

【セッションNo. 2】

Delphi/400 テクニカルセッション

開発者が知りたい実践プログラミングテクニック！

株式会社ミガロ.

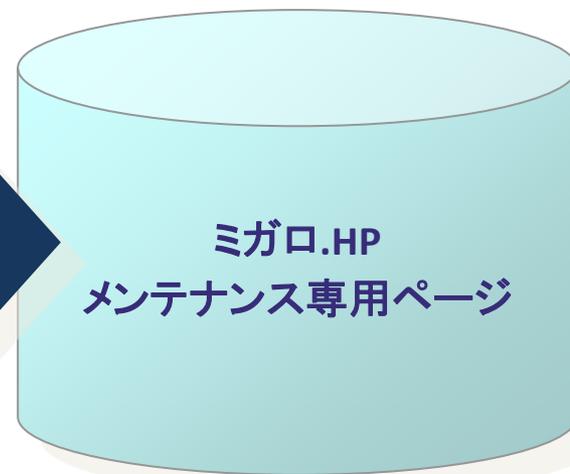
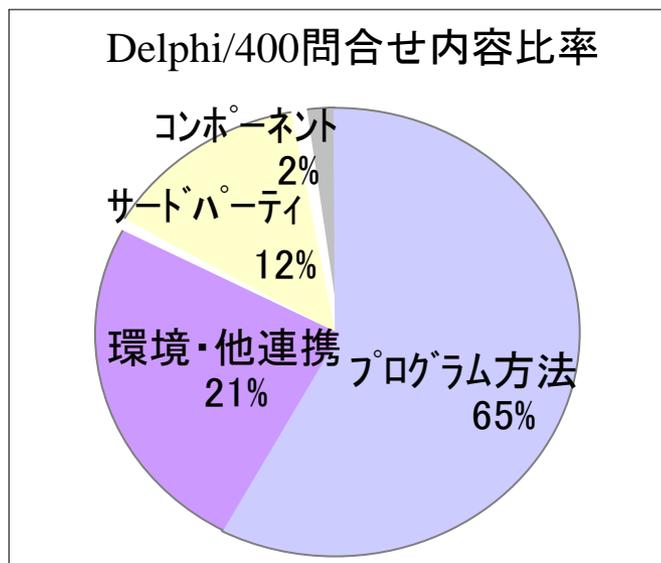
RAD事業部 技術支援課

吉原 泰介

■ Delphi/400テクニカルサポートより 現場で役立つ実践プログラムテクニックをご紹介します



サポートには月100件弱、年間1000件以上のお問合せ



本セッションでは、Delphi/400で開発できる
C/Sアプリケーション、Webアプリケーション、モバイルアプリケーションなど、
様々な形式毎にピックアップした有用な技術テクニックをご紹介します。

【アジェンダ】

1. パスワード期限切れ時のログイン制御 【VCL】

2. SQLを使ったレコード番号採番 【VCL】

3. セッションタイムアウト制御 【IntraWeb】

4. GPSマップ連携 【FireMonkey】

5. 他アプリの起動連携 【FireMonkey】



【実践テクニック1】 パスワード期限切れ時のログイン制御

1.パスワード期限切れ時のログイン制御【VCL】

- ユーザープロファイルのパスワード期限(5250)
IBM i にログインするユーザープロファイルは
パスワードに期限が設定されているとログイン時に変更が求められる。

The image shows two screenshots of an IBM i terminal session. The top screenshot shows the user profile information for user D4001, including system (MIGAROP7), subsystem (QBASE), and display device (QPADEV000Q). The bottom screenshot shows the password change prompt, including the current password, new password, and confirmation password. A red arrow points from the first screenshot to the second. A callout box points to the second screenshot with the text "パスワード変更画面".

セッション A - [24 x 80]
ファイル(F) 編集(E) 表示(V) 通信(C) 777777(A) ウィンドウ(W) ヘルプ(H)
ホスト: 192.168.0.21 ポート: 23 ワークステーション ID: 切断

システム : MIGAROP7
サブシステム : QBASE
表示装置 : QPADEV000Q

ユーザー : D4001
パスワード :
プログラム/プロシージャ :
メニュー :
現行ライブラリー :

セッション A - [24 x 80]
ファイル(F) 編集(E) 表示(V) 通信(C) 777777(A) ウィンドウ(W) ヘルプ(H)
ホスト: 192.168.0.21 ポート: 23 ワークステーション ID: 切断

ユーザー・プロファイル : D4001
パスワード最終変更日時 : 16/10/25
選択項目を入力して、実行キーを押してください。
現在のパスワード :
新しいパスワード :
新しいパスワード (確認用) :

F3= 終了

パスワード変更画面

1.パスワード期限切れ時のログイン制御【VCL】



- ユーザープロファイルのパスワード期限 (Delphi/400)
このパスワード期限に伴うパスワード変更は
Delphi/400のログイン機能でも同じように制御される。

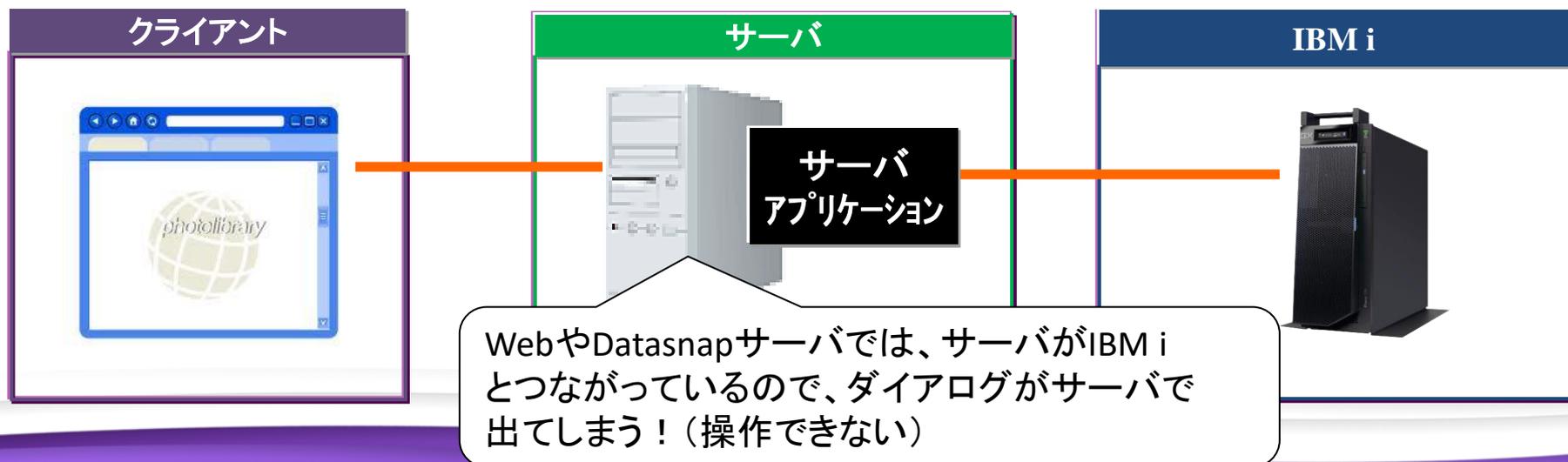
The image shows two screenshots of Delphi/400 dialog boxes. The left dialog, titled 'AS/400へのサイン・オン', contains fields for 'システム: POWER8', 'ユーザーID: D4001', and 'パスワード: ****'. It has 'OK' and 'キャンセル' buttons. A callout box points to it with the text 'ログインダイアログ (プログラム内でログインしても同じ)'. A large red arrow points from this dialog to the right dialog, titled 'パスワード変更'. The right dialog has fields for '新しいパスワードを入力:', 'パスワード:', and 'パスワードの確認:'. It also has 'Ok' and 'キャンセル' buttons. A callout box points to it with the text 'パスワード変更ダイアログが自動表示'.

1.パスワード期限切れ時のログイン制御【VCL】

- ただしシチュエーションによっては、パスワード変更のダイアログを見せたくない場合もある。
例えば・・・

ユーザーに直接パスワードを変更させたくない！

サーバアプリケーションでダイアログを出したくない！



1.パスワード期限切れ時のログイン制御【VCL】

- パスワード変更ダイアログの制御方法
期限で表示されるパスワード変更ダイアログはサインオンの結果を5250同様に自動応答している為、**プログラムでは制御することはできない。**

ただし、環境設定でこの応答を事前に変更しておくことが可能
環境設定ファイルは**Delphi/400のバージョンによって異なる（下記）**
ので注意が必要。

【Delphi/400 XE以前】

C:¥Windows¥win.ini

【Delphi/400 XE3以降】

C:¥CO4XX¥CO400.ini

※CO4XXはバージョンによって異なる。

上記環境設定ファイルの[CO400]セクションに下記を追記するとダイアログを出さずにエラーとすることができる。

TCPSCREEN=N

1.パスワード期限切れ時のログイン制御【VCL】

- 応答動作を環境設定ファイルで制御

AS/400へのサイン・オン

システム: POWER8

ユーザーID:

パスワード:

ログイン結果を環境設定ファイルで
応答制御する

TCPSCREEN=Y or 設定なし

パスワード変更

新しいパスワード入力

パスワード:

パスワードの確認:

TCPSCREEN=N

Project14

ローカル名が必要です。

明示的なエラーで、変更をさせない。

1.パスワード期限切れ時のログイン制御【VCL】



• プログラミング制御例

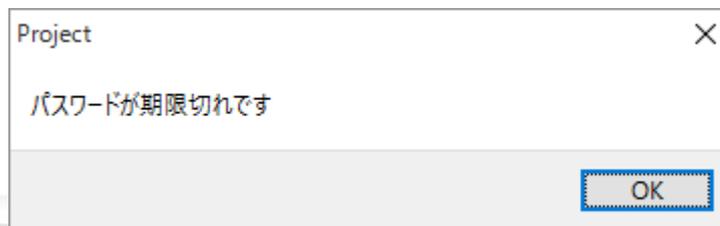
接続処理ロジック

usesにScderrsを追加

```
try
  AS4001.Connect;
except
  on e: EComError do
  begin
    if e.ErrorCode = 140 then
    begin
      ShowMessage('パスワードが期限切れです');
    end;
  end;
end;
```

エラーコードを140でチェックすれば
パスワードエラーを判断できる。

アプリケーションとしてのエラーに
メッセージを置き換えることが可能



【実践テクニック2】 SQLを使ったレコード番号採番

2. SQLを使ったレコード番号採番【VCL】

- SQLで取得したレコードの番号

Query系のコンポーネントを使うとSQLでIBM i のデータを取得することができるが、データによってはコード（例えば商品CD）を持っていてもレコード番号をデータとして持たない場合がある。

例) 商品マスタ

PRODCD	PRODNM	PROZKS
PJ00000001	い・ろ・は・す	300
PK00000002	ボルヴィック	150
PC00000003	エビアン	80
PD00000004	クリスタルガイザー	90
PF00000005	おいしい水	120
PE00000006	コントレックス	110
PG00000007	南アルプス天然水	150
PH00000008	サントリー天然水	80

商品CDは持っているがアプリケーションでは**レコード番号(順番)**で画面表示や帳票印刷の明細番号に使いたい！

2. SQLを使ったレコード番号採番【VCL】

- SQLで取得したレコードの番号
例えば条件をつけてデータを抽出した場合、
連番のコード（商品コード）でもレコード番号としては使用できない。

PRODCD	PRODNM	PROZKS
PC00000003	エビアン	80
PD00000004	クリスタルガイザー	90
PF00000005	おいしい水	120
PE00000006	コントレックス	110
PH00000008	サントリー天然水	80

【条件付きSQLで抽出】

```
SELECT PRODCD,PRODNM,PROZKS FROM PRODUCT WHERE PROZKS < 150
```

2. SQLを使ったレコード番号採番【VCL】

- RPGでレコード番号を付与する

【対応方法1】

ひとつの方法としてRPGで抽出結果をワークファイルに出力して、レコード番号の項目を追加する。

商品マスタ

ワークファイル



RPG
プログラム

NO	PRODCD	PRODNM	PROZKS
1	PJ00000001	い・ろ・は・す	300
2	PK00000002	ボルヴィック	150
3	PC00000003	エビアン	80
4	PD00000004	クリスタルガイザー	90
5	PF00000005	おいしい水	120
6	PE00000006	コントレックス	110
7	PG00000007	南アルプス天然水	150
8	PH00000008	サントリー天然水	80

ワークファイルに新項目としてレコード番号を持たせる。
SQLはこのワークファイルにアクセス

2. SQLを使ったレコード番号採番【VCL】



- SQLでレコード番号を付与する
【対応方法2】

SQL文上でROW_NUMBER関数を使ってレコード番号項目を作成する。

NO	PRODCD	PRODNM	PROZKS
1	PC00000003	エビアン	80
2	PD00000004	クリスタルガイザー	90
3	PE00000006	コントレックス	110
4	PF00000005	おいしい水	120
5	PH00000008	サントリー天然水	80

【ROW_NUMBER関数を使ったSQL】

```
SELECT CAST(ROW_NUMBER() OVER(ORDER BY PRODCD) AS INTEGER) NO  
PRODCD,PRODNM,PROZKS FROM PRODUCT WHERE PROZKS < 150
```

2. SQLを使ったレコード番号採番【VCL】

- SQLでレコード番号を付与する

ROW_NUMBER関数 構文

CAST(ROW_NUMBER()

OVER(ORDER BY 対象フィールド) AS INTEGER)

【SQLでレコード番号を付与するメリット】

実現できる結果としては、RPGのワークファイルと同じだが、開発する上で、RPGやワークファイルの仕組みを都度作成する必要がない(=開発効率アップ！)

2. SQLを使ったレコード番号採番【VCL】

- SQLでレコード番号を付与する

サンプルのSQL実行例

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  FDQuery1.Close();
  FDQuery1.SQL.Clear;

  if ToggleSwitch1.State = tssOff then
  begin
    FDQuery1.SQL.Add(' SELECT PRODCD, PRODNM, PROZKS FROM PRODUCT WHERE PROZKS < 150' );
  end
  else
  begin
    FDQuery1.SQL.Add(' SELECT CAST(ROW_NUMBER() OVER(ORDER BY PRODCD) AS Integer) NO, ');
    FDQuery1.SQL.Add(' PRODCD, PRODNM, PROZKS FROM PRODUCT WHERE PROZKS < 150' );
  end;

  FDQuery1.Open();
end;
```

スイッチでROW_NUMBER関数
を含めるかを制御。

【実践テクニック3】 セッションタイムアウト制御

3.セッションタイムアウト制御【IntraWeb】

- Webアプリケーションのセッションタイムアウト
Webアプリケーションは、C/Sアプリケーションと違い、IBM i やWebサーバーと常時接続のジョブとはなっていない。一定時間操作や処理がないとWebサーバ接続（セッション）は切断される。



操作や処理がないとセッションが切断されてアプリも継続できない。

3.セッションタイムアウト制御【IntraWeb】

- Webアプリケーションのセッションタイムアウト

Webアプリケーションのセッションタイムアウトは大きく2つある。
どちらかのタイムアウトに達するとセッションは切断される。

①Webサーバ(IIS)側のタイムアウト設定

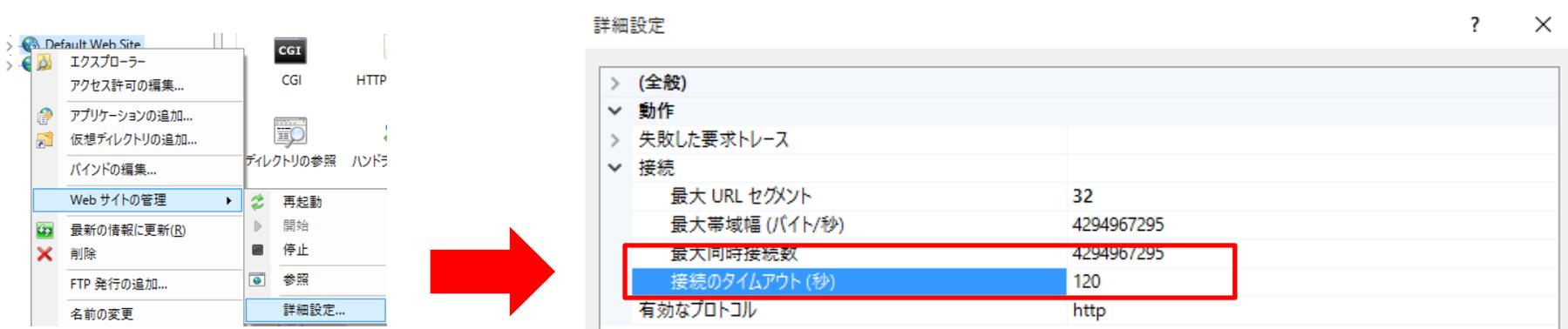
②Webアプリケーション側のタイムアウト設定

3.セッションタイムアウト制御【IntraWeb】

①Webサーバ（IIS）側のタイムアウト設定

IISのバージョンによっても設定箇所も異なるが、下記はIIS10の内容。

- ・ Webサイト管理の詳細設定にある接続タイムアウト値（画面は標準値）

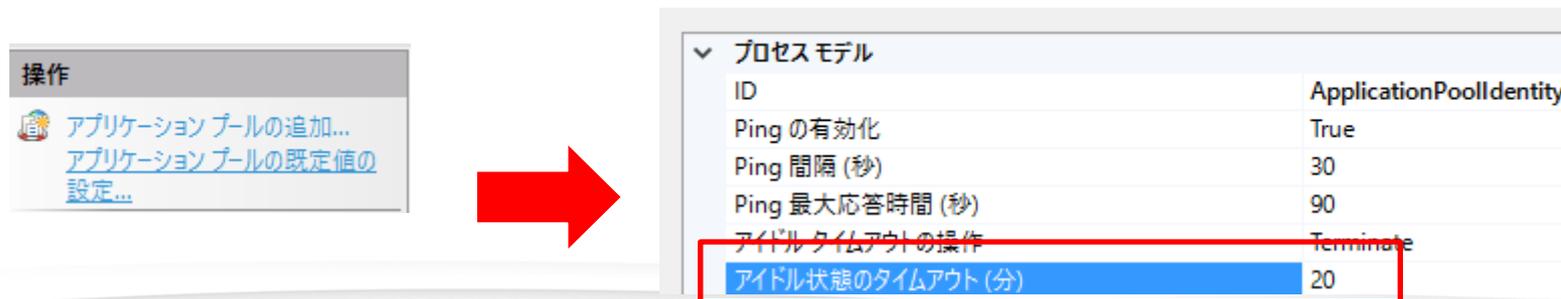


The screenshot shows the IIS Web Site Management console. On the left, the 'Web サイトの管理' (Web Site Management) menu is open, and the '詳細設定...' (Advanced Settings...) button is highlighted. A red arrow points to the '詳細設定' (Advanced Settings) window. In this window, the '接続' (Connections) section is expanded, and the '接続のタイムアウト (秒)' (Connection Timeout (seconds)) setting is highlighted with a red box, showing a value of 120.

設定項目	値
最大 URL セグメント	32
最大帯域幅 (バイト/秒)	4294967295
最大同時接続数	4294967295
接続のタイムアウト (秒)	120
有効なプロトコル	http

- ・ アプリケーションプールのアイドルタイムアウト値

アプリケーションプールの既定値



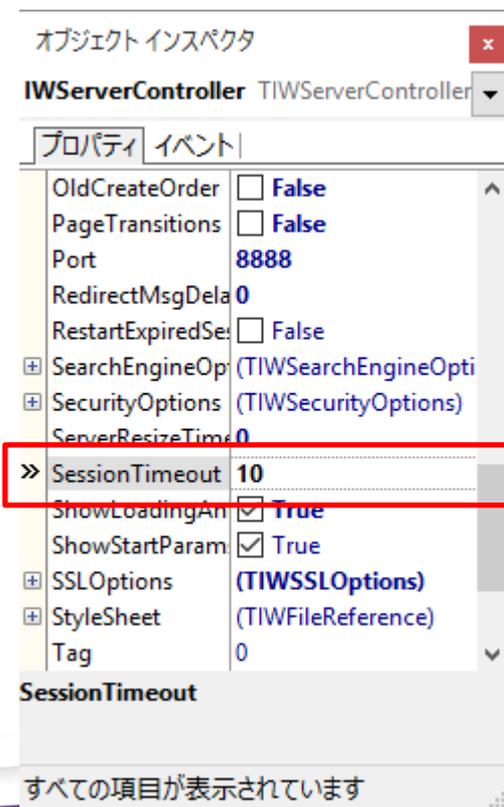
The screenshot shows the IIS Application Pools console. On the left, the 'アプリケーションプールの既定値の設定...' (Configure Application Pool Defaults...) link is highlighted. A red arrow points to the 'プロセスモデル' (Process Model) section of the application pool settings. In this section, the 'アイドル状態のタイムアウト (分)' (Idle Timeout (minutes)) setting is highlighted with a red box, showing a value of 20.

設定項目	値
ID	ApplicationPoolIdentity
Pingの有効化	True
Ping 間隔 (秒)	30
Ping 最大応答時間 (秒)	90
アイドルタイムアウトの操作	Terminate
アイドル状態のタイムアウト (分)	20

3.セッションタイムアウト制御【IntraWeb】

②Webアプリケーション側のタイムアウト設定

IntraWebアプリケーションプログラムにある
ServerControllerのSessionTimeOutプロパティ値（画面は標準値）



基本的にはIIS側に大きなタイムアウト値を設定し、アプリ側で小さいタイムアウト値を設定しておけばプログラムで制御しやすい。

①IISタイムアウト値 > ②Webアプリタイムアウト値

3.セッションタイムアウト制御【IntraWeb】

- IntraWebプログラムでの制御例（IW12以前）
IntraWebアプリケーションプログラムにある
ServerControllerのSessionTimeOutURLプロパティ値



URLか、htmlファイルを設定しておけば、
タイムアウト時に画面遷移する

IntraWeb14以降は大きく仕組みが変わる(次頁以降)

IntraWeb開発元Atozed社

<http://docs.atozed.com/docs.dll/development/Migrating%20to%20IntraWeb%20XIV.html>

3.セッションタイムアウト制御【IntraWeb】



- htmlソース例

IWSessionTimeOut.htmlとIWSessionInvalid.html どちらも同じ

```
<html>
<head>
<META http-equiv="content-type" content="text/html; charset=UTF-8">
<!-- <meta http-equiv="refresh" content="0.1;url=http://www.migaro.co.jp/"> -->
<META http-equiv="Content-Style-Type" content="text/css">
<title></title>
</head>
<body>
<a>Timeout</a>
</body>
</html>
```

タイムアウト時に呼び出されたら
画面を遷移する内容をhtmlに
記述しておく
(もしくはエラーメッセージなど)

セッションタイムアウトの基本はIWSessionTimeOut.htmlが使用されるが、Webサーバ上のセッションが認識できなくなると別の例外(Invalidエラー)が発生するため、それに対応したIWSessionInvalid.htmlも同じ内容で用意する

3.セッションタイムアウト制御【IntraWeb】

- 補足

タイムアウトの画面遷移で「http500」のエラーになる場合はIISから外部ドメインへのアクセス規制がされていることがある。その場合、web.configファイルで設定を変更して対応。
(※ web.configはIISの設定や制約が定義された環境ファイル)

IISのwwwrootにweb.config

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.webServer>
    <httpErrors existingResponse="PassThrough" /> //★
  <httpProtocol>
    <customHeaders>
      <remove name="X-UA-Compatible" />
      <add name="X-UA-Compatible" value="IE=7" />
    </customHeaders>
  </httpProtocol>
</system.webServer>
</configuration>
```

外部ドメインもエラーにしない

3.セッションタイムアウト制御【IntraWeb】



• デモ

ログイン中
株式会社ミガロ様
ログアウト

注文入力画面

会社: 株式会社ミガロ
ご担当者: 吉原 泰仁E
入先郵便番号: 556-0017
入先住所: 大阪市浪速区湊町2-1-57舞波サンケイビル13F
入先電話番号: 06-6251-8601 希望品日: 2012/11/21

注文明細金額 ¥133,140
送料 ¥1,050
合計金額 ¥134,190

指定したページに画面遷移

キーワードを入力

新着情報 - 製品情報 - システム構築 - セミナー&トレーニング - お問い合わせ - 導入事例 - 会社情報

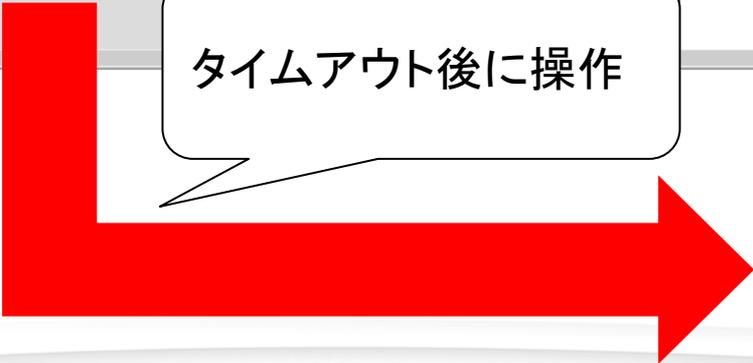
第19回
Delphi/400
テクニカルセミナー開催!
すぐに役立つテクニックから最新技術情報までご紹介

東京開催はコチラへ 大阪開催はコチラへ

製品のご紹介

Delphi/400 SP4i Business4Mobile

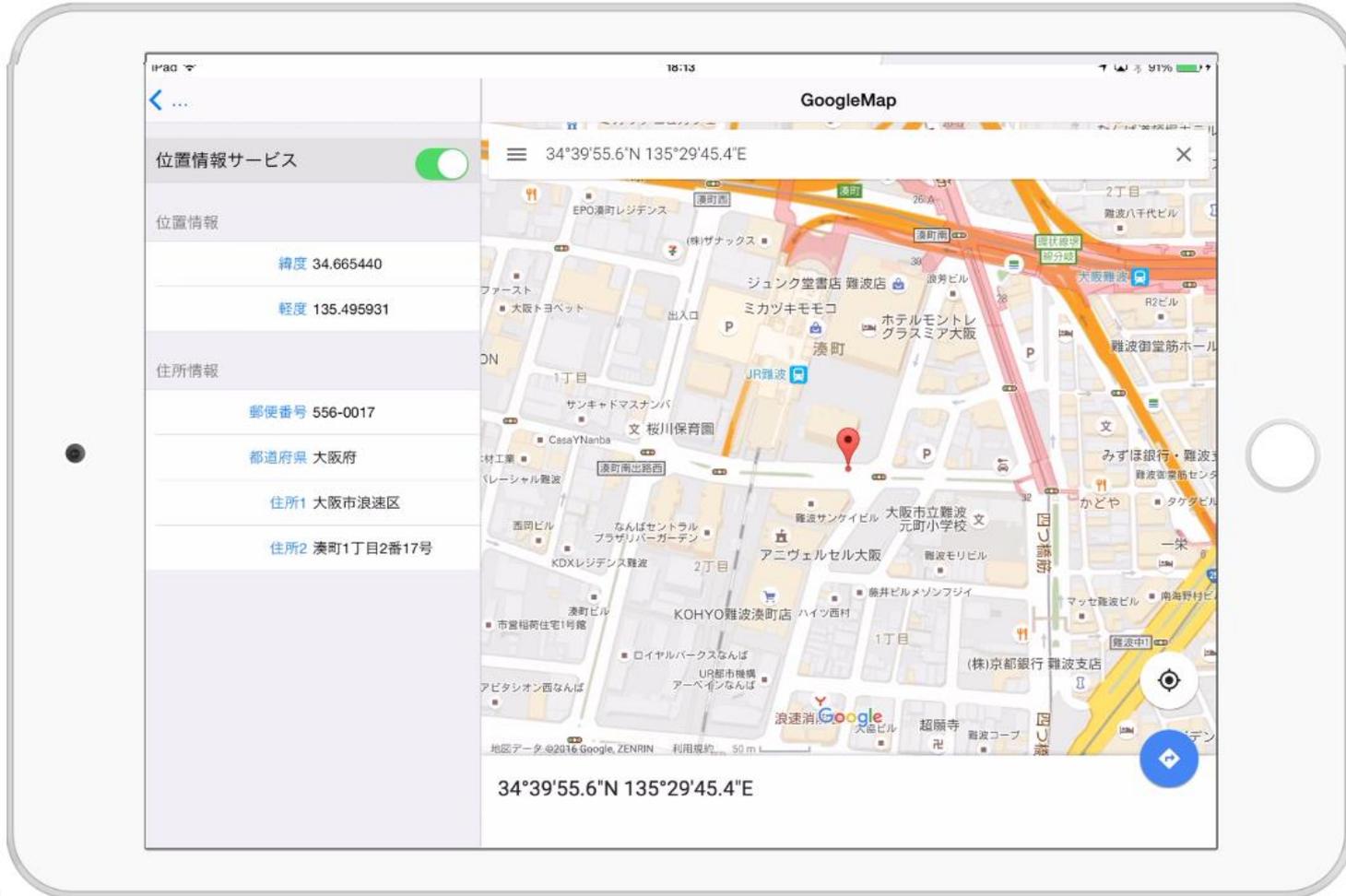
タイムアウト後に操作



【実践テクニック4】 GPSマップ連携

4.GPSマップ連携【FireMonkey】

- 現在位置をGoogleMAP表示する



4.GPSマップ連携【FireMonkey】

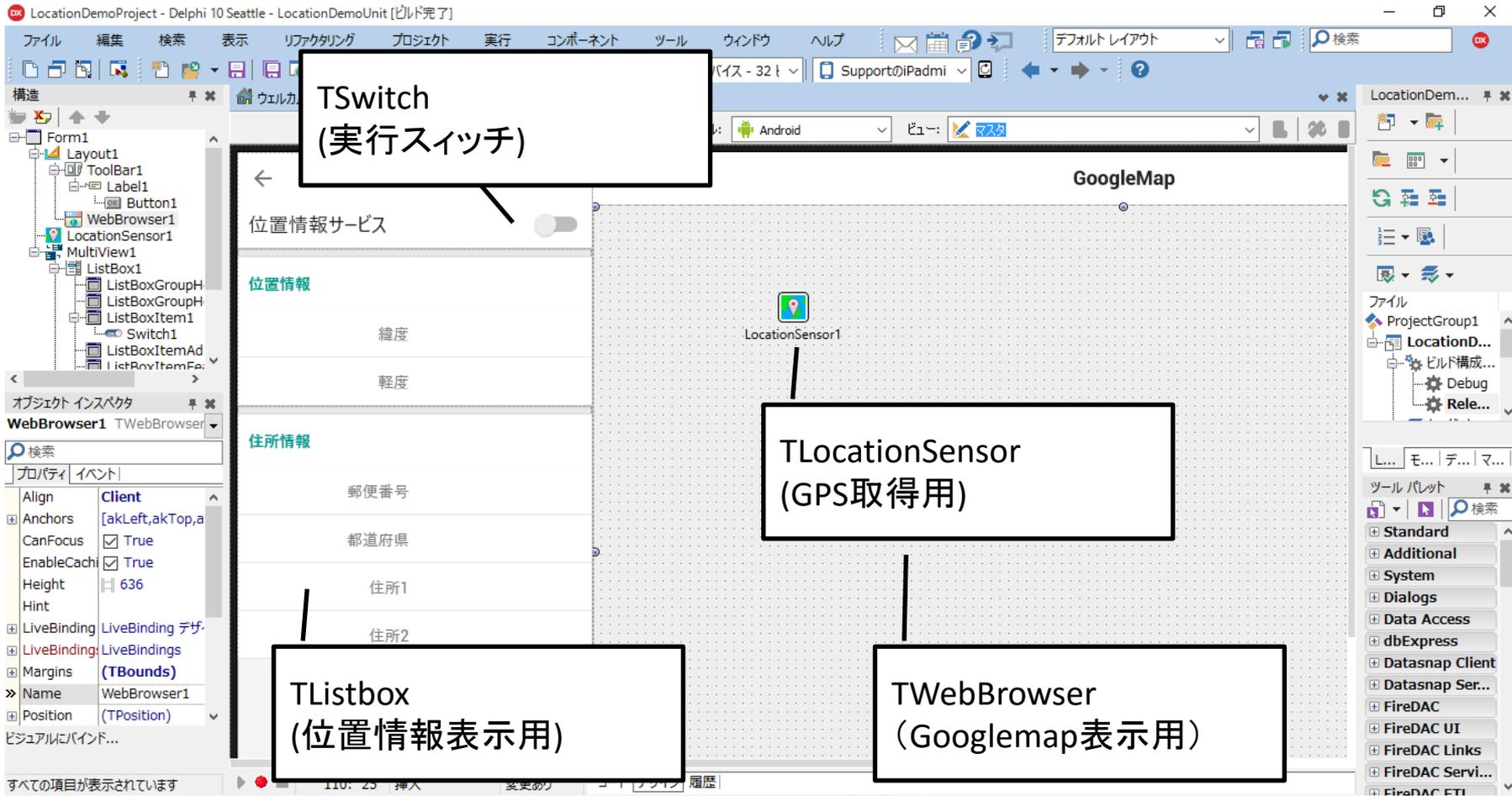
- 現在位置をGoogleMAP表示する

処理の大枠としてはデバイスのGPSで位置情報（緯度、経度）を取得して位置情報をGoogleMapのURLに渡してブラウザに表示する。



4.GPSマップ連携【FireMonkey】

- 現在位置をGoogleMAP表示する（設計）



4.GPSマップ連携【FireMonkey】

- 現在位置をGoogleMAP表示する（実装手順1）

定義

```
private
{ Private declarations }
FGeocoder: TGeocoder; //位置情報処理用(逆ジオコーディング用)
procedure OnGeocodeReverseEvent(const Address: TCivicAddress); //位置情報表示処理
```

ジオコーディングとは、住所、郵便番号などの地理データを地理座標に変換する処理。
逆ジオコーディングとは、地理座標をそれ以外の住所などの地理データに変換する処理。

位置情報表示処理

```
procedure TForm1.OnGeocodeReverseEvent(const Address: TCivicAddress);
begin
  ListBoxItemPostalCode.ItemData.Detail := Address.PostalCode; //郵便番号
  ListBoxItemAdminArea.ItemData.Detail := Address.AdminArea; //都道府県
  ListBoxItemLocality.ItemData.Detail := Address.Locality; //住所1
  ListBoxItemFeatureName.ItemData.Detail := Address.FeatureName; //住所2
end;
```

4.GPSマップ連携【FireMonkey】

- 現在位置をGoogleMAP表示する（実装手順2）

GPSイベント(位置情報変更処理)

```
procedure TForm1.LocationSensor1LocationChanged(Sender: TObject;
  const OldLocation, NewLocation: TLocationCoord2D);
var
  URLString: String;
  LSettings: TFormatSettings;
  LDecSeparator : Char;
begin //形式設定
  LDecSeparator := FormatSettings.DecimalSeparator;
  LSettings := FormatSettings;
  try
    FormatSettings.DecimalSeparator := '.';
    //現在位置情報の表示
    ListBoxItemLatitude.ItemData.Detail := Format('%2.6f', [NewLocation.Latitude]);
    ListBoxItemLongitude.ItemData.Detail := Format('%2.6f', [NewLocation.Longitude]);

    //GoogleMapのURLを生成
    URLString := Format('https://maps.google.com/maps?q=%2.6f,%2.6f',
      [ NewLocation.Latitude, NewLocation.Longitude]);
  finally
    FormatSettings.DecimalSeparator := LDecSeparator;
  end;
```

4.GPSマップ連携【FireMonkey】

- 現在位置をGoogleMAP表示する（実装手順3）

GPSイベント(位置情報変更処理) 続き

```
//WebブラウザへのGoogleMap表示
WebBrowser1.Navigate(URLString);

//逆ジオコーディング
try
  if not Assigned(FGeocoder) then
    begin
      if Assigned(TGeocoder.Current) then
        FGeocoder := TGeocoder.Current.Create;
      if Assigned(FGeocoder) then
        FGeocoder.OnGeocodeReverse := OnGeocodeReverseEvent;
    end;
except
  ListBoxGroupHeader1.Text := '取得エラー';
end;

//位置情報表示処理の実行
if Assigned(FGeocoder) and not FGeocoder.Geocoding then
  FGeocoder.GeocodeReverse(NewLocation);
end;
```

逆ジオコーディングで、地理座標をそれ以外の住所などの地理データに変換する処理。

逆ジオコーディングで、取得した位置情報を表示処理にまわす。

4.GPSマップ連携【FireMonkey】

- 現在位置をGoogleMAP表示する（実装手順4）

OnSwitchイベント(位置情報取得開始)

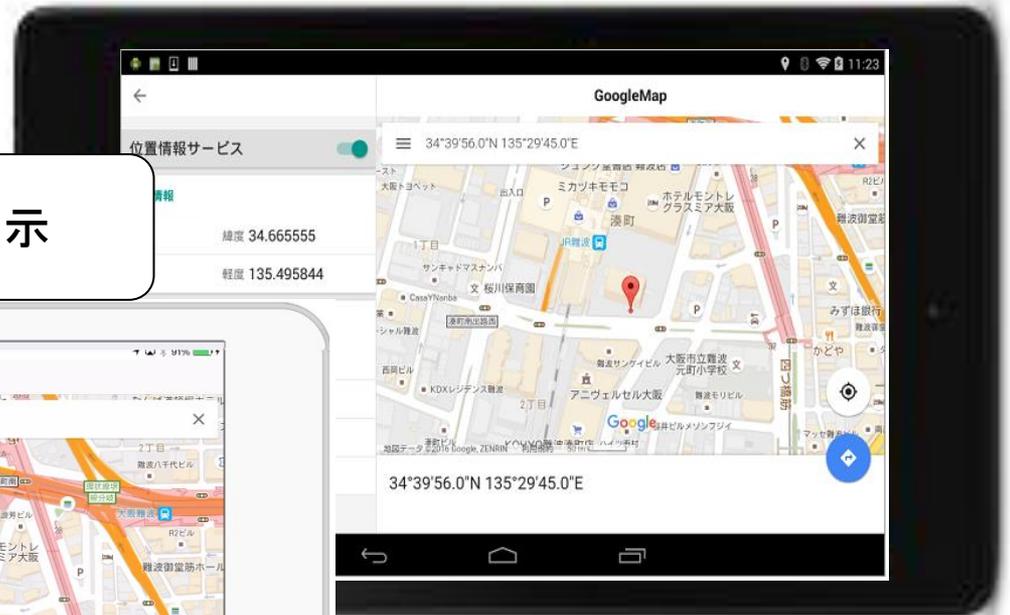
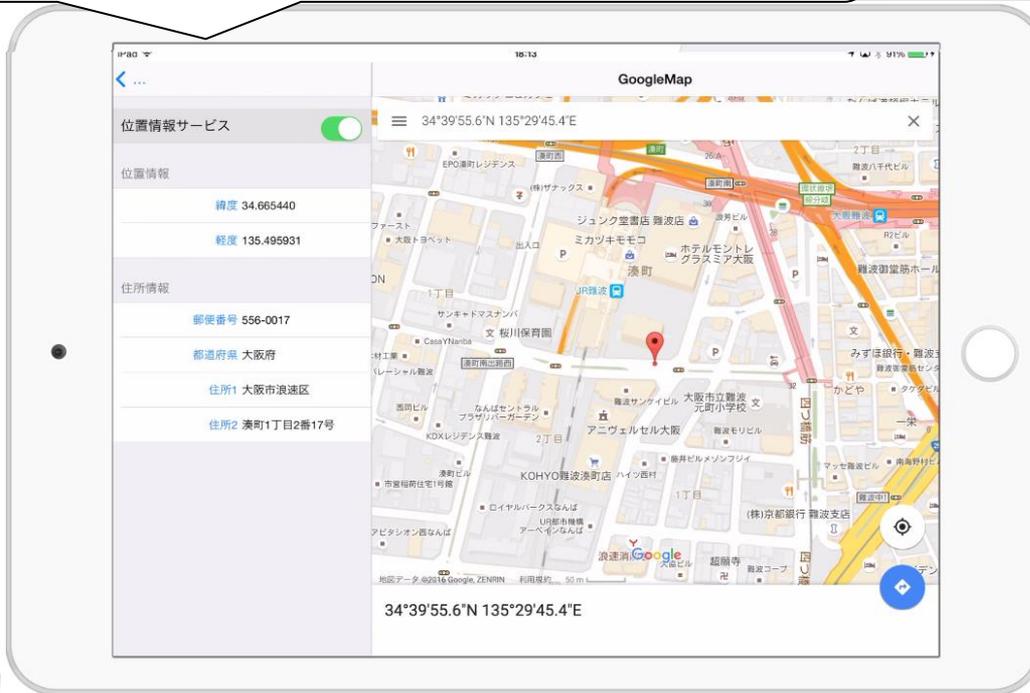
```
procedure TForm1.Switch1Switch(Sender: TObject);  
begin  
    LocationSensor1.Active := Switch1.IsChecked;  
end;
```

4.GPSマップ連携【FireMonkey】



- デモ（実装はiOSでもAndroidでも同じ）

スイッチを入れると位置情報と地図を表示



【実践テクニック5】 他アプリの起動連携

5.他アプリの起動連携【FireMonkey】

- モバイルアプリでの連携

Windows上のEXEアプリでは、ShellExecuteやCreateProcessを使って別のアプリを起動できる。

iOSやAndroidなどモバイルのアプリ上でも、別のアプリを起動することはできるが、呼び出される側のアプリにはURLスキームと呼ばれるインターフェースが用意されている必要がある。



例えばGoogleMapのアプリを呼び出すにはGoogleMapのURLスキームを呼び出す。

5.他アプリの起動連携【FireMonkey】

- URLスキームとは？

URL スキームとは、WebのURLのような形式で書かれているアドレス

例えばGoogleMapのiOSアプリであれば

comgooglemaps://



ミュージックのアプリであれば

music://



というURLスキームを持っており、これを呼び出すとGoogleMapやミュージックアプリが起動する。

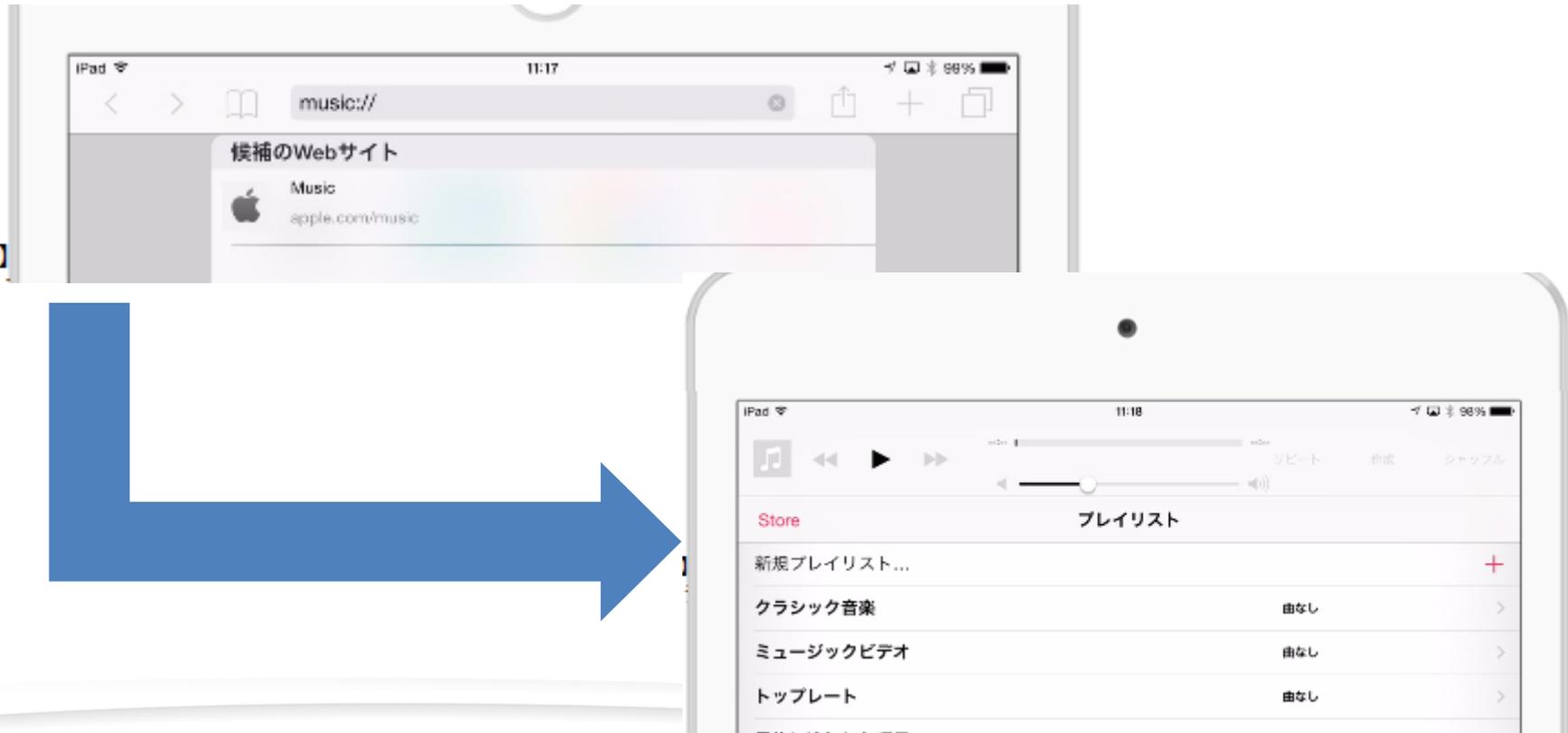
逆にURLスキームを持っていないアプリは呼び出すことができない

5.他アプリの起動連携【FireMonkey】

iOS

- URLスキームでアプリを呼び出してみる（ブラウザ）

例えばブラウザから「**music://**」をURLとして入力してもアプリを起動できる。



5.他アプリの起動連携【FireMonkey】

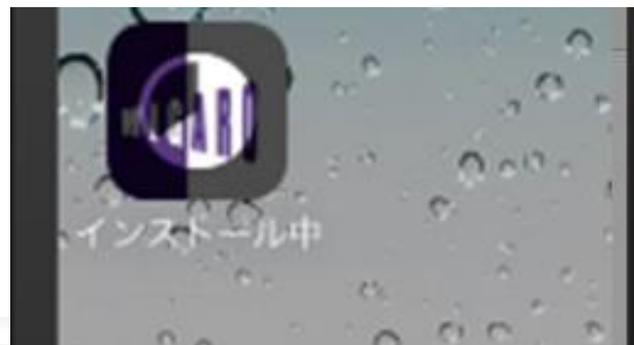
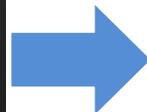
iOS

- URLスキーム（補足）

開発したiOSアプリをWebサーバからインストールする場合、html上で指定するリンクはURLスキームを使っている。

```
<h1>iOSダウンロードサイトサンプル</h1>
<form>
  <a href="itms-services://?action=download-manifest&url=https://Webサーバ /Project1.plist">
    アプリケーションダウンロード</a><br>
</form>
```

リンクの内容はURLスキームで指定する。



5.他アプリの起動連携【FireMonkey】

iOS

- URLスキームのいろいろ（例：iOS標準アプリ）

message://

• メール

mailto://

• メール(新規作成)

music://

• ミュージック

itms-apps://

• App Store

facetime://

• Facetime

itms://

• iTunes

x-web-search://

• Safari

ibooks://

• iBooks

x-apple-reminder://

• リマインダー

tel://電話番号

• 電話(確認をする場合はtelprompt:)

5.他アプリの起動連携【FireMonkey】

iOS



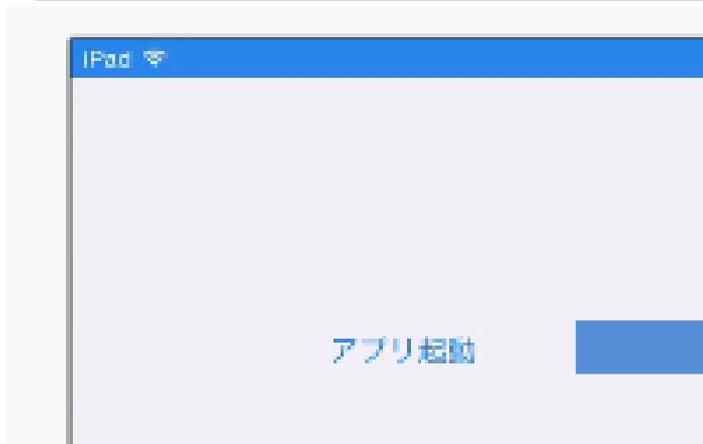
- URLスキームでアプリを呼び出してみる (プログラム)

OnCreateイベント(初期処理)

例)

uses[FMX.Helpers.iOS, Macapi.Helpersを追加

```
procedure TForm1.Button1Click (Sender: TObject);  
begin  
  SharedApplication.openURL(StrToNSUrl('comgooglemaps://'));  
end;
```



5.他アプリの起動連携【FireMonkey】

iOS

- 開発プログラムにURLスキームを含めるには？
通常のiOS開発アプリケーションにはURLスキームは含まれない。
URLスキームを定義するにはplist.infoファイルに直接追加する必要がある。

[プロジェクト|配置]から配置マネージャを起動して次の手順でplist.infoファイルを差替える。

- 1.オリジナルのinfo.plistファイルをコピーして編集（内容は次ページ）
- 2.配置マネジャーで編集したinfo.plistファイルを追加
- 3.オリジナルのinfo.plistファイルを配置マネジャーでチェックを外す
- 4.編集したinfo.plistファイルのリモート名をInfo.plistに変更。

ローカルパス	ローカル名	型	構成	プラットフォーム	リモート...	リモート名
<input checked="" type="checkbox"/> iOSDevice32¥Release¥	ResourceRules.plist	ProjectiOSD...	Release	[iOSDevice32]	¥	ResourceRules.plist
<input checked="" type="checkbox"/> iOSDevice32¥Release¥	LocationDemoProject.entitle...	ProjectIOSEn...	Release	[iOSDevice32]	../¥	LocationDemoProject.entit
<input checked="" type="checkbox"/> iOSDevice32¥Release¥	LocationDemoProject	ProjectOutput	Release	[iOSDevice32]	¥	LocationDemoProject
<input checked="" type="checkbox"/> iOSDevice32¥Release¥	LocationDemoProject.dSYM	ProjectiOSD...	Release	[iOSDevice32]	X6(D	LocationDemoProject
<input type="checkbox"/> iOSDevice32¥Release¥	LocationDemoProject.info.plist	ProjectiOSIn...	Release	[iOSDevice32]	¥	Info.plist
<input checked="" type="checkbox"/> iOSDevice32¥Release¥	LocationDemoProject.info2.plist	File	Release	[iOSDevice32]	¥	info.plist

5.他アプリの起動連携【FireMonkey】

iOS

- 開発プログラムにURLスキームを含めるには？

plist.Infoファイルに編集で追記する内容

```
<key>CFBundleURLTypes</key>
<array>
  <dict>
    <key>CFBundleTypeRole</key>
    <string>Editor</string>
    <key>CFBundleURLName</key>
    <string>TEST URL</string>
    <key>CFBundleURLSchemes</key>
    <array>
      <string>TEST</string>
    </array>
  </dict>
</array>
```

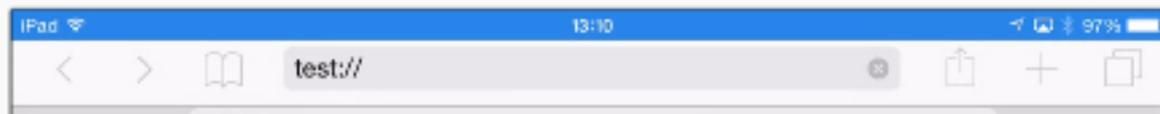
URLスキームを「TEST」と設定。

5.他アプリの起動連携【FireMonkey】

iOS

- URLスキームで開発アプリを呼び出す (ブラウザ)

ブラウザから「**TEST://**」を
URLとして入力すると開発アプリを起動できる。



5.他アプリの起動連携【FireMonkey】



- 開発アプリをURLスキームを呼び出す

OnClickイベント(連携起動処理)

例)

uses [iOSSapi.Foundation, FMX.Helpers.iOS, IdURI] を追加

```
procedure TForm1.Button1Click (Sender: TObject);
```

```
var
```

```
NSU: NSURL;
```

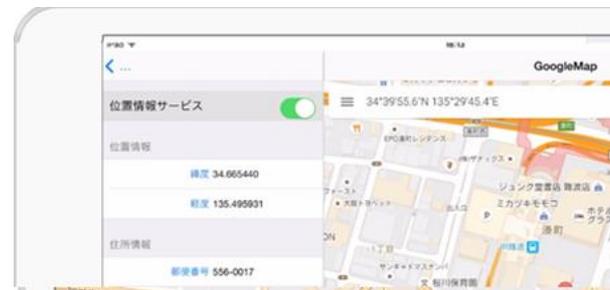
```
begin
```

```
NSU := StrToNSURL(('TEST://')); //ここに呼び出すURLリクエスト
```

```
if SharedApplication.canOpenURL(NSU) then
```

```
exit(SharedApplication.openURL(NSU));
```

```
end;
```

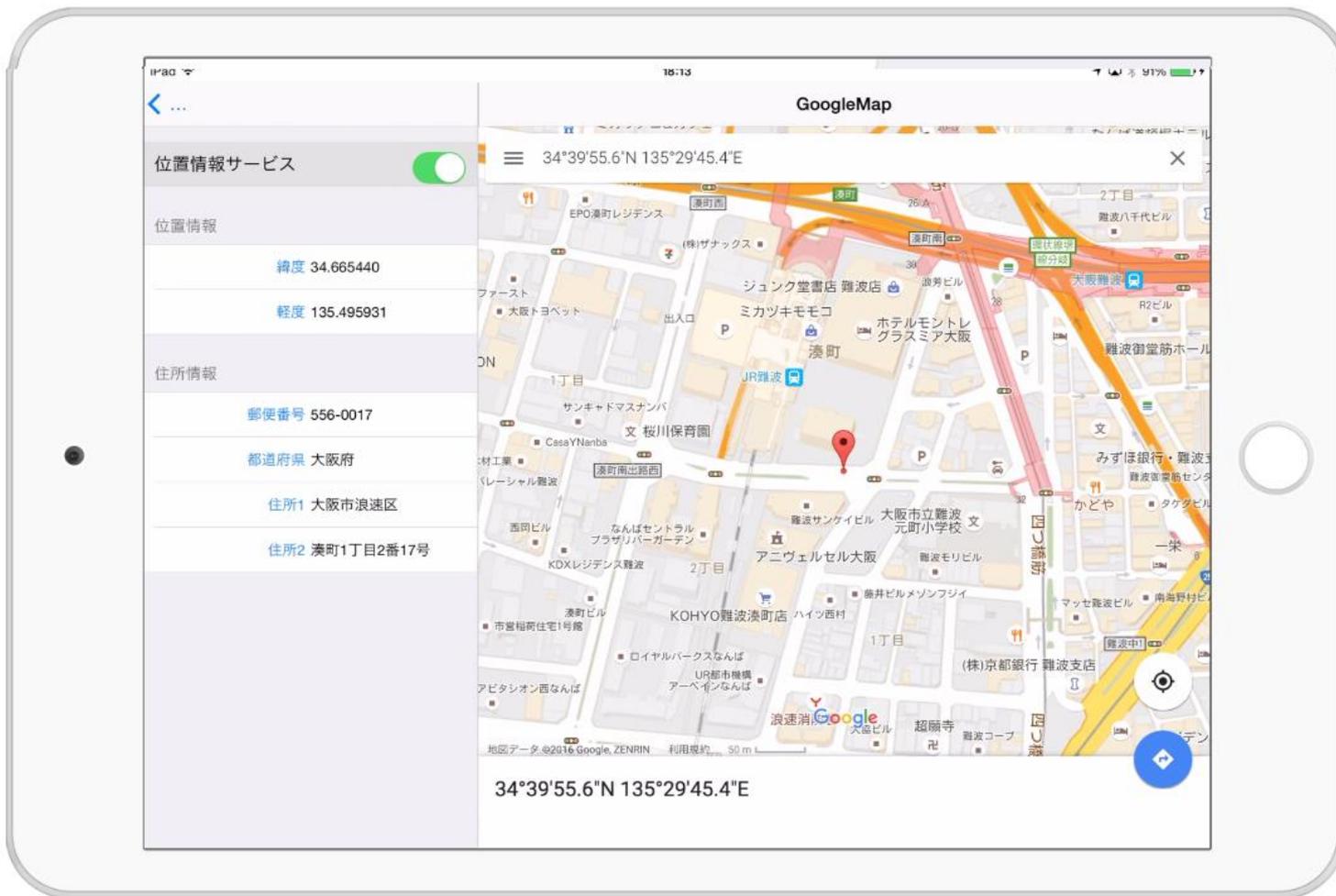


5.他アプリの起動連携【FireMonkey】

iOS



- デモ



5.他アプリの起動連携【FireMonkey】

Android

- 開発プログラムにURLスキームを含めるには？
通常のAndroid開発アプリケーションには
URLスキームは含まれない。

URLスキームを定義するにはAndroidManifest.xmlファイルに追加する必要がある。

Delphiではテンプレートファイルが用意されているので、プロジェクトと同じフォルダにあるAndroidManifest.template.xmlへ次のように追記する。

AndroidManifest.template.xmlファイルに編集で追記する内容

```
<intent-filter>  
  <action android:name="android.intent.action.VIEW" />  
    <category android:name="android.intent.category.DEFAULT" />  
    <category android:name="android.intent.category.BROWSABLE" />  
    <data android:scheme="TEST"/>  
</intent-filter>
```

URLスキームを「TEST」と設定。

5.他アプリの起動連携【FireMonkey】

Android

- 開発アプリをURLスキームで呼び出す

OnClickイベント(連携起動処理)

例)

usesに以下を追加

```
FMX.Helpers.Android, Androidapi.JNI.GraphicsContentViewText,  
Androidapi.JNI.Net, Androidapi.JNI.JavaTypes, Androidapi.Helpers;
```

```
procedure TForm1.Button1Click (Sender: TObject);
```

```
var
```

```
    Intent: JIntent;
```

```
begin
```

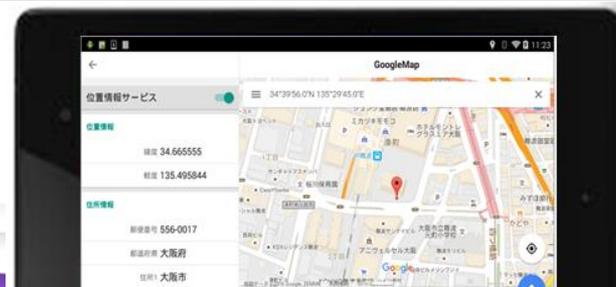
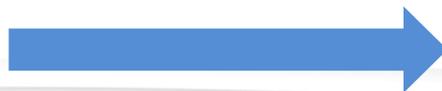
```
    Intent := TJIntent.JavaClass.init(TJIntent.JavaClass.ACTION_VIEW,
```

```
    TJnet_Uri.JavaClass.parse(StringToJString('TEST:// ')));
```

```
    SharedActivity.startActivity(Intent);
```

```
end;
```

```
end;
```



5.他アプリの起動連携【FireMonkey】



- デモ



5.他アプリの起動連携【FireMonkey】

- デバイスで異なるコーディング

今回のプログラムでは、URLスキームを呼び出すプログラムがiOSとAndroidでコーディングが違っている。

1ソース、1アプリで管理するためには

条件付きコンパイル

が活用できる。

例)

{`$IFDEF` 条件}

プログラムコード自体を
IF文のような構文で使い分けることができる。

条件一致する場合のみ動作するコード

{`$ENDIF` 条件}

5.他アプリの起動連携【FireMonkey】

iOS

Android

• 条件付きコンパイル

usesの例

```
uses
  System.SysUtils, System.Types, System.UITypes, System.Classes,
  System.Variants,
  FMX.Types, FMX.Controls, FMX.Forms, FMX.Graphics, FMX.Dialogs,
  FMX.Controls.Presentation, FMX.StdCtrls,

{$IFDEF ANDROID}
  FMX.Helpers.Android, Androidapi.JNI.GraphicsContentViewText,
  Androidapi.JNI.Net, Androidapi.JNI.JavaTypes, Androidapi.Helpers;
{$ELSE}
{$IFDEF IOS}
  iOSapi.Foundation, FMX.Helpers.iOS, Macapi.Helpers;
{$ELSE};
{$ENDIF IOS}
{$ENDIF ANDROID}
```

呼び出しプログラムの例では
usesもiOSとAndroidで異なるが
1つのソースで記述できる！

Androidだけ有効

iOSだけ有効

5.他アプリの起動連携【FireMonkey】

iOS

Android

- 条件付きコンパイル

OnClickの例

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
{ $IFDEF ANDROID }
```

```
var
```

```
    Intent: JIntent;
```

```
begin
```

```
    Intent := TJIntent.JavaClass.init(TJIntent.JavaClass.ACTION_VIEW,
```

```
    TJnet_Uri.JavaClass.parse(StringToJString('TEST:// ')));
```

```
    SharedActivity.startActivity(Intent);
```

```
{ $ELSE }
```

```
{ $IFDEF IOS }
```

```
begin
```

```
    SharedApplication.openURL(StrToNSUrl('TEST://'));
```

```
{ $ELSE };
```

```
{ $ENDIF IOS }
```

```
{ $ENDIF ANDROID }
```

```
end;
```

Androidだけ有効

iOSだけ有効

5.他アプリの起動連携【FireMonkey】

- 条件付きコンパイル

条件付きコンパイルで使用できる条件はOSのようなプラットフォームやCPU、デバッグ/リリース、機能など様々【条件シンボル定義 プラットフォーム例】

定義	説明
Win32	32bit Windowsアプリケーション
Win64	64bit Windowsアプリケーション
MACOS	Mac OS X アプリケーション
IOS	iOSアプリケーション
Android	Androidアプリケーション

詳しい条件はヘルプや下記Web上を参考ください。

<http://docwiki.embarcadero.com/RADStudio/Seattle/ja/条件付きコンパイル> (Delphi)

3.まとめ

- ユーザーパスワードの期限変更制御は環境で設定できる
- SQLのデータ抽出では構文でレコード番号が付与できる
- IntraWebのセッションタイムアウトはWebサーバとアプリケーションで管理する
(IntraWebはバージョンに注意)
- GPS情報は専用コンポーネントで取得してGoogleMapを呼び出すことができる
- モバイルアプリはURLスキーム情報を付与することでアプリ間の呼び出しができる

ご清聴ありがとうございました。