【A2】Delphi/C++Builderテクニカルセッション

はじめてのDataSnap

2013年X月X日

株式会社三菱電機ビジネスシステム

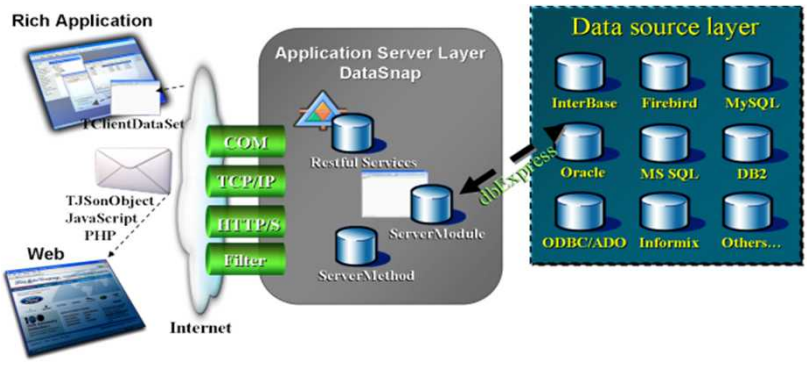
田中 芳起

Ver.1.0.0

DataSnapの概要

多層型のデータベースアプリケーションを構築するためのフレームワーク

・Delphi3で実装された技術で、Delphi5までは「MIDAS」と呼ばれていたが、Delphi6からは「DataSnap」と名称を変更
・Win32ベースのWindowsネイティブアプリケーション開発を前提とした技術だった
・Delphi2009からは、COM依存性を排除して再構築。「.NETクライアント」からも接続可能となった
・Delphi 2010からは、Windowsサービス／Webサービス／インターネットサーバーAPI／HTTP認証 等をサポート



はじめてのDataSnap

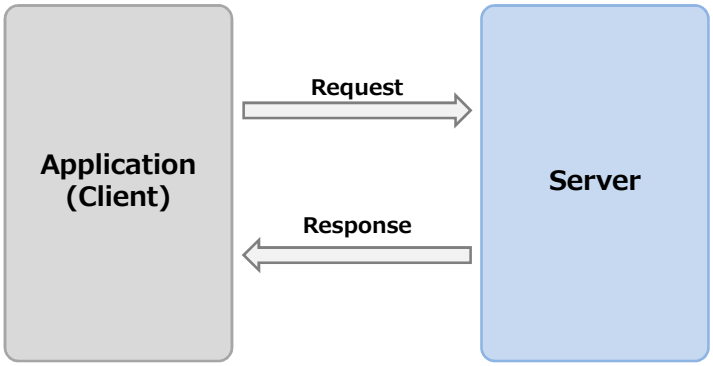
作成するアプリケーションの説明

Visual N@VI
Oracle Object Management Tool

アプリケーションの概要

これから作成するDataSnapを使ったアプリケーションは、次の通り

・クライアントからリクエストを送出し、サーバーからの結果をクライアントに表示する
・プロトコルは、TCP/IPを使用する



Visual N@VI
Oracle Object Management Tool

はじめてのDataSnap

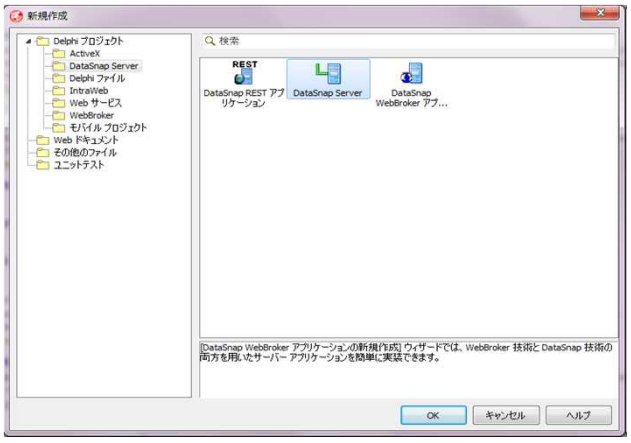
サーバーの作成

Visual N@VI
Oracle Object Management Tool

新規プロジェクトを作成

ウィザードを使って新規プロジェクトを作成する

・[ファイル | 新規作成 | その他...] メニューを選択すると、下の画面が表示される
・左のペインから「DataSnap Server」、右のペインから「DataSnap Server」を選択し「OK」ボタンを押す



Visual N@VI
Oracle Object Management Tool

サーバー機能の選択

DataSnapサーバーに追加する機能を選択する

・(通信)プロトコル
・認証
・サーバーメソッドクラス
・フィルタ
・JavaScriptファイル
・モバイルコネクタ

ここでは、標準設定のまま「次へ」を押す

Visual N@VI
Oracle Object Management Tool

プロジェクト種類の選択

DataSnapサーバーの種類を選択する

作成するアプリケーション・タイプを次の3種類から選択

・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
・(Windows)サービスアプリケーション

ここでは、「VCLフォームアプリケーション」を
選択し「次へ」を押す

Visual N@VI
Oracle Object Management Tool

ポート番号の指定

クライアントからのリスニングを行うポート番号を指定する

前ページで選択した「プロトコル」のポート番号を指定

ここでは、標準設定のまま「次へ」を押す

※[ポートのテスト]ボタンを押すと、指定のポートが使用可能かどうかを確認できる

Visual N@VI
Oracle Object Management Tool

サーバー・メソッド・クラスの上位クラスを指定

サーバーメソッド クラスの上位の型を指定する

・TComponent
・TDataModule
・TDSServerModule

ここでは、「TComponent」を選択し「完了」を押す

Visual N@VI
Oracle Object Management Tool

サーバープログラムの自動生成

ウィザードで次の3つのユニットが自動生成される

[Unit1]

```
unit Unit1;

interface

uses Winapi.Windows, Wi
  System.Classes, Vcl.Gr

type
  TForm1 = class(TForm)
    procedure FormClose(
  private
    { private 宣言 }
  public
    { public 宣言 }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

uses ServerContainerUnit

end.
```

[ServerMethodsUnit1]

```
unit ServerMethodsUnit1;

interface

uses System.SysUtils,

type
{$METHODINFO ON}
  TServerMethods1 = cl
  private
    { private 宣言 }
  public
    { public 宣言 }
    function EchoStrin
    function ReverseSt
  end;
{$METHODINFO OFF}

implementation

uses System.StrUtils;

function TServerMetho
begin
  Result := Value;
end;

  :
```

[ServerContainerUnit1]

```
unit ServerContainerUnit1;

interface

uses System.SysUtils, System.Classes,
  Datasnap.DSTCPServerTransport,
  Datasnap.DSServer, Datasnap.DSCommonServer,
  Datasnap.DSAuth, IPPeerServer;

type
  TServerContainer1 = class(TDataModule)
    DSServer1: TDSServer;
    DSTCPServerTransport1: TDSTCPServerTransport;
    DSServerClass1: TDSServerClass;
    procedure DSServerClass1GetClass(DSServerClass: TDSServerClass;
      var PersistentClass: TPersistentClass);
  private
    { private 宣言 }
  public
  end;

var
  ServerContainer1: TServerContainer1;

implementation

uses Winapi.Windows, ServerMethodsUnit1;

{$R *.dfm}
```


Visual N@VI
Oracle Object Management Tool

サーバー・コンテナ ユニットを見てみる（1/2）

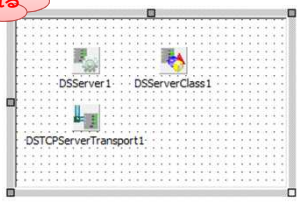
ウィザードで自動生成される「ServerContainerUnit1」には、3つ*1 のコンポーネントが配置されている

- **TDSServer :**
  すべてのDataSnapコンポーネントを結びつけるための
  メインのサーバー設定コンポーネント

- **TDSServerClass :**
  公開するクラスごとに必要なコンポーネント。
  publicなインターフェイスを持つクラスを参照する

- **TDSTCPServerTransport :**
  転送プロトコルと使用するTCP/IP等の設定を定義する
  コンポーネント

必ず配置される



*1 「サーバー機能の選択」でチェックした内容によって配置されるコンポーネントが異なる

Visual N@VI
Oracle Object Management Tool

サーバー・コンテナ ユニットを見てみる（2/2）

```
unit ServerContainerUnit1;

interface

uses System.SysUtils, System.Classes,
  Datasnap.DSTCPServerTransport,
  Datasnap.DSServer, Datasnap.DSCommonServer,
  Datasnap.DSAuth, IPPeerServer;

type
  TServerContainer1 = class(TDataModule)
    DSServer1: TDSServer;
    DSTCPServerTransport1: TDSTCPServerTransport;
    DSServerClass1: TDSServerClass;
    procedure DSServerClass1GetClass(DSServerClass: TDSServerClass;
      var PersistentClass: TPersistentClass);
  private
    { private 宣言 }
  public
  end;

var
  ServerContainer1: TServerContainer1;

implementation

uses Winapi.Windows, ServerMethodsUnit1;

{$R *.dfm}

procedure TServerContainer1.DSServerClass1GetClass(
  DSServerClass: TDSServerClass; var PersistentClass: TPersistentClass);
begin
  PersistentClass := ServerMethodsUnit1.TServerMethods1;
end;

end.
```

OnGetClassが自動追加されている

Server Class

Visual N@VI
Oracle Object Management Tool

TDSServerコンポーネント

すべてのDataSnapコンポーネントを結びつけるためのメインのサーバー設定コンポーネント

プロパティ／メソッド	説　明
AutoStart	True: TDSServerコンポーネントがロードされたときにサーバーを自動的に起動
ChannelQueueSize	送信のキューに入れられるメッセージの数を指定
ChannelResponseTimeout	クライアントがレスポンスを返すまで待機する時間を指定 (単位：ミリ秒)
Started	True: DataSnapサーバーを開始されている
Start	DataSnapサーバーを開始する (AutoStartがFalseの場合、このメソッドを呼ぶ)
Stop	DataSnapサーバーを停止する

```
var
 MyDSServer: TDSServer;
begin
  // DataSnapサーバーを作成
  MyDSServer := TDSServer.Create(Self);
  // DataSnapサーバーを開始
  if MyDSServer.Started then MyDSServer.Start;

  // DataSnapサーバーを停止し、メモリを解放
  MyDSServer.Stop;
  MyDSServer.Free;
end;
```

Visual N@VI
Oracle Object Management Tool

TDSServerClassコンポーネント

リモートクライアントにpublicメソッドを公開するために使われるサーバーサイドのクラスを定義する
公開するクラスごとに必要なコンポーネント

プロパティ／メソッド	説　明
LifeCycle	インスタンス(Server class)のライフ サイクルを指定 ・**Session**：DataSnapSession毎にインスタンスを使用 (既定値) ・**Server**: サーバー毎にインスタンスを使用 ・**Invocation**: メソッドの呼び出し毎にインスタンスを使用
Server	DataSnapサーバーを指定
OnGetClass	サーバークラスを指定

[OnGetClassの使用例]

```
procedure TServerContainer1.DSServerClass1GetClass(
  DSServerClass: TDSServerClass; var PersistentClass: TPersistentClass);
begin
  PersistentClass := ServerMethodsUnit1.TServerMethods1;
end;
```


Visual N@VI
Oracle Object Management Tool

TDSTCPServerTransportコンポーネント

DataSnapのサーバーとクライアント間の通信を担当し、TCP/IPを通信プロトコルとして使用する

プロパティ／メソッド	説　明
AuthenticationManager	認証マネージャを指定
BufferKBSize	TCP/IPが読み／書きするバッファ サイズを指定 (単位：KB)
Filters	プロセス内インスタンスに対する、DataSnap通信フィルタを指定
MaxThreads	スケジューラで許されるスレッドの最大数を指定
PoolSize	スレッド プールに割り当てられる最大数を指定
Port	TCP/IPサーバーサイド ポートを指定 (デフォルトは211)
Server	DataSnapサーバーを指定

Visual N@VI
Oracle Object Management Tool

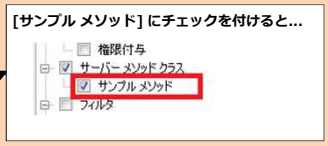
サーバー・メソッド ユニットを見てみる

ServerMethodsUnit1には、サーバー・クラスとサーバーメソッドが自動的に追加されている

```
unit ServerMethodsUnit1;

interface

uses System.SysUtils, System.Classes, Datasnap.DSServer, D

type
{$METHODINFO ON}
  TServerMethods1 = class(TComponent)
  private
    { private 宣言 }
  public
    { public 宣言 }
    function EchoString(Value: string): string;
    function ReverseString(Value: string): string;
  end;
{$METHODINFO OFF}

implementation

uses System.StrUtils;

function TServerMethods1.EchoString(Value: string): string;
begin
  Result := Value;
end;

function TServerMethods1.ReverseString(Value: string): string;
begin
  Result := System.StrUtils.ReverseString(Value);
end;

end.
```

Server Class

[サンプル メソッド] にチェックを付けると...



Visual N@VI
Oracle Object Management Tool

サーバー・メソッドの追加

サーバー・クラス(TServerMethods1)にメソッドを追加する

[サーバー・クラス]

```
{$METHODINFO ON}
  TServerMethods1 = class(TComponent)
  private
    { private 宣言 }
  public
    { public 宣言 }
    function EchoString(Value: string): string;
    function ReverseString(Value: string): s
    function Add(a, b: Double): Double;
    function Dic(a, b: Double): Double;
    function Mult(a, b: Double): Double;
    function Sub(a, b: Double): Double;
  end;
{$METHODINFO OFF}
```

追加！

[追加したサーバー・メソッド]

```
function TServerMethods1.Add(a, b: Double): Double;
begin
  Result := a + b;
end;

function TServerMethods1.Dic(a, b: Double): Double;
begin
  Result := a / b;
end;
...
```

Visual N@VI
Oracle Object Management Tool

インスタンスを停止するコードを付加する

フォームを閉じる前に DataSnapサーバーのインスタンスを停止させるコードを追加する

```
uses ServerContainerUnit1; *1

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  ServerContainer1.DSServer1.Stop;
end;
```

*1 usesに「 ServerContainerUnit1 」を追加する

Visual N@VI
Oracle Object Management Tool

はじめてのDataSnap

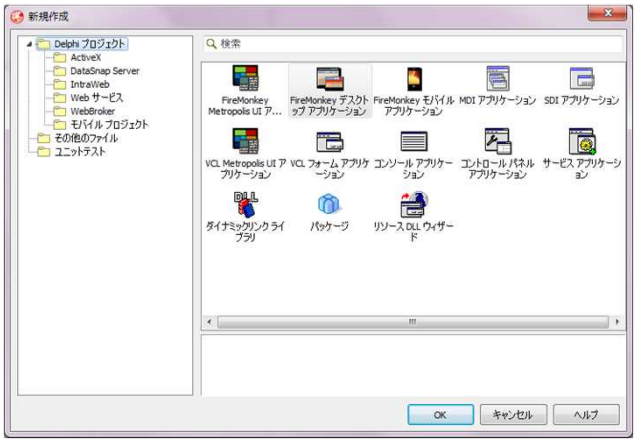
クライアントの作成

Visual N@VI
Oracle Object Management Tool

新規プロジェクトを追加

プロジェクトグループにクライアントの新規プロジェクトを作成する

・プロジェクト・マネージャのプロジェクト・グループ名を右クリックし「新規プロジェクトの追加…」を選択
・左のペインから「Delphi プロジェクト」を選択、右のペインから「FireMonkey デスクトップ アプリケーション」を選択し、「OK」ボタンを押す

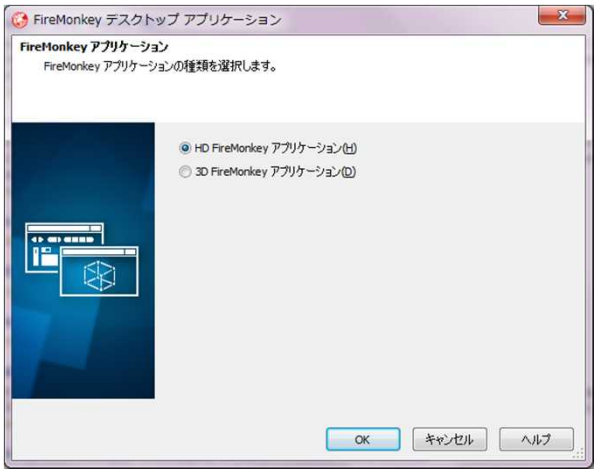


Visual N@VI
Oracle Object Management Tool

FireMonkyアプリケーションを作成する

FireMonkey アプリケーションの種類を選択する

・「HD FireMonkey アプリケーション」を選択し、「OK」ボタンを押す



Visual N@VI
Oracle Object Management Tool

FireMonkyアプリケーションを作成する

コントロールをFormに配置する

・プロジェクトとプロジェクト・グループを保存する

Visual N@VI
Oracle Object Management Tool

DataSnap クライアント・クラスの作成

重要なポイントは、サーバーで実装されているすべてのインターフェースを作成すること

・プロジェクト・グループでサーバー・プログラムを選択し、「デバッガを使わずに実行」で起動する

・