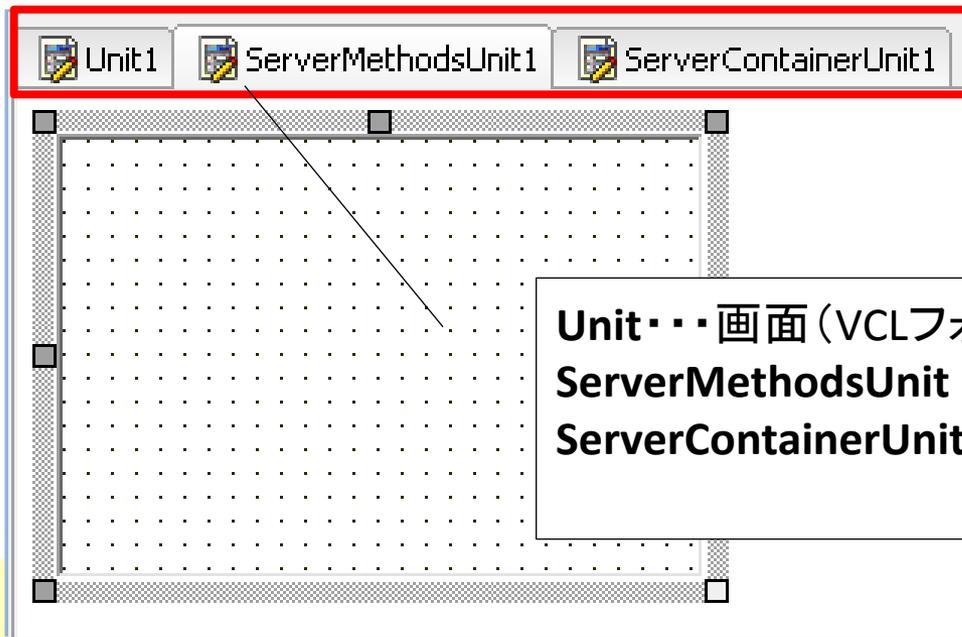
サーバ上でどのように実行するかを選択します。  
サーバに常駐で起動させる場合はサービスアプリケーションを使います。

接続通信の設定  
標準はTCP/IPを使います。  
※HTTP経由での通信も可能です。

## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### • サーバアプリケーションの作り方③

選択した構成によってモジュールが生成



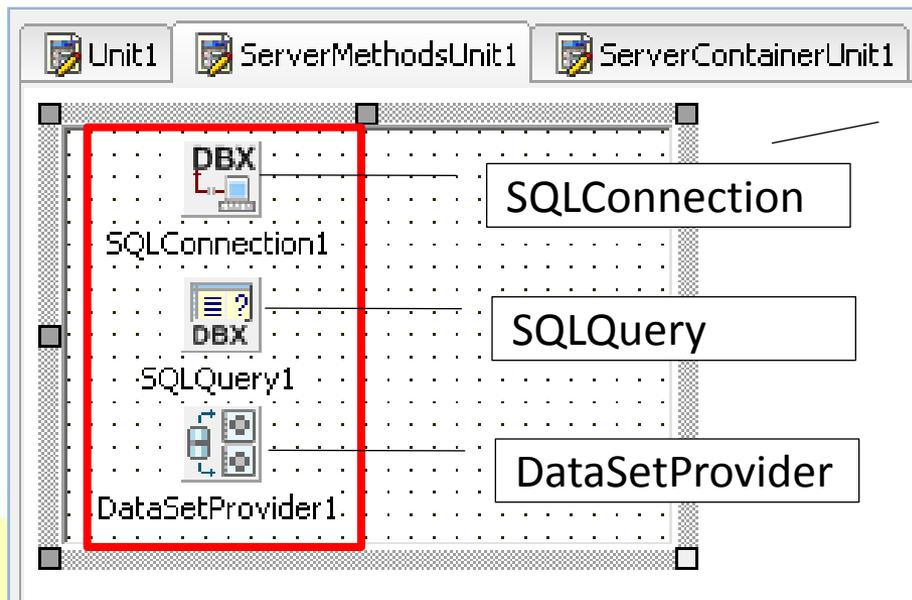
**Unit**・・・画面 (VCLフォームアプリケーション時のみ)  
**ServerMethodsUnit**・・・サーバが提供する機能  
**ServerContainerUnit**・・・通信を制御する機能  
(データモジュール)

## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### • サーバアプリケーションの作り方④

ServerMethodsに次のコンポーネントを配置

SQLConnection、SQLQuery、DataSetProvider



今回はIBM i への接続とSQL発行をサーバアプリケーションの機能としてServerMethodsに実装します

## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### • サーバアプリケーションの作り方⑤

#### SQLConnectionコンポーネントの設定

The screenshot shows the Visual Studio IDE with a project containing several components: Unit1, ServerMethod1, DBX, SQLConnection1, SQLQuery1, and DataSetProvider1. The SQLConnection1 component is highlighted with a red box. A callout box points to the 'ConnectionName' property in the Object Inspector, which is set to 'CO400CONNECTION'. Another callout box points to the 'Params' property, which is set to '(TStrings)'. A blue arrow points from the 'Params' property to the '値リストの編集' (Edit Value List) dialog box. This dialog box shows a table of key-value pairs for the connection string, with several fields highlighted in red boxes and labeled with callouts.

キー	値
ErrorResourcefile	
BlobSize	-1
DriverName	CO400
Database	MIGAROIS
LocaleCode	0000
IsolationLevel	ReadCommitted
Decimal Separator	
Password	D400
User Name	D400
HostName	MIGAROIS
RoleName	CO413

Callouts for the highlighted fields in the '値リストの編集' dialog:

- 接続名 (points to Database)
- ユーザーパスワード (points to Password)
- 接続名 (points to HostName)
- ライブラリ名 (points to RoleName)

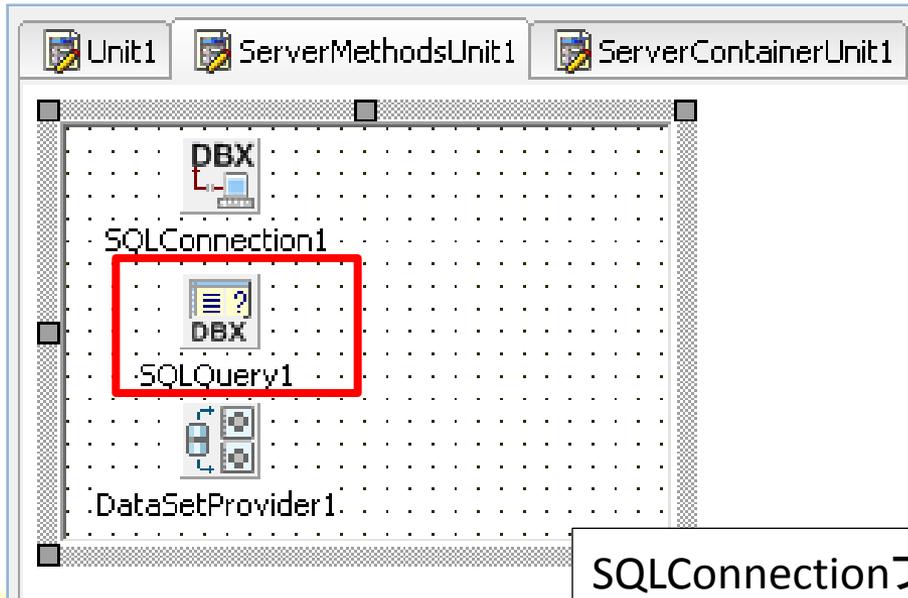
Other callouts:

- ConnectionNameプロパティを“CO400CONNECTION”に設定 (points to ConnectionName in Object Inspector)
- Paramsプロパティを設定 (points to Params in Object Inspector)

## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### • サーバアプリケーションの作り方⑥

#### SQLQueryコンポーネントの設定

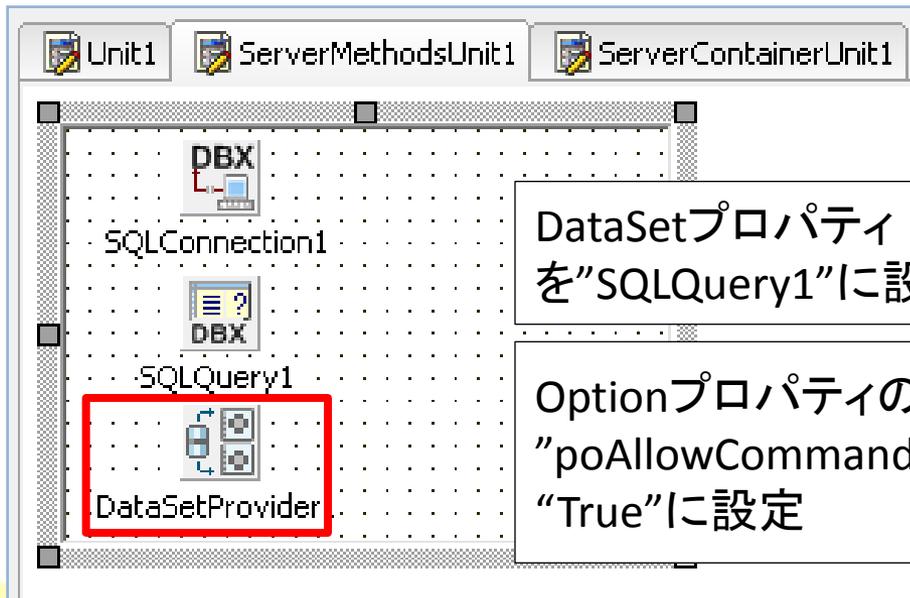


SQLConnectionプロパティ  
を” SQLConnection1”に設定

## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

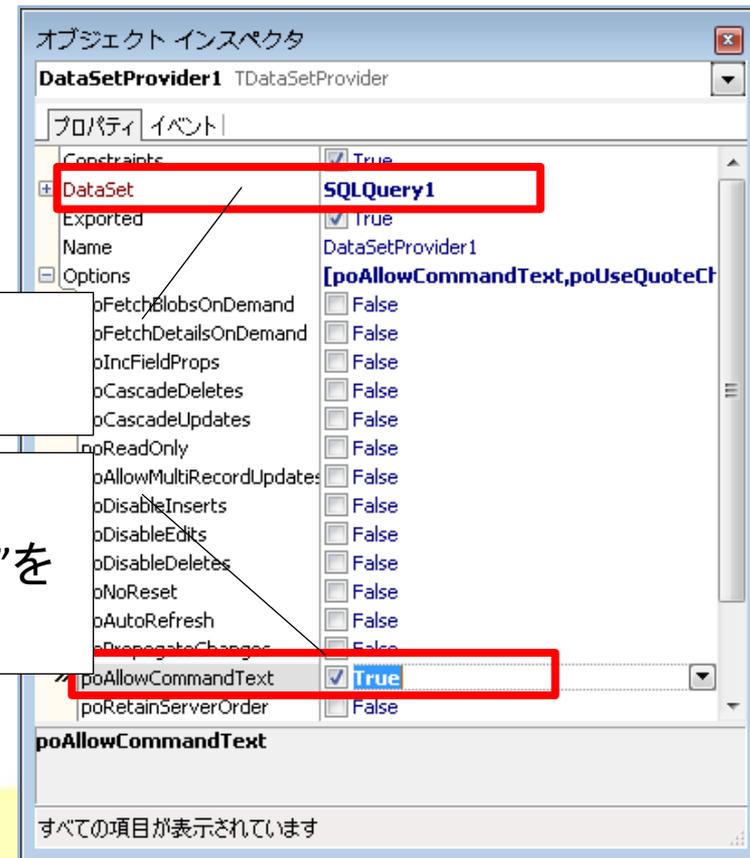
### • サーバアプリケーションの作り方⑦

#### DataSetProviderコンポーネントの設定



DataSetプロパティ  
を"SQLQuery1"に設定

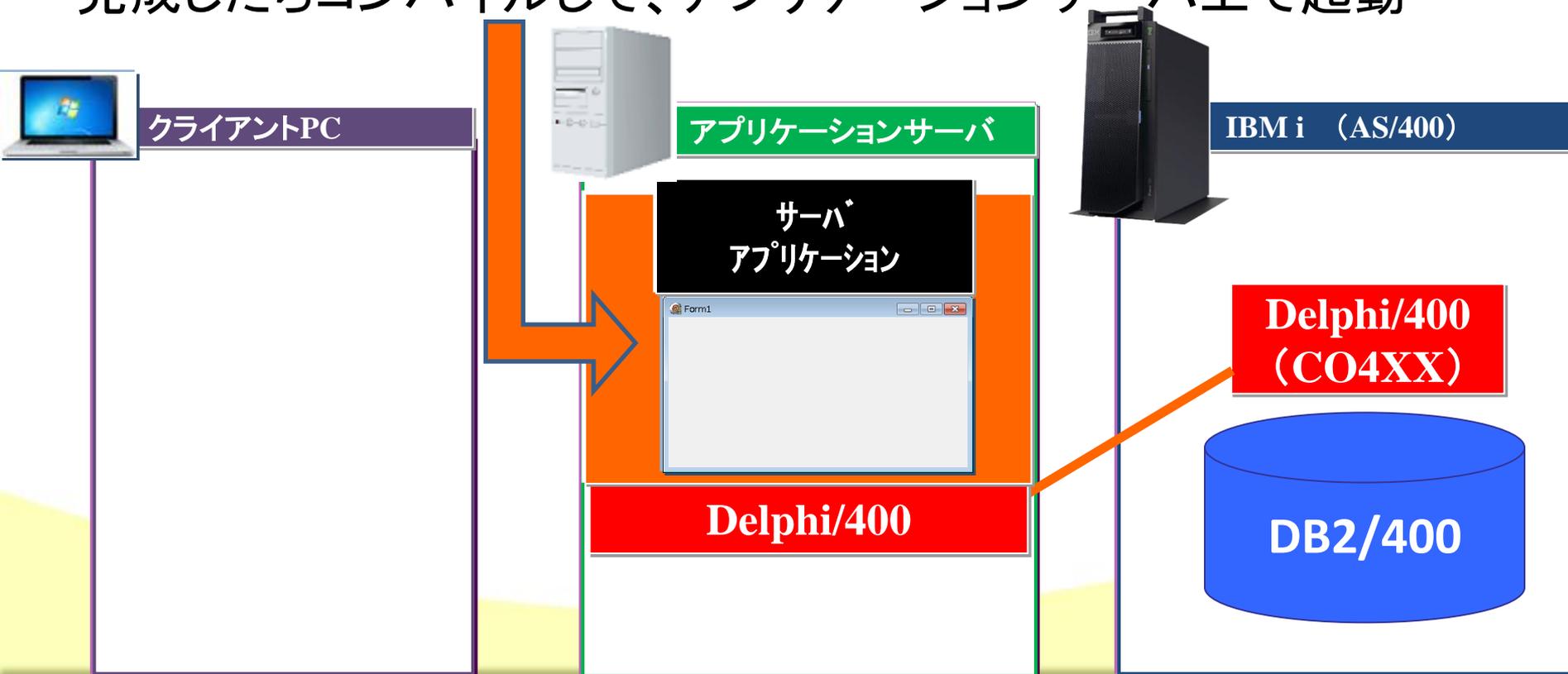
Optionプロパティの  
"poAllowCommandText"を  
"True"に設定



## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

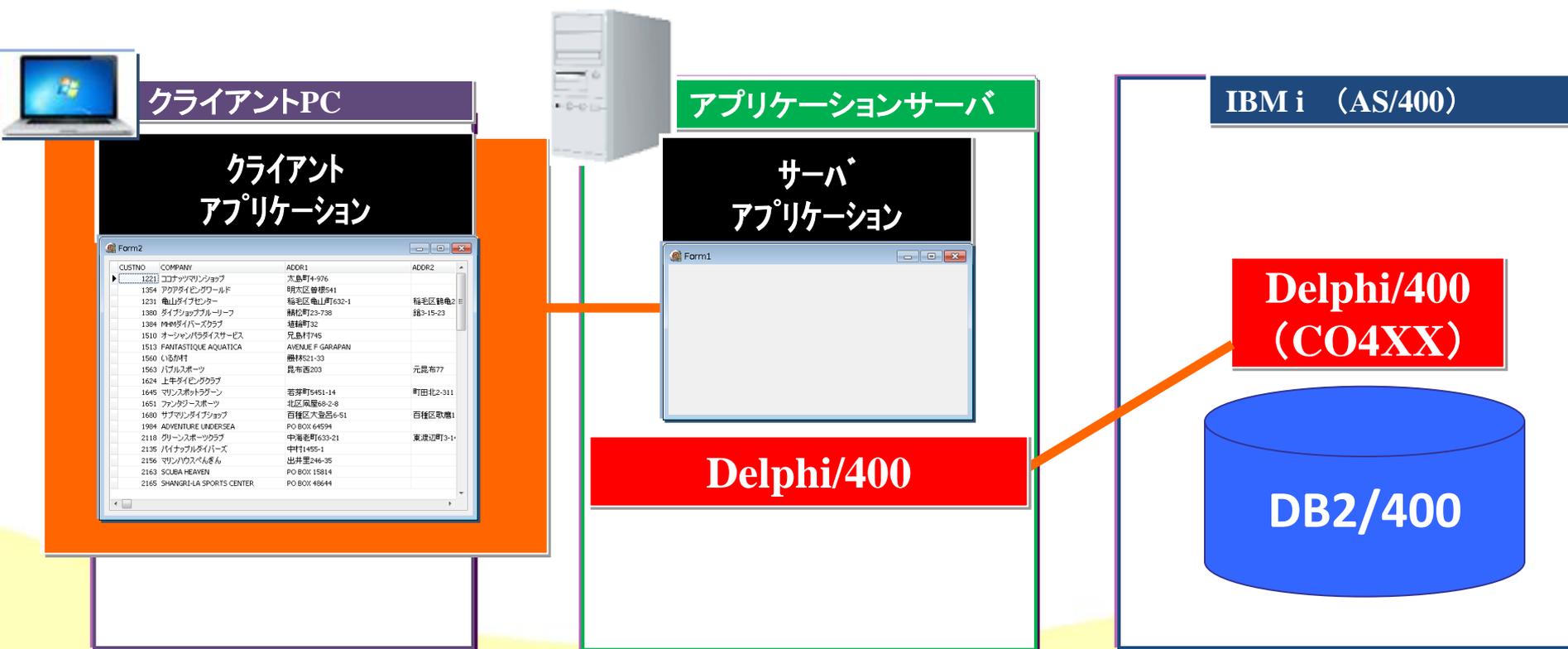
### • サーバアプリケーションの作り方⑧

完成したらコンパイルして、アプリケーションサーバ上で起動



## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

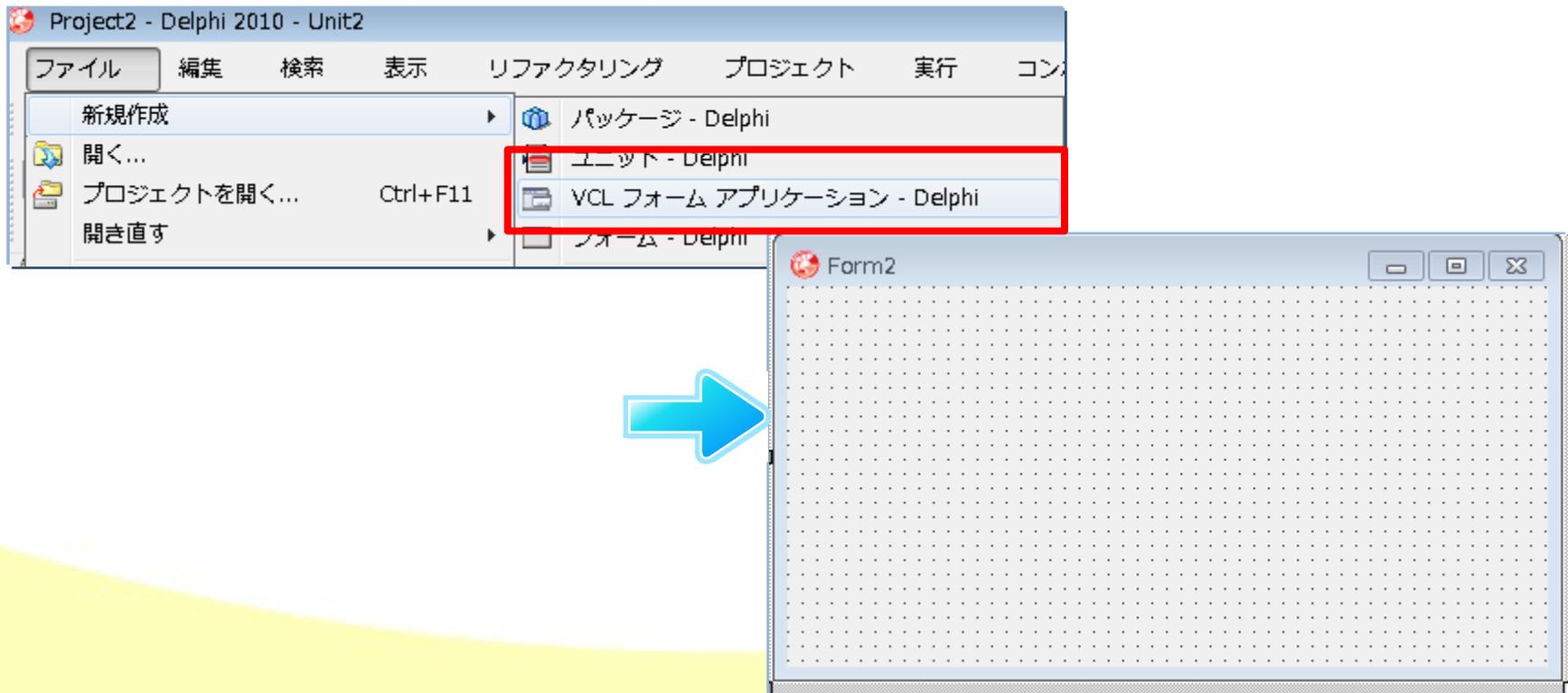
- クライアントアプリケーションの作り方



## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### ・ クライアントアプリケーションの作り方①

メニューの[ファイル|新規作成|VCLフォームアプリケーション] を選択



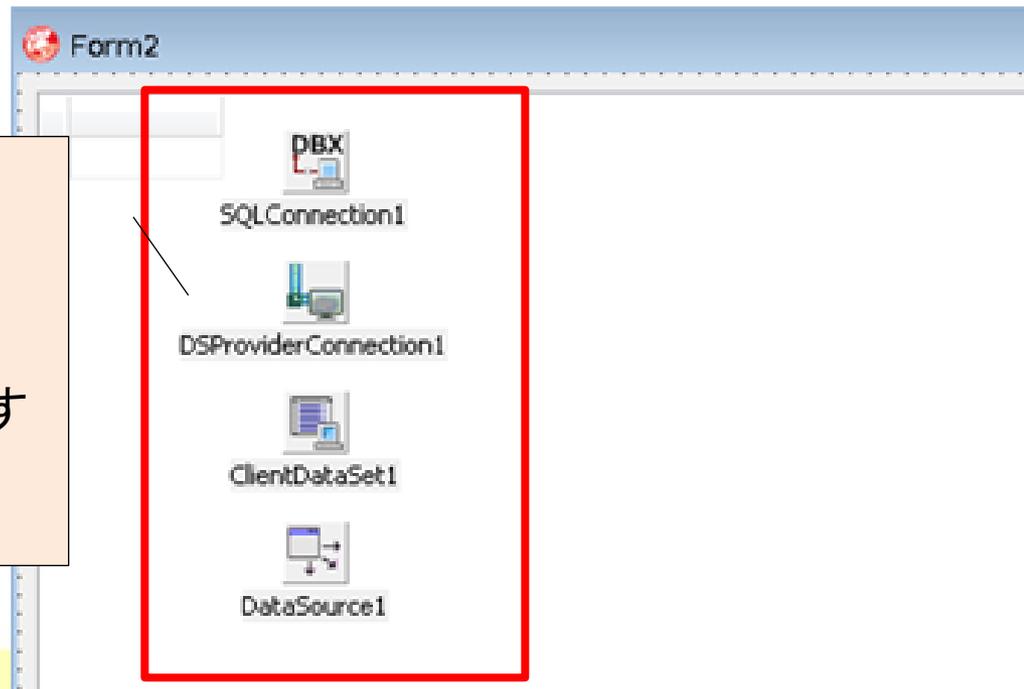
## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### • クライアントアプリケーションの作り方②

フォームに次のコンポーネントを配置

SQLConnection、DSProviderConnection、ClientDataSet、DataSource、DBGrid

今回はサーバアプリケーションを利用して、IBM iに接続せずにデータを表示する機能を実装します



## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### • クライアントアプリケーションの作り方③

#### SQLConnectionコンポーネントの設定

The screenshot shows the Visual Studio IDE with the following elements:

- Form2**: A form with a **SQLConnection1** component highlighted by a red box.
- Object Inspector**: Shows the properties of **SQLConnection1**. The **ConnectionName** property is set to **DATASNAPCONN**. The **LoginPrompt** property is set to **False**. The **Params** property is set to **(TStrings)**.
- Value List Editor**: A dialog box titled "値リストの編集" (Edit Value List) with the following table:

キー	値
DriverName	DataSnap
HostName	localhost
port	211

Callouts and annotations:

- Red boxes highlight **SQLConnection1**, **ConnectionName**, **LoginPrompt**, and **Params** in the Object Inspector.
- Red boxes highlight **HostName** and **port** in the Value List Editor.
- Text boxes provide instructions: "LoginPromptプロパティを'False'に設定", "Paramsプロパティを設定", "ConnectionNameプロパティを'DATASNAPCONNECTION'に設定", and "アプリケーションサーバ名" (pointing to DriverName).
- A text box states "使用するポート デフォルト:211" (Using port default: 211) pointing to the port value.

## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### • クライアントアプリケーションの作り方④

#### DSPProviderConnectionコンポーネントの設定

ServerClassNameプロパティを" TServerMethods1"に入力設定

SQLConnectionプロパティを" SQLConnection1"に設定

オブジェクトインスペクタ

DSPProviderConnection1 TDSPProviderConnection

プロパティ イベント

Connected	<input type="checkbox"/> False
Name	DSPProviderConnection1
ServerClassName	TServerMethods1
SQLConnection	SQLConnection1
Tag	0

ServerClassName

すべての項目が表示されています

## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### • クライアントアプリケーションの作り方⑤

#### ClientDataSetコンポーネントの設定

② RemoteServerを設定していると  
ProviderNameプロパティが選択できるの  
で”DataSetProvider1”を設定  
※サーバアプリケーションが起動している必要があります。

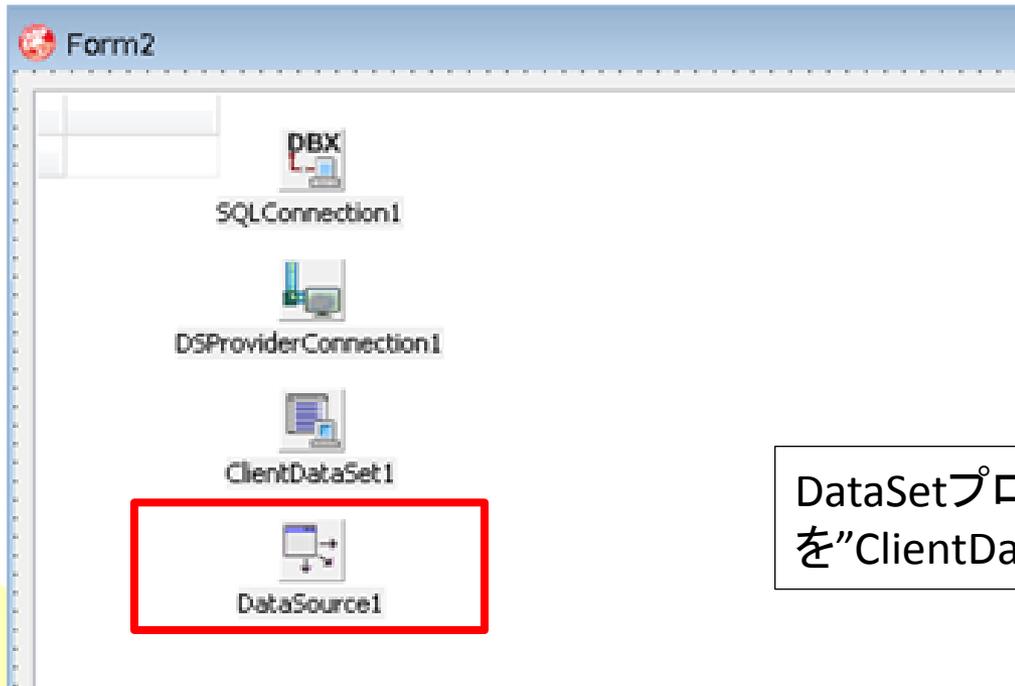
CommandTextプロパティ  
に実行するSQLを設定  
例) SELECT \* FROM CUSTOMER

① RemoteServerプロパティに  
”DSPProviderConnection1”を設定

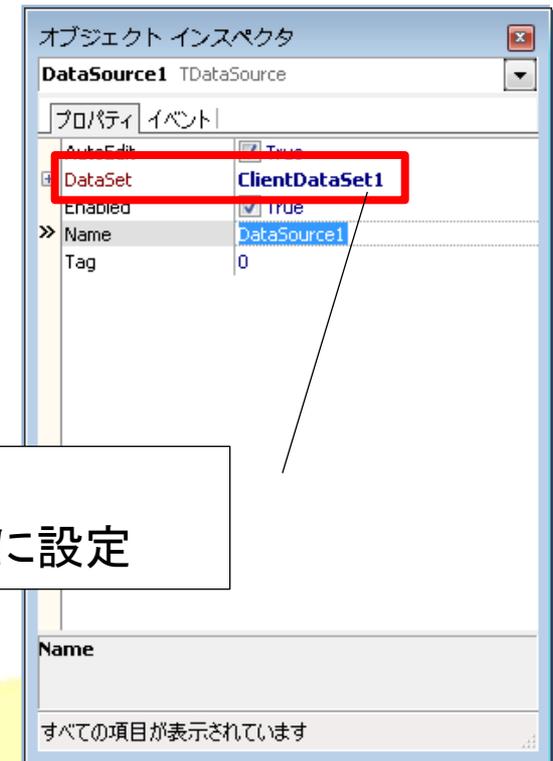
## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### • クライアントアプリケーションの作り方⑥

#### DataSourceコンポーネントの設定



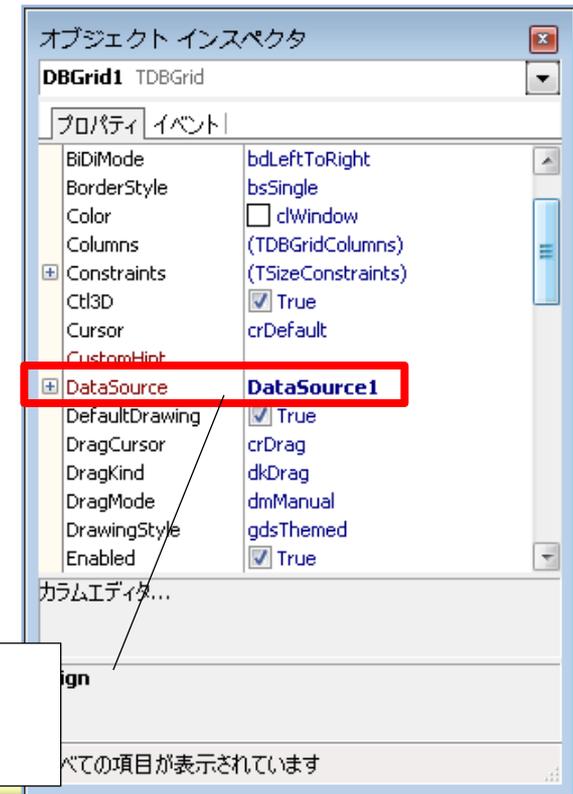
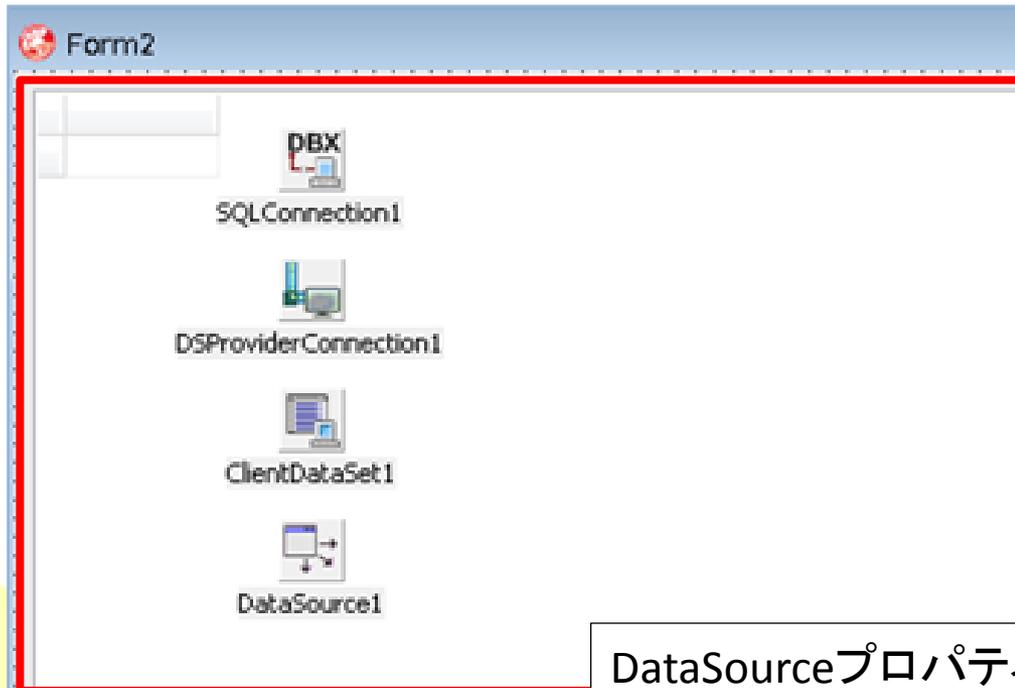
DataSetプロパティ  
を”ClientDataSet1”に設定



## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### • クライアントアプリケーションの作り方⑦

#### DBGridコンポーネントの設定



DataSourceプロパティ  
を"DataSource 1"に設定

## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

### • クライアントアプリケーションの作り方⑧

データの接続(サーバアプリケーションへ接続)

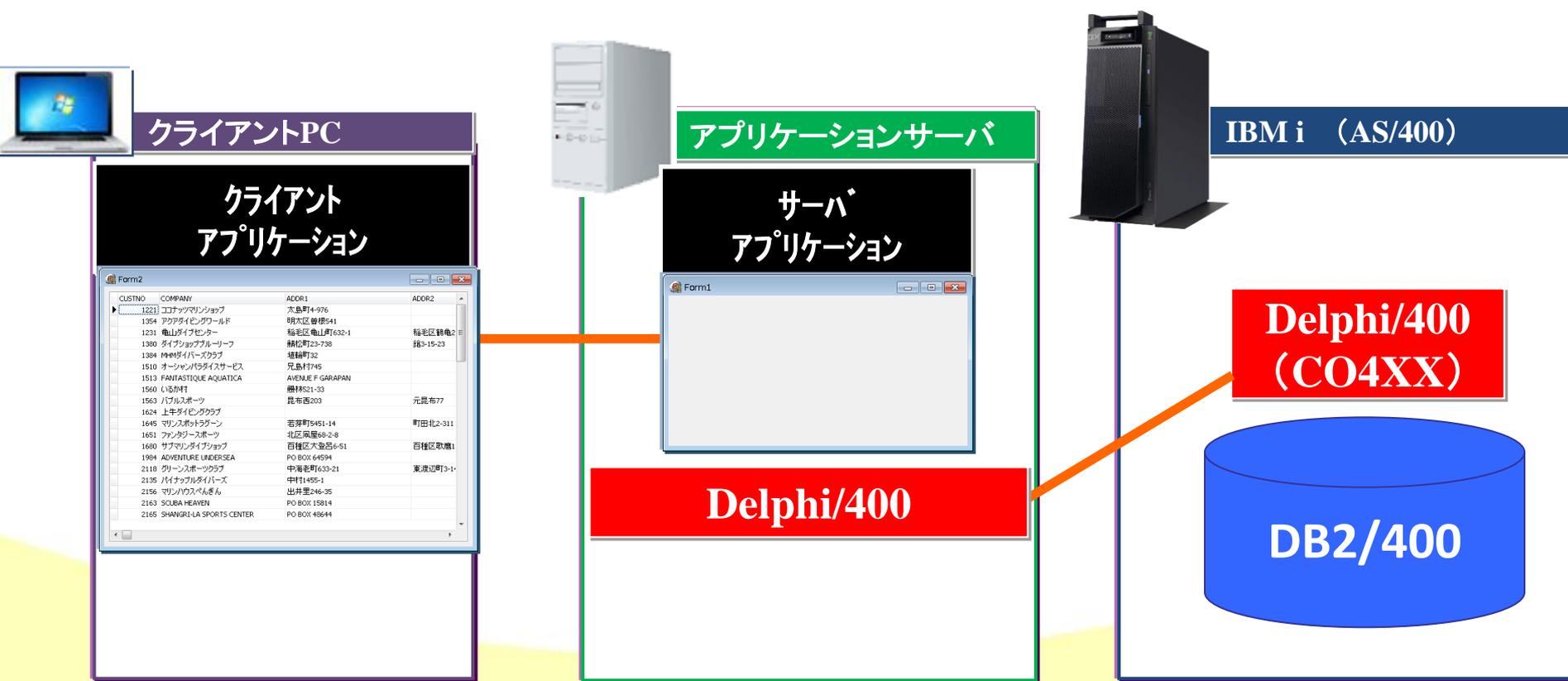
CUSTNO	COMPANY	ADDR1
1221	ココナ DBX リンショップ	太島町4-976
1354	マホク SQLConnection1	明太区曾根541
1231	亀山ダイブセンター	稲毛区亀山町632-1
1380	ダイブ ブルーリーフ	鯉松町23-738
1384	MSMProviderConnection1	埴輪町32
1510	オープンタンバコサービス	兄島村745
1513	FAN QUE AQUATICA	AVENUE F GARAPAN
1560	ClientDataSet1	鯉林521-33
1563	ClientDataSet1	昆布西203
1624	上牛 シングクラブ	
1645	マリン トラグーン	若芽町5451-14
1651	-DataSource1	
1680	サブマリンダイブショップ	

IBM iに接続する  
設定はありませんが  
IBM iの情報を取得して  
表示することに成功！

ClientDataSetコンポーネントの  
Activeプロパティを"True"に設定するとデータが表示

## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

- 3階層によってクライアントがEXEだけでIBM iへ接続完成



## ■ Q3. C/Sアプリケーションで3階層接続を行うには？

- 2階層と3階層のメリット  デメリット 

### 2階層

Native接続やセッションなど  
Delphi/400の機能をフル活用できる。

クライアントPC毎にIBM i への接続の  
環境構築が必要。

### 3階層

基本dbExpressでSQL中心の処理になる。  
Nativeなどは作りこみが必要。

クライアントPC毎にIBM i への接続の  
環境構築が不要。



2階層より劣る点もありますが、Webブラウザよりも  
自由なC/S画面を使いつつ、PC管理も軽減できます。

ご清聴ありがとうございました。