

【セッションNo. 2】

Delphi/400 最新技術情報

新バージョンDelphi/400 10 Seattle のご紹介

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】

## 1.Windowsとモバイルの動向

## 2.新バージョンDelphi/400 10 Seattle

2-1.Windows10の正式サポート

2-2.強化されたIoT機能

2-3.マルチデバイス開発機能の拡充

2-4.FireDACを使ったIBM i 接続機能

2-5.IDE機能の強化

## 3.まとめ

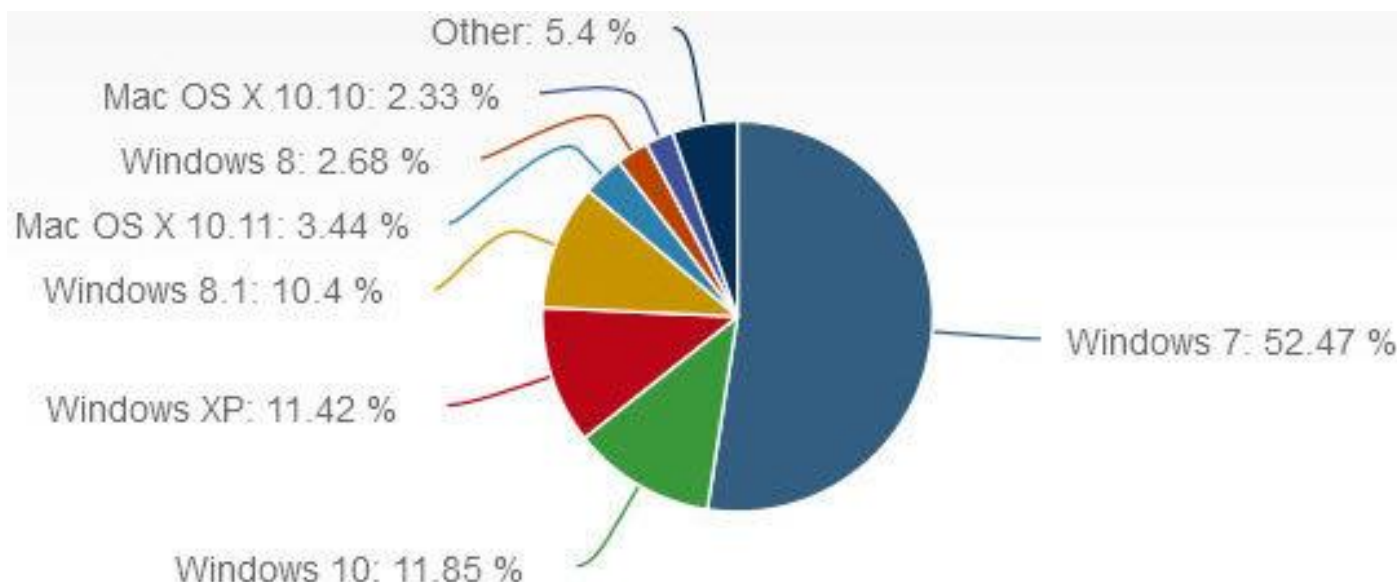
# 1.Windowsとモバイルの動向

## ■ 1.Windowsとモバイルの動向

- Windowsの動向

Windows10がリリースされ、そろそろ1年になり、導入数も大きく進展。

Windows10の利用率は既にXPや8.1を超えている。



NetMarketShare 2016年デスクトップ向けOS比率

# ■ 1.Windowsとモバイルの動向

## • モバイルの動向

企業のモバイル導入がここ数年で急速に進んだことで、モバイルに対する視点も変化がある。

### 企業でのモバイル利用の視点

#### これまでの取り組み

- ・導入機器や環境の選定
- ・使い方の浸透(用意された用途範囲)



#### 最近の取り組み

- ・業務に特化した活用・モバイルシステム構築
- 新しい活用用途の開拓!

## ■ 1.Windowsとモバイルの動向

- 近日リリースを予定しているDelphi/400の新バージョン『10 Seattle』はそうしたIT背景に応える機能を備えて登場！



## 2.新バージョンDelphi/400 10 Seattle

## ■ 2.新バージョンDelphi/400 10 Seattle

- 新バージョンDelphi/400 10 Seattle

Windows10対応をはじめ、モバイルやIBM i のデータベースエンジン機能など大幅な機能強化が実装されたバージョン

### 【 Delphi/400 10 Seattle の特徴】

2-1.Windows10の正式サポート

2-2.強化されたIoT機能

2-3.マルチデバイス開発機能の拡充

2-4.FireDACを使ったIBM i 接続機能

2-5.IDE環境の強化



## ■ 2-1.Windows10の正式サポート

- Windows 10を正式動作環境としてサポート

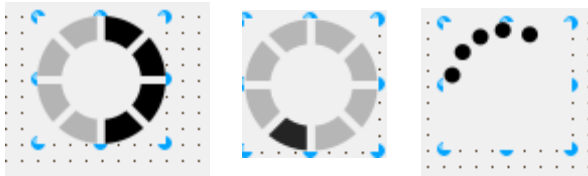


Windows 10

- Windows 10の新しいユーザーインターフェースを使用できる新VCLコンポーネントを多数追加
  - TActivityIndicatorコンポーネント
  - TSearchBoxコンポーネント
  - TToggleSwitchコンポーネント
  - TSplitViewコンポーネント
  - Windows 10 スタイル(スタイル)

## ■ 2-1.Windows10の正式サポート

- TActivityIndicatorコンポーネント



Windows10風のアニメーション  
コンポーネント

- TSearchBoxコンポーネント



エディットと検索ボタンが  
一体化したコンポーネント

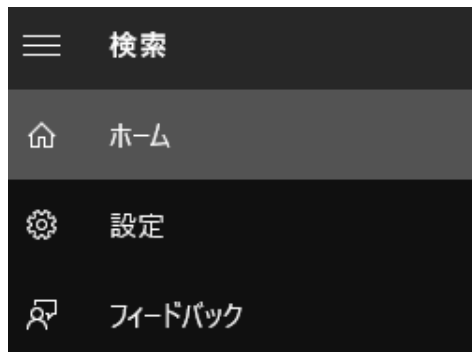
- TToggleSwitchコンポーネント



Windows10風のスイッチコンポーネント

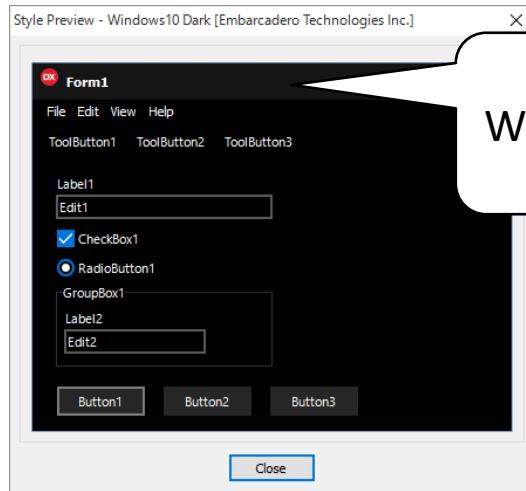
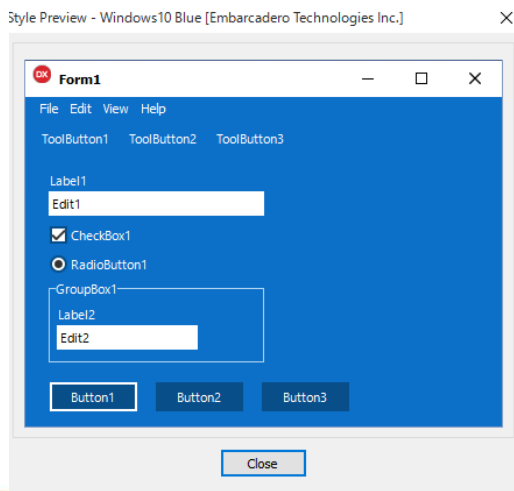
## ■ 2-1.Windows10の正式サポート

- TSplitViewコンポーネント



Windows10のライド風に表示する  
ビューコンポーネント

- Windows 10 スタイル(スタイル)



Windows10風の専用スタイル

## ■ 2-1.Windows10の正式サポート

- Windows10の新しいユーザーインターフェース例



TSplitView

Windows 10 スタイルBlack

TSearchBox

Windows10サンプル

第18回テクニカルセミナー

第1セッション  
第2セッション  
第3セッション  
第4セッション

トラック: 基調講演

タイトル: IoTでどう変わる? これからの業務システムとは

会社: エンバカデロ・テクノロジーズ

講演者: 日本法人代表 藤井 等 様

高い注目を集めているIoT (Internet of Things: モノのインターネット)。従来の企業システムとは直接関係ない将来の技術と捉えている方も多いようです。しかし、IoTが真の価値を発揮するのは、企業システムと連携したときです。  
IoTとは何か? 企業システムにはどのように関わるのか? また、どのようにこれらの技術を取り入れていくことができるのか?  
今回、IoTサポートを強化したDelphiによる実演を交え、この最新技術がどのように企業システムを変えていくのかを紹介します。

Windows10サンプル

第18回テクニカルセミナー

トラック: Delphi/400 最新技術情報

タイトル: 新バージョンDelphi/400 10

会社: 株式会社ミガロ。

講演者: 吉原 泰介

昨年よりWindows10や新しいモバイル機器を導入・検討される企業も増えてきました。この背景を踏まえ、今年の新機能「Delphi/400 10」では、「V FireDAC」新機能

TToggleSwitch

オン

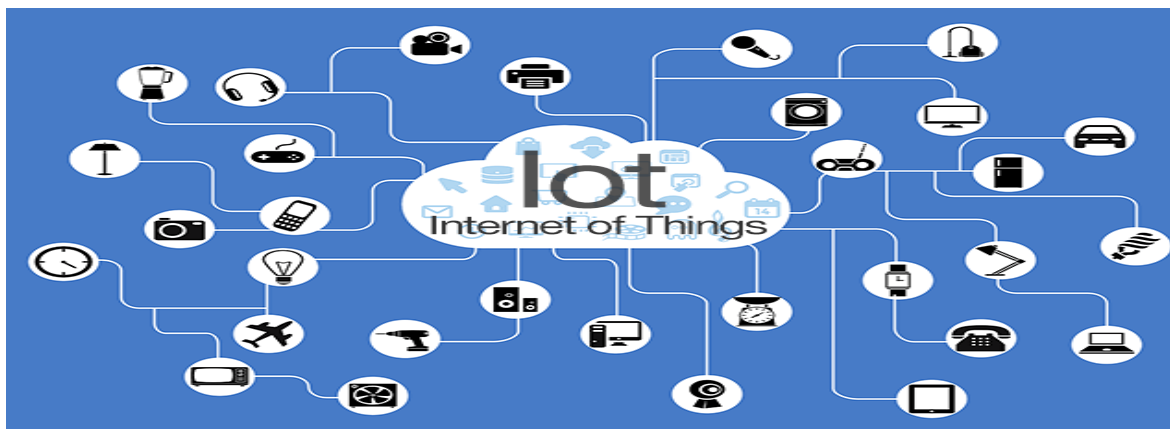
TActivityIndicator  
からの読み込み



## ■ 2-2.強化されたIoT機能

### • IoTとは？

Internet of Thingsの略で「モノのインターネット」とも呼ばれる。PC等の機器だけでなく、世の中の様々なモノに通信機能を持たせ、インターネットに接続したり相互に通信することで、自動制御、計測、データの収集などを活用する。



例えば、リストバンドで健康管理の情報を連動したり、バスや電車などのリアルタイムな運行情報を公開発信したり、施設内での位置にあわせて音声ガイドを発信するなど、様々な用途で試みが行われています。  
(お店でクーポンや商品情報が発信されるのも、そのひとつ)

## ■ 2-2.強化されたIoT機能

### • Beacon (ビーコン)とは？

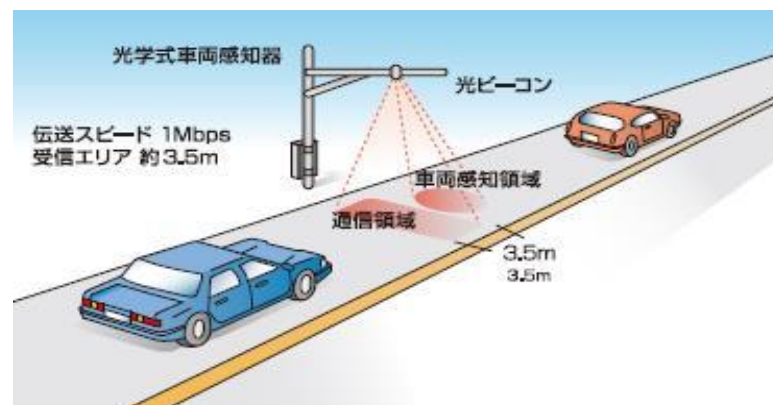
IoTが身近なところで代表される機能のひとつがBeacon(ビーコン)。BeaconはBluetoothを使い、半径十数メートルに信号を発信する。用途としては、スマートフォンなどの機器が信号範囲に入ってきたら情報を通知したり(お店のクーポン配信等)、信号範囲から位置を特定して位置情報機能として使うこともできる。

#### 【利用例】

道路交通情報通信システム(VICS)では、渋滞の状況や所要時間、通行止めなどの情報を電波ビーコンや光ビーコンで車に情報を伝えている。

(一般財団法人 道路交通情報通信システムセンター)

また登山で携帯する「雪崩ビーコン」もその名の通り、非常時に場所を発信するBeacon機器。



## ■ 2-2.強化されたIoT機能

- TBluetooth、TBluetoothLEコンポーネント  
画面上にドラッグ & ドロップするだけでアプリケーションを  
Bluetooth、BluetoothLE(LowEnergy)に対応できる



- TBeaconコンポーネント  
Beacon情報を感知する



MonitorizedRegionsプロパティ内の  
アイテムでUUIDを指定

MonitorizedRegions	(TBeaconRegionCollection)
UUID	{00000000-0000-0000-0000-000000000000}

- TBeaconDeviceコンポーネント  
Beacon情報を発信する



UUIDプロパティを指定

UUID	{00000000-0000-0000-0000-000000000000}
------	--

※UUIDとは、全世界で同じ値を持つことがない一意な識別子  
BeaconはUUID、MajorID、MinorIDなどで信号を判別します。

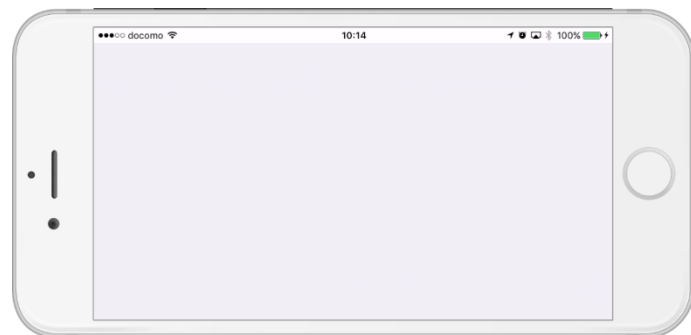


## ■ 2-2.強化されたIoT機能



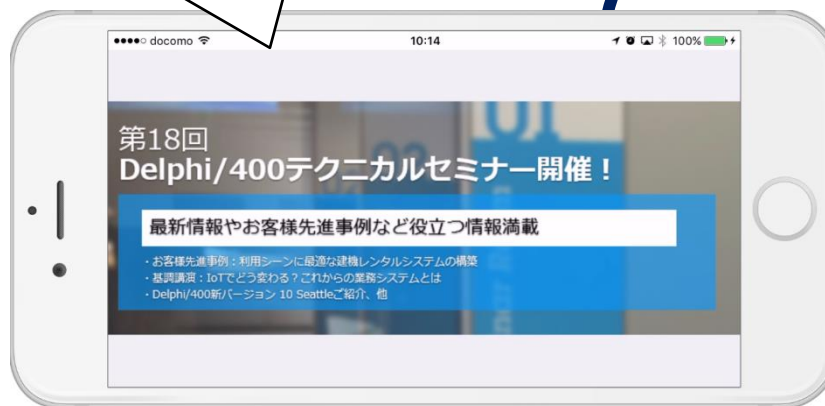
- Beacon受信サンプルアプリ開発  
機能概要

### Beaconエリア外



移動

② Beaconの信号エリアに入るとアプリで広告表示！



① Beacon機器※から信号を発信

Beacon発信エリア



Beacon機器

※Beacon機器は業界の標準規格であるiBeaconとAltBeaconに対応

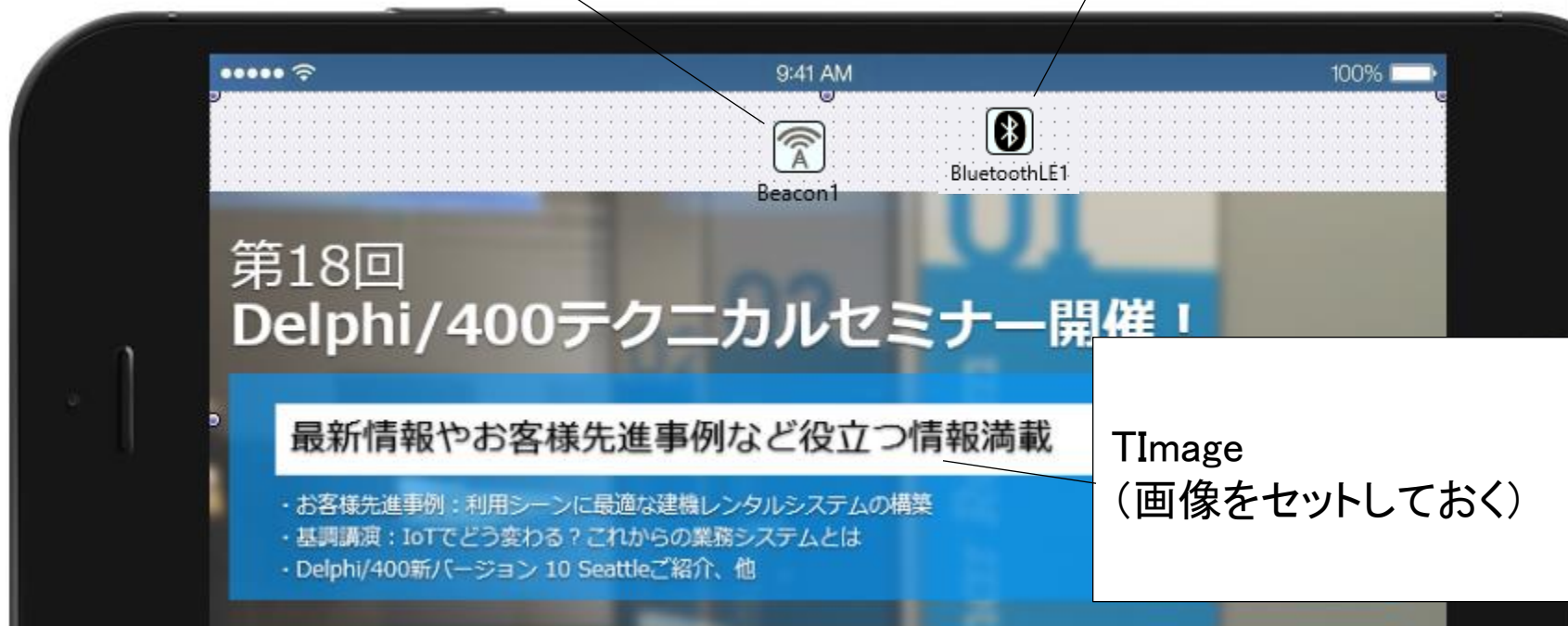


## ■ 2-2.強化されたIoT機能

- Beacon受信サンプルアプリ開発手順①  
コンポーネントの配置

TBeacon  
(Beacon処理用)

TBluetoothLE  
(配置だけ)



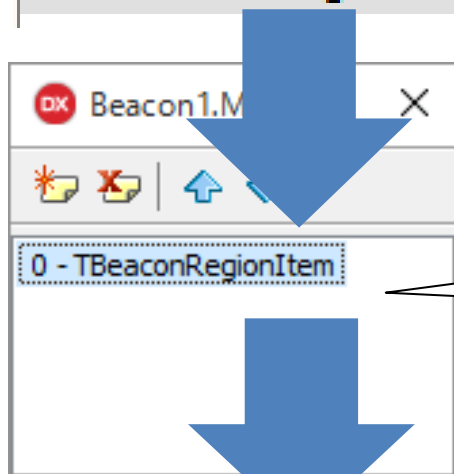
TImage  
(画像をセットしておく)

## ■ 2-2.強化されたIoT機能

- Beacon受信サンプルアプリ開発手順②

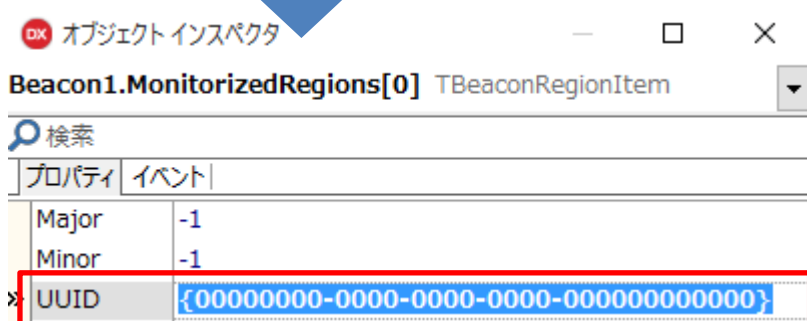
TBeaconプロパティの設定

MonitorizedRegions (TBeaconRegionCollection)



MonitorizedRegionsプロパティをダブルクリック

アイテムを追加



Beaconに合わせたUUIDをセット

## ■ 2-2.強化されたIoT機能

### • Beacon受信サンプルアプリ開発手順③

#### OnCreateイベント(初期処理)

```
procedure TForm1.FormCreate(Sender: TObject);  
Begin  
    Image1.Visible := False; //画像を非表示  
    Beacon1.Enabled := True; //Beaconを有効化  
end;
```

#### OnBeaconEnterイベント(Beaconエリアに入った処理)

```
procedure TForm1.Beacon1BeaconEnter(const Sender: TObject;  
    const ABeacon: IBeacon; const CurrentBeaconList: TBeaconList);  
begin  
    Image1.Visible := True; //画像を表示  
end;
```

#### OnBeaconExitイベント(Beaconエリアから出た処理)

```
procedure TForm11.Beacon1BeaconExit(const Sender: TObject;  
    const ABeacon: IBeacon; const CurrentBeaconList: TBeaconList);  
begin  
    Image1.Visible := False; //画像を非表示  
end;
```

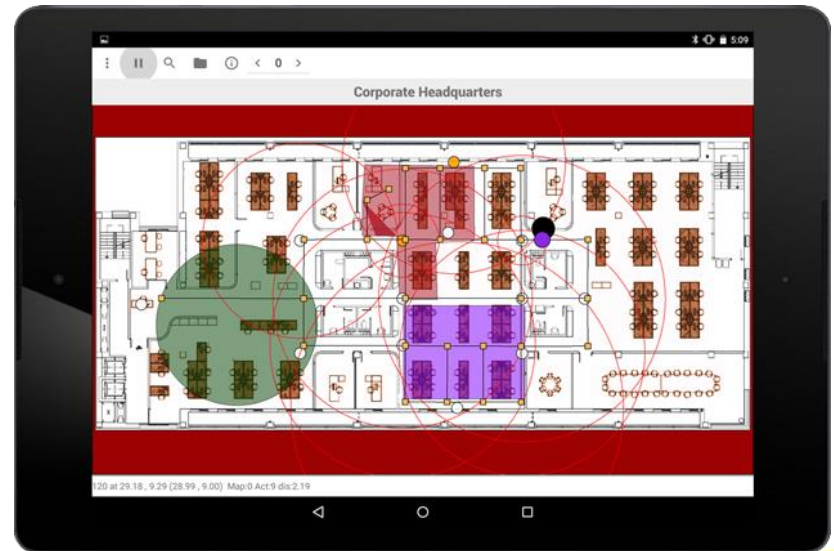
## ■ 2-2.強化されたIoT機能

### • BeaconFence

GPSを使用することのできない屋内でも、Beaconを複数個配置することで、より精密な位置情報を把握・制御することが可能。

BeaconFenceでは、複数Beaconからの位置制御を専用ビジュアルツールの「Beacon Fencing Map Editor」を用いることで、マップ上にビーコンの位置やゾーンを定義して容易に管理できるソリューションです。（無償/有償範囲あり）  
BeaconFenceの導入は GetItパッケージマネージャからインストール。

[GetItパッケージマネージャ] 詳細はP.36



## ■ 2-3.マルチデバイス開発機能の拡充

### • マルチデバイス開発

Delphiのマルチデバイス開発では、FireUIによって1つのプログラムからOSやデバイスのサイズ毎に画面設計をカスタマイズできる。  
これによりPCやモバイルのプログラムも共通化でき、高い生産性を実現！

マルチデバイス対応

1つのDelphiプログラム(ビジネスロジック)

デバイスの違いを  
細かく対応可能

FireMonkeyマルチデバイスデザイナー(FireUI)





## ■ 2-3.マルチデバイス開発機能の拡充



### ● マルチデバイスプレビュー

10 Seattleではこのマルチデバイス開発に強力な支援機能が追加され、開発時の特定デバイス画面だけでなく、対応する全てのデバイス画面を一度にプレビュー・比較しながら開発できる。

【対象デバイスの選択】

【マルチデバイスプレビュー画面】

作成済みビュー

- Windows デスクトップ
- OS X デスクトップ
- iPad
- iPhone 4 インチ
- Android 10 インチ タブレット
- Android 4 インチ スマートフォン
- Android 5 インチ スマートフォン
- Android 7 インチ タブレット

使用可能なプレビュー

- Surface Pro
- iPhone 3.5 インチ
- iPhone 4.7 インチ
- iPhone 5.5 インチ
- Android 3.5 インチ スマートフォン
- Google Glass
- Samsung Galaxy S2
- Samsung Galaxy S4
- Samsung Galaxy Tab 7.0

すべてチェック

完了

マルチデバイスプレビュー

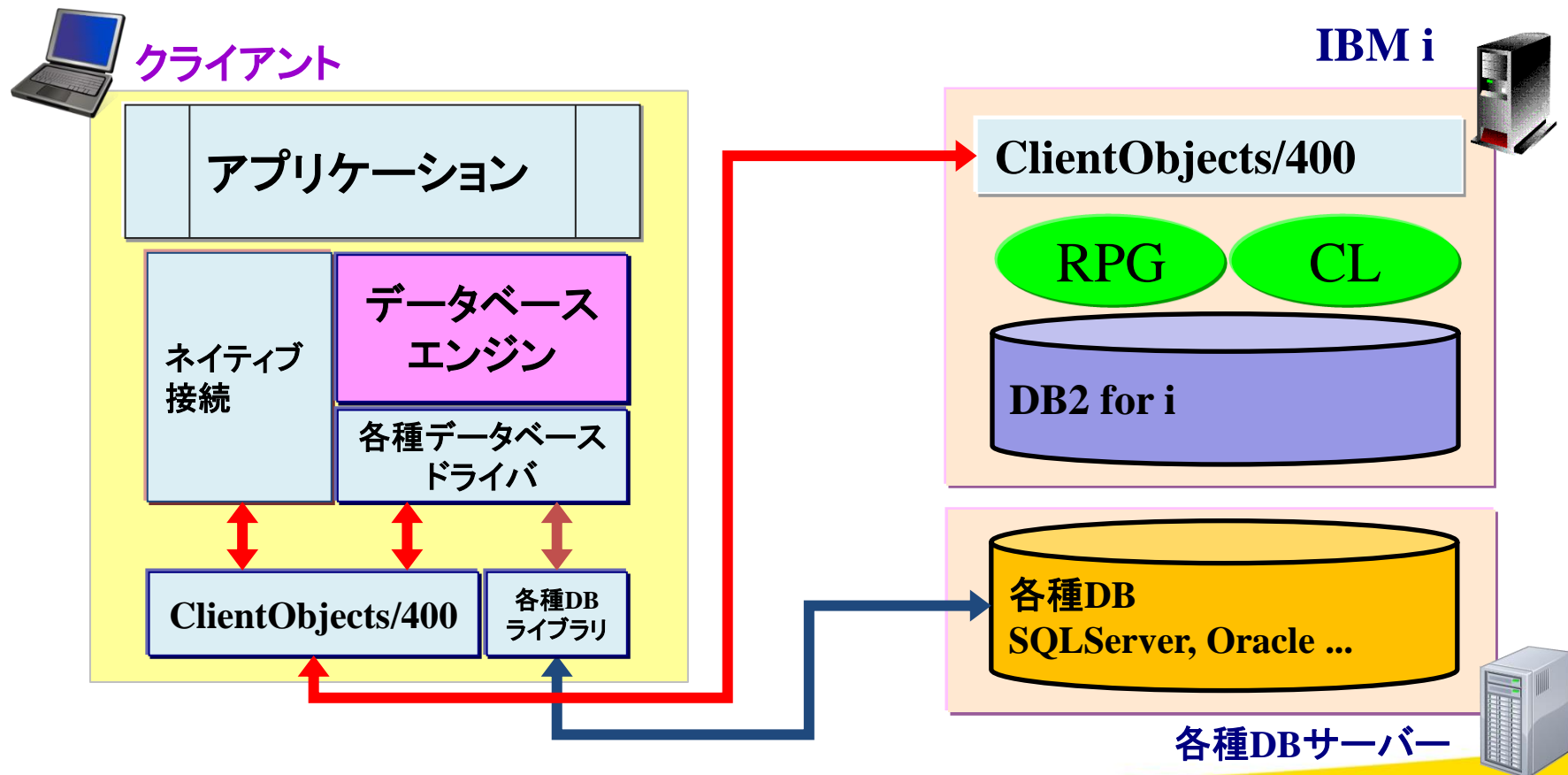
Windows デスクトップ iPad iPhone 4 インチ Android 10 インチタブレット

開発中に選択したデバイスの画面をプレビューで比較

Project1.dproj - プロジェクト マネージャ | モデル ビュー | テータ エクスプローラ | マルチデバイス プレビュー

## ■ 2-4.FireDACを使ったIBM i 接続機能

- データベースエンジン (dbExpress や BDE)  
DelphiはデータベースエンジンによってIBM i 以外のDBに対しても同手法のプログラミングでマルチDB対応の開発が可能。



## ■ 2-4.FireDACを使ったIBM i 接続機能

### • FireDACとは？

FireDACはdbExpressやBDEに続く、新しいデータベースエンジン



#### 【特徴】

- ・Windows 32bit/64bit対応
- ・FireMonkey対応
- ・BDE Administratorのような初期インストールは不要
- ・双方向データセット形式

	FireDAC	dbExpress	BDE
Windows32bit	○	○	△
Windows64bit	○	○	×
FireMonkey	○	○	×
初期インストール	不要	不要	必要
データセット形式	双方向	単方向	双方向
速度	○	△	○

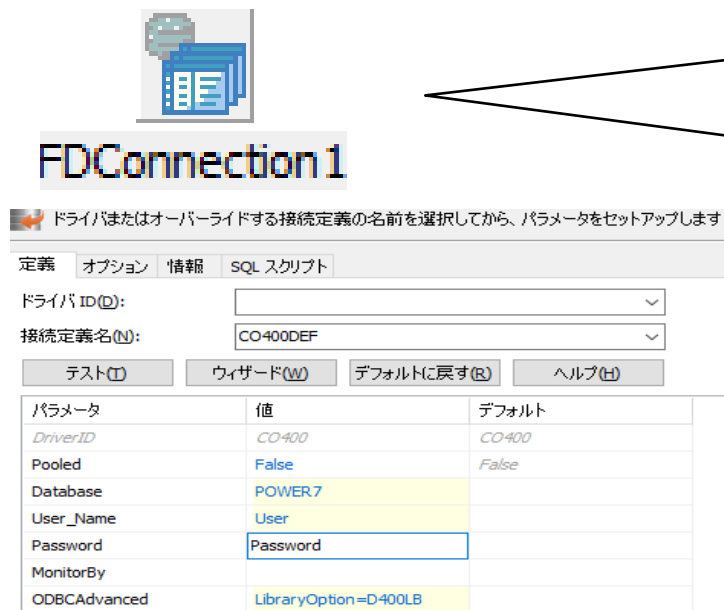
※dbExpressの速度は単方向でClientDataSetの使用率が高いため△としています。



## ■ 2-4.FireDACを使ったIBM i 接続機能

### FireDACの基本コンポーネント

- TFDConnectionコンポーネント



IBM i への接続を制御するコンポーネント  
ダブルクリックすることで専用エディタから接  
続の設定が可能

- TFDPhysCO400DriverLinkコンポーネント



TFDConnectionにDelphi/400のIBM i 用  
ドライバ情報を提供するコンポーネント

## ■ 2-4.FireDACを使ったIBM i 接続機能

- TFDTableコンポーネント



FDTable1

単一のファイルを指定して  
データを取得・操作するコンポーネント

- TFDQueryコンポーネント



FDQuery1

SQLを実行してデータを取得・操作するコンポーネント

- TFDGUIxWaitCursorコンポーネント



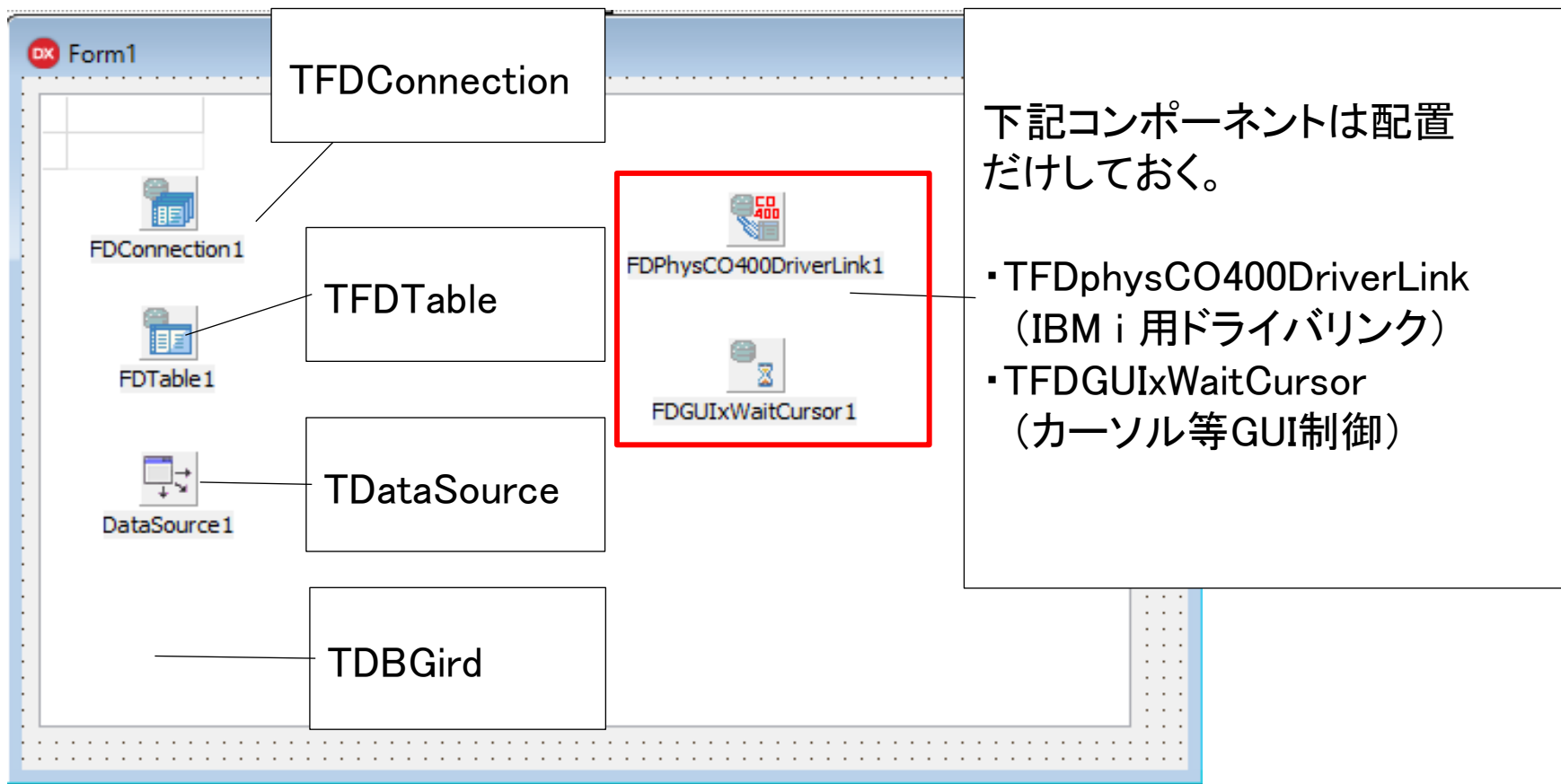
FDGUIxWaitCursor1

待機カーソルなどの制御用コンポーネント

## ■ 2-4.FireDACを使ったIBM i 接続機能



- FireDACのIBM i 接続実装手順①  
コンポーネントの配置



## ■ 2-4.FireDACを使ったIBM i 接続機能

### • FireDACのIBM i 接続実装手順②

TFDConnecitonの設定



FDConnection1

ダブルクリックで右の  
接続エディタ画面が起動

ドライバまたはオーバーライドする接続定義の名前を選択してから、パラメータをセットアップします

定義 オプション 情報 SQL スクリプト

接続定義名に  
CO400DEFを選択

ドライバ ID(D):

接続定義名(N):

CO400DEF

テスト(T)

ウィザード(W)

デフォルトに戻す(R)

ヘルプ(H)

パラメータ	値	デフォルト
DriverID	CO400	
Pooled	False	
Database	POWER7	
User_Name	User	
Password	Password	
MonitorBy		
ODBCAdvanced	LibraryOption=D400LB	

DatabaseにConfigurationの  
AS/400 Name (接続先)を設定

User\_Name/Passwordに  
アカウント情報を設定

ライブラリを指定する場合は、ODBCAdvancedに  
LibraryOption = ライブラリ名 で設定

## ■ 2-4.FireDACを使ったIBM i 接続機能

### • FireDACのIBM i 接続実装手順② 補足

TFDConencitonの設定

前ページのODBCAdvancedパラメータには、次のオプション指定も可能  
(複数指定する場合は ; で連結記述)

#### 【設定値】

- ライブラリの指定 : LibraryOption=ライブラリ名
- 区切り文字の指定 : Namingconvention=  
(区切り文字は . または / デフォルトはスラッシュ)
- トランザクション指定 : Commitment=トランザクションレベル  
(トランザクションレベルは \*NONE、\*CHG、\*CS、\*ALL)
- バッファカウント指定 : BUFFERCOUNT=バッファ数  
(20~100を目安に指定。大量データ転送時に有効)

## ■ 2-4.FireDACを使ったIBM i 接続機能

### • FireDACのIBM i 接続実装手順③

TFDTableの設定



FDTable1

TableNameプロパティに対象のファイルを設定。  
TFDConnectonでライブラリを指定している場  
合はリストから選択。

TFDQueryを使う場合は、  
SQLプロパティにSQLを記述。



## ■ 2-4.FireDACを使ったIBM i 接続機能

### • FireDACのIBM i 接続実装手順④

TDBGridへの表示設定



DataSource1

【TDataSource】  
DataSetプロパティに  
FDTable1をセット



【TDBGrid】  
DataSourceプロパティに  
atasource1をセット



## ■ 2-4.FireDACを使ったIBM i 接続機能

- FireDACのIBM i 接続実装手順⑤  
データへのアクセス

The screenshot shows the Delphi IDE with the FireDAC table properties window open. The 'Active' property is set to 'True'. A callout box points to this property with the text: 【TFDTable】 ActiveプロパティをTrueにしてデータをOpen. The background shows a data table with columns: COMPANY, ADDR1, and ADDR.

COMPANY	ADDR1	ADDR
コナツマリンショップ2	大島町4-976-321	東京
ダイブハウススタートル5	東荻5-8-7	
ダイビングベース新井8	新井2-14	新井
アリアダイビングセンター	FDPPhysCO400DriverLink1	
1351 亀山ダイブセンター	稲毛区亀山町632-1	稲毛
ダイブショップブルーリーフ	鯖松町23-738	鎗3-1
FDTable1 MHMダイバーズクラブ	埴輪町32	
1984 ADVENTURE UNDERSEA	POFDGUIxWaitCursor 1	
211R グリーンスポーツクラブ	中海老町633-21	東渡
パイナッブルダイバーズ	中村1455-1	
DataSource コーンハウスペンギン	出井里246-35	
2163 SCUBA HEAVEN	PO BOX 15814	
2165 SHANGRI-LA SPORTS CENTER	PO BOX 48644	
2315 ダイビングショップマーメイド	富ヶ谷3-21-4	鍋田
2353 マリンセンター楽柿浦	西区楽柿浦5-6-11	



## ■ 2-4.FireDACを使ったIBM i 接続機能

- FireDACのIBM i 接続実装手順⑥  
コンパイルと実行

CUSTNO	COMPANY	ADDR1	ADDR
1221	ココナッツマリンショップ2	大島町4-976-321	東京
1513	ダイブハウススタートル5	東荻5-8-7	
3444	ダイビングベース新井8	新井2-14-3	新井:
1231	アクアダイビングセンター	明太区曾根541	
1351	亀山ダイブセンター	稲毛区...町632-1	稲毛[
1380	ダイブショップブルーリーフ	鯖松	鎗3-1
1384	MHMダイバーズクラブ	埴輪	
1984	ADVENTURE UNDERSEA		
2118	グリーンスポーツクラブ		
2135	パイナップルダイバーズ		
2156	マリンハウスぺんぎん		
2163	SCUBA HEAVEN		
2165	SHANGRI-LA SPORTS CENTER		
2315	ダイビングショップマーメイド		
2353	マリンセンター楽柿浦		

これだけの手順で FireDACでIBM i に接続する照会アプリケーションが完成！



## ■ 2-4.FireDACを使ったIBM i 接続機能

### • FireDACのパフォーマンス

<計測前提環境>

IBM i :V7R1

対象ファイル: 1Record (30Field/350Byte) を350万Records保有

計測実行PC: Windows10 64bit

<実行結果: 30回計測の平均>

SQLで100件のSELECT取得速度		
BDE	dbExpress	FireDAC
0.7 秒	0.6 秒	0.6 秒












SQL+ClientDataSetで10万件のキャッシュ速度		
BDE	dbExpress	FireDAC
80 秒	352 秒	71 秒

※上記当社環境での計測結果の平均値  
なのであくまで目安とお考え下さい。

dbExpressで不得意としていた大量データの  
キャッシュ速度もFireDACではBDE以上の性能!

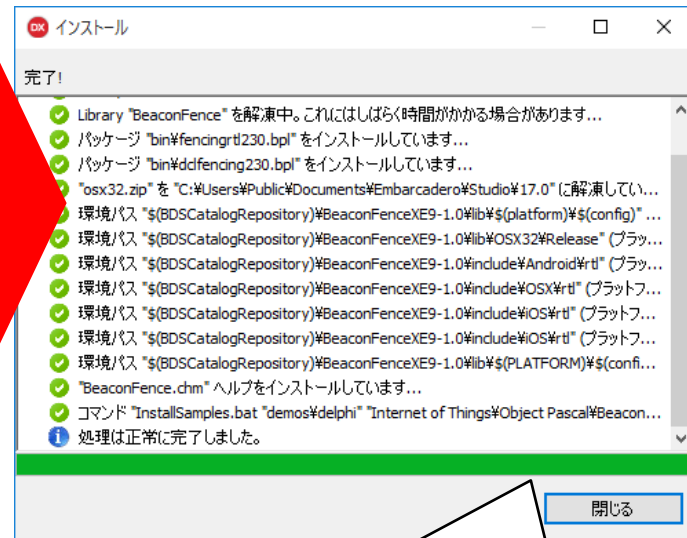
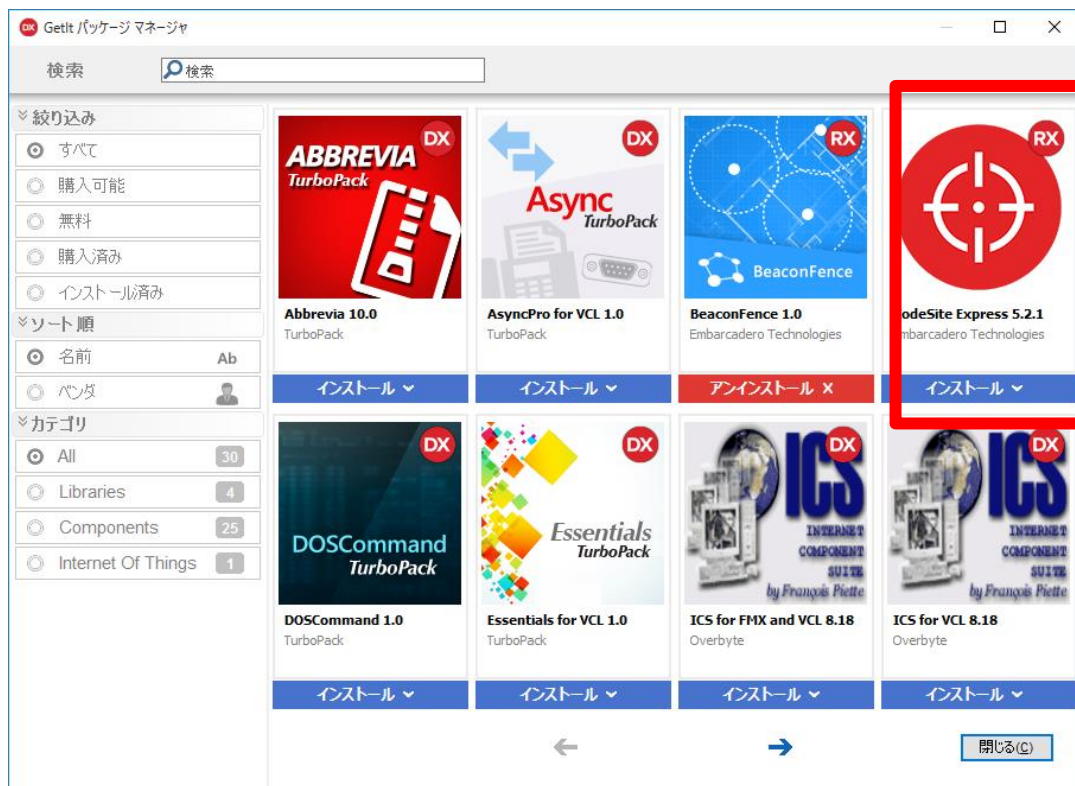
## ■ 2-4.FireDACを使ったIBM i 接続機能

- FireDACへ移行する場合の対応コンポーネント

	【FireDAC】	【dbExpress】	【BDE】	
DB接続制御用 コンポーネント	 FDConnection1	 SQLConnection1	 Database1	置換
データアクセス用 コンポーネント	 FDQuery1   FDTable1	 SQLQuery1   SQLTable1	 Query1   Table1	
データキャッシュ 用コンポーネント (共通)		 DataSetProvider1	 ClientDataSet1	変更不要

## ■ 2-5.IDEの機能強化

- GetItパッケージマネージャ  
10 Seattleでは[ツール|GetItパッケージマネージャ]から  
サードパーティ製品のインストール/アンインストールが簡単に可能



選択するだけで  
自動インストールや  
自動アンインストールが可能

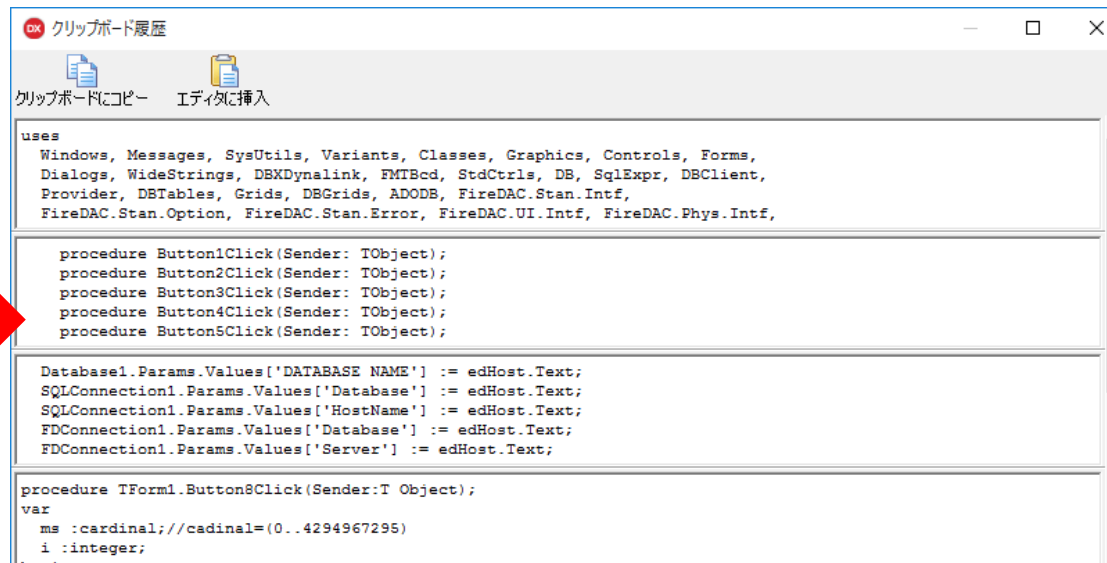
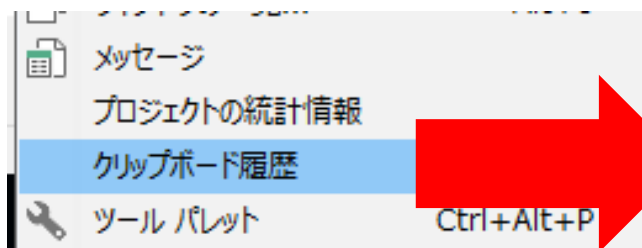
## ■ 2-5.IDEの機能強化

### • IDEの性能改善

IDE自体が大きく改善され、利用可能メモリが約2倍に拡張。  
大規模なプログラムでも安定したIDE環境で開発が可能。

### • クリップボード履歴機能

[表示|クリップボード履歴]より、  
クリップボードにコピーした  
16回分の履歴を再利用可能。



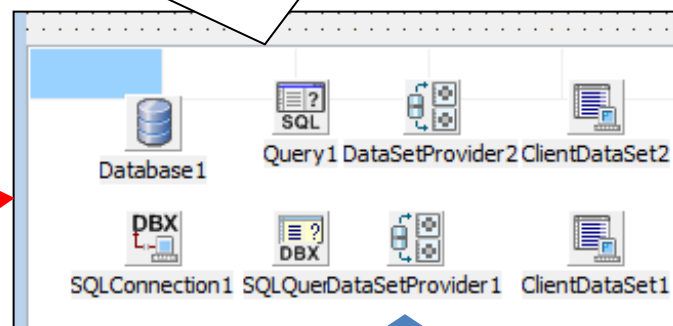
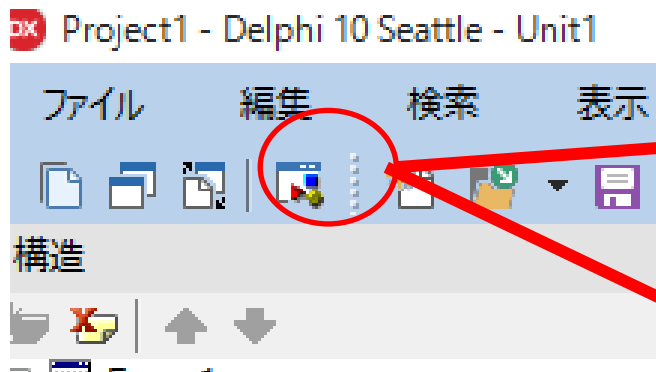
## ■ 2-5.IDEの機能強化

### • 非ビジュアルコンポーネントの表示切替

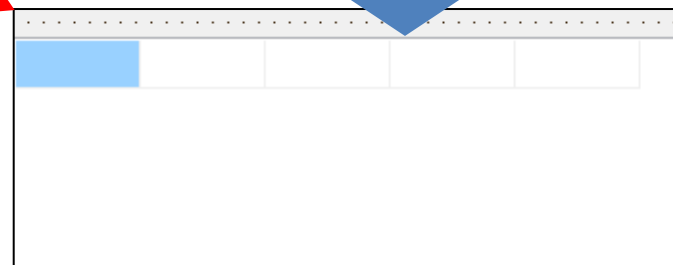
非ビジュアルコンポーネント

(DBコンポーネント等、画面に表示されないもの)  
をメニューから表示/非表示を切り替え可能

QueryやTable等を非表示にして画面デザインを整理！



切替



## 3.まとめ



## ■ 3.まとめ

新バージョン『Delphi/400 10 Seattle』では

- Windows10が正式にサポートされ、専用のユーザーインターフェースに対応したコンポーネントも使用できる
- IoTに対応する機能の1つとして、Beaconに対応するアプリケーションも開発することができる
- マルチデバイス開発では、複数デバイス画面を一度に把握できるプレビュー機能で開発効率がアップ
- FireDACにDelphi/400が対応したことで、よりハイパフォーマンスなIBM i アプリケーションが実現できる
- IDEの改善や補助機能の追加でさらに開発が効率的になっている



**ご清聴ありがとうございました**