

株式会社ミガロ.

システム事業部 プロジェクト推進室

【Delphi/400】

Google Maps Platformを使用したアプリケーション開発テクニック

1. はじめに
2. Delphi/400 で Google マップを表示する方法
 - 2-1. Google Maps Platform の利用手続き
 - 2-2. Google マップを表示する
3. Google マップの機能を連携するテクニック
 - 3-1. Google マップにマーカーを立てる
 - 3-2. クリックしたマーカーの情報を取得する
 - 3-3. Google マップを移動・拡大する
4. 最後に



略歴 福井 和彦
 1972年3月20日生まれ
 1994年3月 大阪電気通信大学 工学部卒業
 2001年4月 株式会社ミガロ、入社
 2001年4月 システム事業部配属

現在の仕事内容
 主に Delphi/400 を使用したシステムの受託開発全般に携わっている。



略歴 小杉 智昭
 1973年5月26日生まれ
 1996年3月 関西大学 工学部卒業
 2002年3月 株式会社ミガロ、入社
 2002年3月 RRAD 事業部配属
 2007年4月 システム事業部配属

現在の仕事内容：
 Delphi/400 を利用した受託開発とシステム保守、導入支援を担当している。

1.はじめに

Google は 2018 年 5 月 2 日(米国時間)に、従来の Google Maps API を同年 6 月 11 日 (米国時間) より Google Maps Platform という新しいサービスに移行することを発表した。新しいサービスでは、これまでスタンダードプランとプレミアムプランの 2 種類あった料金プランが統合され、使用量に応じた従量課金の単一プランとなった。この新しい料金プランでは毎月 200USD 分が無料使用枠として付与されることになる。

これまでは Google Maps API の使用環境が社内イントラネットや会員制サイトなど、非公開環境の場合にはプレミアムプランへの契約 (年間 10,000USD より) が必要であった。しかし Google Maps Platform では、使用環境が公開・非公開に関係なく従量課金の単一プランとなったため、社内ネットワークで使用する業務システムで、Google マップとの連携が行いやすくなった。

ただし、注意していただきたいのは、

請求先アカウントの設定としてクレジットカード情報の登録が必須となる点である。詳しくは、下記「Google Cloud Japan 公式ブログ」の 2018 年 5 月 8 日の記事をご確認いただきたい。

<https://cloud-ja.googleblog.com/2018/05>

そこで本稿では、Delphi/400 で業務システムを開発する際に広く使われている VCL (C/S 型) のプログラムで、Google Maps Platform を使用して Google マップと連携する方法について説明する。

2.Delphi/400でGoogleマップを表示する方法

2-1.Google Maps Platformの利用手続き

Google Maps Platform を使用する場合 API キーが必須となる。そして、

API キーを取得するには Google アカウントが必要となるため、Google アカウントをお持ちでない場合は事前の作成が必要となる。

Google アカウントの作成サイト

<https://accounts.google.com/signup/v2/webcreateaccount?hl=ja&flowName=GlifWebSignIn&flowEntry=SignUp>

(Google で “Google アカウントの作成” を検索すると上位に表示される「Google アカウントの作成」。)

API キーを取得するには、Google アカウントへログインした後、次のサイトの「使ってみる」ボタン【図 1】より、手順に従って登録を行っていく。

Google Maps Platform 公式サイト

<https://cloud.google.com/maps-platform/>

詳しい登録手順は、さまざまなサイト

図1 Google Maps Platformの利用手続きについて①

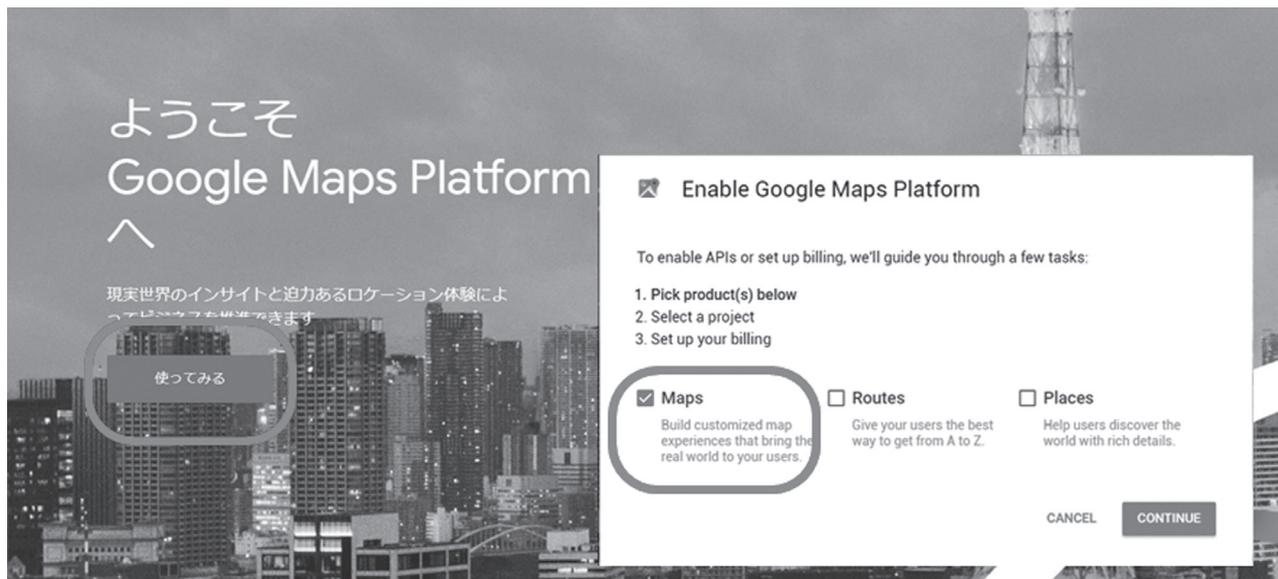
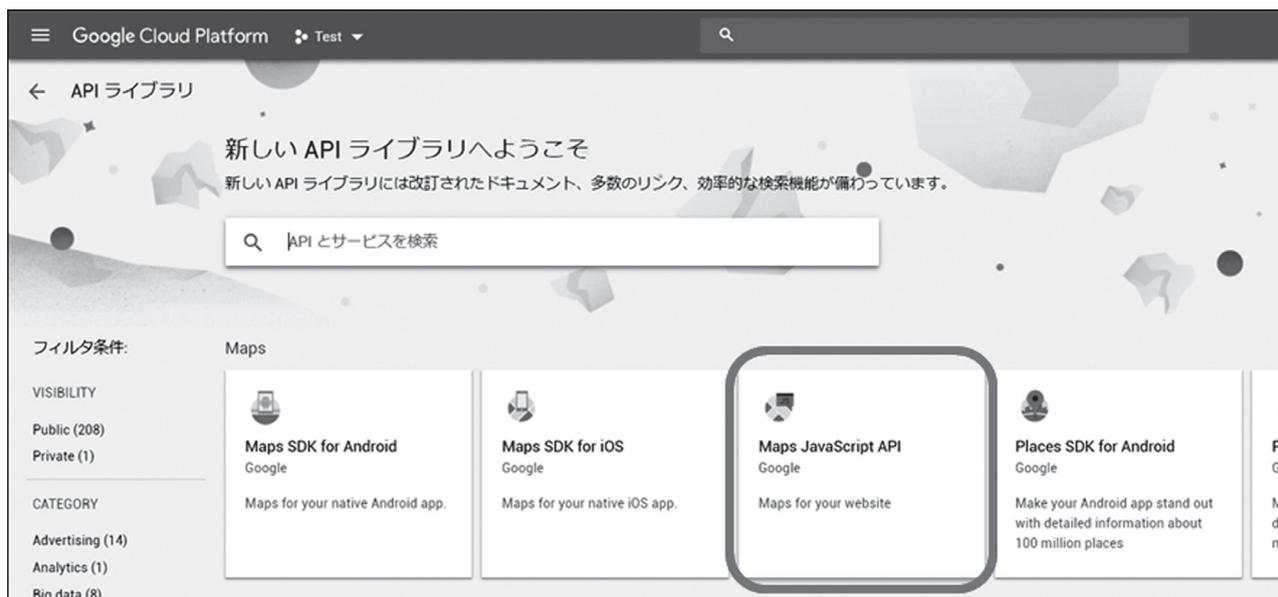


図2 Google Maps Platformの利用手続きについて②



で紹介されているためここでは割愛するが、ポイントは次の3点である。

- (1) プロダクトの選択では「Maps」にチェックを付ける。【図1】
- (2) API ライブラリより「Maps JavaScript API」を有効にする。【図2】
- (3) Maps JavaScript API のキーの制限設定を行う。【図3】【図4】

登録が完了したら、「APIとサービス」の認証情報より API キーを取得する。【図5】

2-2. Googleマップを表示する

VCL フォーム上に、2章で取得した API キーを設定した Google Maps Platform を使用して Google マップを表示してみる。

VCL フォームに Web サイトを表示するには「TWebBrowser」コンポーネントを使用する。TWebBrowser コンポーネントは標準では Internet Explorer7.0 の互換モードとして動作する。Google Maps Platform を使用して Google マップを表示する場合、HTML5 として宣言することが推奨されているため、Internet Explorer7.0 の互換モードでは表示されない。そこで、Internet Explorer の最新バージョンとして動作するように設定をする。【図6】のようにレジストリのキーに値を追加することで設定ができる。

設定が完了したら、Delphi/400 を起動して次の手順で Google マップを表示する画面を作成する。

(1) フォームに TWebBrowser コンポーネントを貼り付ける。【図7】

(2) Google マップ表示用 HTML を定数として定義する（【ソース1】を参照）。

この HTML に、2章で取得した API キーを組み込む。また、Google マップを表示する際、中心位置を緯度と経度で指定する「center」と、地図の拡大レベルを指定する「zoom」が必須指定となる。この HTML で指定している緯度と経度は、オーストラリアのシドニー近郊の位置を指定している。

* HTML は、次の Google のサイトで公開されているサンプル HTML を使

用している。

参考サイト

<https://developers.google.com/maps/documentation/>

（上記 URL にある“Maps JavaScript API”のリンク先）

(3) フォームの OnShow イベントに【ソース2】のように記述する。

ここで OnShow イベントで行っていることについて説明を行う。

【ソース2のポイント①】

TWebBrowser コンポーネントは OLE コンテナであり、このコンポーネントを貼り付けただけでは Internet Explorer オブジェクト（以下、IE）がまだ準備されていない状態である。このままでは IE に HTML を流し込むための命令が使えないため、まず最初に Navigate メソッドを使って空白ページを表示させる指示を行う。そうすることで IE を OLE コンテナ上に準備させる。

【ソース2のポイント②】

ポイント①の Navigate メソッドを実行後、Delphi/400 のプログラムは IE の準備完了を待たずに次へ進んでしまうため、IE の準備が完了したかどうかを ReadyState プロパティで確認する。ReadyState プロパティが返す状態は【表1】のとおりである。

ReadyState プロパティが READYSTATE_INTERACTIVE（IE の操作が可能になる状態）まで待ってから、表示したい HTML を IE に流し込む。

HTML を IE に表示させる方法はいくつもあるが、本稿では汎用性の高いストリームを使う方法で行う。ストリームを使うメリットは以下のとおりである。

●ストリームを使うメリット

- ・プログラムコード内に直接埋め込んだ HTML コードが利用可能。
- ・ファイルストリームと連携させることで HTML ファイルを利用可能。
- ・エンコード指定を行うことで、文字コードの変更が可能。

【ソース2のポイント③】

ポイント②で IE が操作可能になるまで待っているが、エラー等に備えて TWebBrowser コンポーネントの Document プロパティが利用可能になっていることを確認する。

次に文字列リストとストリームを準備する。ストリームは TMemory Stream を使用する。HTML の文字列定数（cGoogleMap）を文字列リストの Text プロパティへセットし、文字列リストの SaveToStream メソッドでストリームに書き込む。ただし、SaveToStream メソッドで書き込んだ直後はストリームの位置が HTML の末尾に移動してしまうため、ストリームの Seek メソッドで先頭位置へ戻している。（位置の変更は Position プロパティでも可能）

そして、IE の IPersistStreamInit インターフェースの Load メソッドでストリームを渡すのだが、Delphi/400 のストリーム（TStream 型）を COM ベースの IStream 型へ変換する必要があるため、TStreamAdapter クラスを経由して渡す。TStreamAdapter クラスは TInterfacedObject クラスを継承しており、自動解放されるため明示的な解放を行う必要はない。最後にストリームと文字列リストを解放して、IE への表示処理は終了となる。

完成した画面を実行すると、オーストラリアのシドニー周辺の地図が表示される。【図8】

3. Googleマップの機能を連携するテクニック

3-1. Googleマップにマーカーを立てる

この章では、サンプルファイルに登録されている複数の営業所の位置を、緯度と経度を使用して Google マップ上にマーカーを立てる方法について説明をする。完成イメージは【図9】となる。また、緯度と経度についてはファイルに登録されているものとする。この章以降で使用するサンプルファイルのイメージは、【図10】【図11】を参照していただきたい。サンプルファイルは、IBM i の特定のライブラリーに存在するものとする。

図3 Google Maps Platformの利用手続きについて③



図4 Google Maps Platformの利用手続きについて④



それでは、次の手順で画面を作成していく。

(1)【図 12】のように、フォームに TDBGrid コンポーネントと TWebBrowser コンポーネントを貼り付ける。そして、サンプルファイルを参照するために FireDAC 関連のコンポーネントと TDataSource コンポーネントを貼り付ける。FireDAC 関連の各コンポーネントの設定は次のとおり。

●TFDConnection コンポーネントの設定

接続エディタにて設定する。

- ・接続定義名：CO400DEF
- ・Database：Delphi/400Configuration の AS/400Name
- ・User_Name：IBM i 接続ユーザー
- ・Password：IBM i 接続ユーザーのパスワード
- ・ODBCAdvanced：LibraryOption
=[サンプルファイルが有る Library 名]

●TFDQuery コンポーネントの設定

SQL プロパティにサンプルファイルを参照する以下の SQL 文を埋め込む。(SQL 文)

```
SELECT * FROM [ サンプルファイル名 ]  
ORDER BY SPEGCD
```

TDBGrid コンポーネントのカラムには営業所名 (SPEGNM) を設定する。

(2)Google マップ表示用 HTML は【ソース 3】【ソース 4】を参照。この章では HTML を 2 つに分割している。分割している理由については後述する。また【ソース 3】では、Google マップにマーカーを立てるために Google Maps Platform の API を使用した JavaScript 「setMarker」を組み込んでいる。

【ソース 4】では、Google マップ表示後に地図の中心にしたい緯度と経度を組み込んでいる。本稿では「難波サンケイビル 本社」を中心にするため、その緯度と経度を組み込んでいる。そして 3 章と同様に、2 章で取得した API キーを HTML へ組み込む。

(3) フォームの OnShow イベントでは、Google マップ表示用 HTML を使用して TWebBrowser コンポーネントに Google マップを表示している。OnShow イベントに【ソース 5】【ソース 6】のように記述する。それでは OnShow イベントで行っている内容について詳しく見ていこう。

【ソース 5 のポイント①】

IBM i に接続し、TDBGrid コンポーネントに営業所名の一覧を表示する。

【ソース 5 のポイント②】

ここでは Google マップを表示する HTML を編集している。HTML の編集イメージは【図 13】を参照していただきたい。

まず、サンプルファイルを全件参照しながら、各レコードの緯度 (SPLATI) と経度 (SPLNGI) を取得する。そして【ソース 3】で組み込んだ、マーカーを立てる JavaScript 「setMarker」の引数に取得した緯度と経度をセットする。この setMarker を【ソース 3】の HTML の続きとして、サンプルファイルに登録されているデータ全件分を結合する。こうすることで、1 回の HTML のリクエストで全件分の営業所のマーカーが付いた状態の Google マップを表示することができる。

【ソース 6 のポイント①】

【ソース 6】の Google マップを表示する処理については 3 章の内容と同じになる。ただし、Google マップを表示する HTML をセットする個所は、【ソース 5 のポイント②】で説明した編集結果をセットするようにする。

完成した画面を実行すると完成イメージ【図 9】のように、各営業所の位置にマーカーが立っている Google マップが表示される。

3-2. クリックしたマーカーの情報を取得する

この章では、3-1 で作成した画面に機能を追加し、クリックした Google マップのマーカーに保管している情報 (営業所 CD) を取得して、その情報を基に営

業所の詳細情報を表示する方法について説明する。ただし、マーカーの情報を取得するためには TWebBrowser コンポーネントを経由して JavaScript を実行して戻り値を取得する必要がある。しかし、TWebBrowser コンポーネントには JavaScript を実行するメソッドは実装されていない。独自で実装することも可能ではあるが簡単ではない。

そこで本稿では、JavaScript を実行するメソッドなど TWebBrowser コンポーネントでは実装されていないメソッドやプロパティが実装されている「TEmbeddedWB」コンポーネントというフリーのコンポーネントを利用する。TEmbeddedWB コンポーネントは bsalsa productions にて開発・公開されていたフリーのコンポーネント群の 1 つだが、現在は little earth solutions によって GitHub 上でメンテナンス・公開されている。

TEmbeddedWB コンポーネントのダウンロードサイト

<https://github.com/littleearth/Delphi-EmbeddedWB>

GitHub 上では Delphi 5 ~ Delphi 10 Seattle までが対象となっているが、Delphi 10.1 Berlin や Delphi 10.2 Tokyo 向けの dpk ファイルも存在する (* dpk ファイルの読み込み前に res ファイルを削除しなければ正しく取り込めない)。ダウンロードサイトより TEmbeddedWB コンポーネントをダウンロードして Delphi/400 の開発環境にインストールをする。

それでは、次の手順に従って、4 章で作成した画面に機能を追加していく。

(1)【図 14】に従ってコンポーネントの追加/変更を行う。変更内容は、まず TWebBrowser コンポーネントを先程インストールした TEmbeddedWB コンポーネントに入れ替える。

次に、TTimer コンポーネントを追加し、Enabled プロパティを False、Interval プロパティを 200 に設定する。TTimer コンポーネントの用途については後述する。そして、営業所の詳細情報を表示するために TDBEdit を追加する。

図5 APIキーの取得



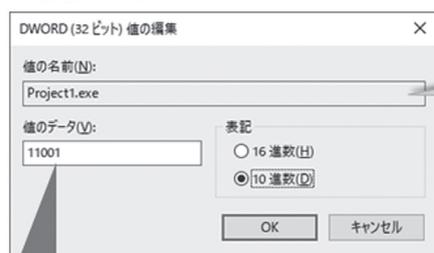
図6 Googleマップを表示する①

レジストリのキーに値を追加する

<値を追加するレジストリのキー>

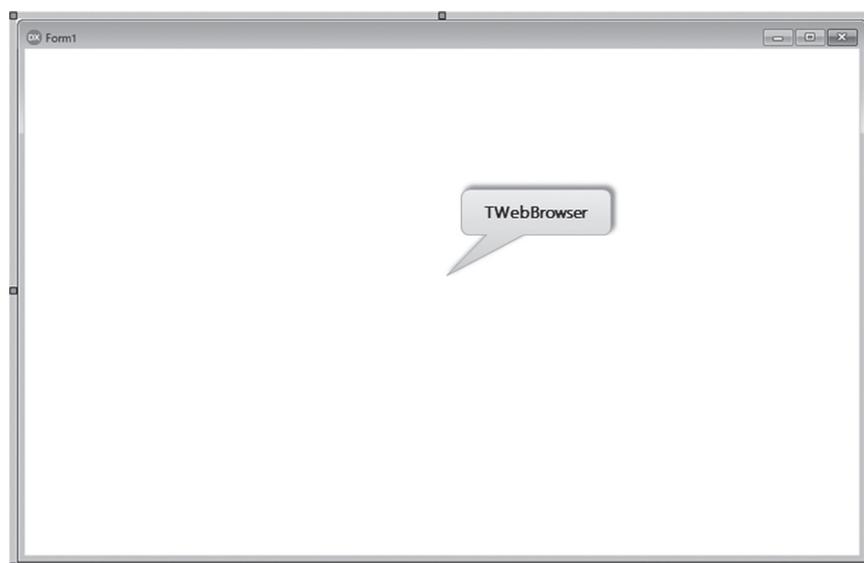
¥HKKEY_CURRENT_USER¥Software¥Microsoft¥Internet Explorer¥Main¥FeatureControl¥FEATURE_BROWSER_EMULATION

<追加する値>



値 (10進数)	値 (16進数)	バージョン
11001	0x2AF9	Internet Explorer 11, Edgeモード (最新のバージョンでレンダリング)
11000	0x2AF8	Internet Explorer 11

図7 Googleマップを表示する②



(2) Google マップを表示する HTML 【ソース 3】 【ソース 4】 に対して、【ソース 7】 【ソース 8】 のように変更を行う。変更内容は以下のとおり。

【ソース 8 の変更点①】

説明の順番が前後するが、グローバル変数である営業所 CD (mrkegcd) を追加宣言する。この営業所 CD (mrkegcd) を【ソース 7】内で使用する。

【ソース 7 の変更点①】

まず setMarker の引数に営業所 CD を受け取る項目 (egcd) を追加する。

【ソース 7 の変更点②】

受け取った営業所 CD (egcd) を、マーカーに紐付けて保管する営業所 CD (marker.egcd) にセットする。

【ソース 7 の変更点③】

マーカーにクリックイベントを定義し、マーカーをクリックした際に、マーカーに保管している営業所 CD (marker.egcd) をグローバル変数の営業所 CD (mrkegcd) にセットする。

【ソース 7 の変更点④】

新たに営業所 CD (mrkegcd) の値を取得する JavaScript 「getmrkegcd」を追加し、営業所 CD (mrkegcd) を戻り値として返すようにする。

(3) フォームの OnShow イベントを【ソース 9】に従って変更を行う。変更内容は、【ソース 7】にて setMarker の引数に営業所 CD が追加されたため、【ソース 5】の Google マップを表示する HTML を編集しているところで使用している setMarker の引数に、【ソース 9 の変更点①】のようにサンプルファイルの営業所 CD (SPEGCD) をセットするようにする。これでマーカーを立てる際に営業所 CD (SPEGCD) がマーカーに紐付いて保管される。

(4) Google マップのマーカーをクリックした時に、【ソース 7】で setMarker に定義したクリックイベントで、マーカーで保管している営業所 CD がグローバル変数の営業所 CD にセットされている。VCL からは TEmbeddedWB コン

ポーネントの OnClick イベント (以下 OnClick イベントとする) で、【ソース 7】で実装された getmrkegcd を実行してグローバル変数の営業所 CD を戻り値とし受け取るようにする。しかし、OnClick イベントの方がマーカーのクリックイベントよりも先に発生してしまうため、戻り値が正しく取得できない。そこで、OnClick イベントと、マーカーのクリックイベントが非同期で発生することを考慮し、TTimer コンポーネントで getmrkegcd の実行タイミングを遅らせる。

OnClick イベントと TTimer コンポーネントの OnTimer イベントを【ソース 10】のように実装する。

【ソース 10 のポイント①】

OnClick イベントでは、TTimer コンポーネントの OnTimer イベントを起動するのみとなる。

【ソース 10 のポイント②】

TTimer コンポーネントの OnTimer イベントで、OnClick イベント発生後 (200 ミリ秒後) に getmrkegcd を実行して戻り値の営業所 CD を取得する。JavaScript である getmrkegcd を実行するには TEmbeddedWB コンポーネントに実装されている 「ExecScriptEx」メソッドを使用する。ExecScriptEx メソッドは実行した Script の戻り値を取得できるメソッドで、語尾に “Ex” が付いていない 「ExecScript」メソッドも実装されているが、こちらは Script を実行するのみのメソッドとなる。

【ソース 10 のポイント③】

取得した営業所 CD でサンプルファイルより該当営業所 CD を検索し詳細情報を表示させる。

ここまでで実装した「クリックしたマーカーの営業所 CD を取得する」処理のイメージは【図 15】を参照いただきたい。

完成した画面を実行すると完成イメージ【図 16】のように、Google マップ上のマーカーをクリックすると、画面左の営業所一覧はクリックした営業所にレコードが移動し、それに合わせて

画面下の営業所詳細情報はクリックした営業所の情報が表示される。

3-3. Google マップを移動・拡大する

この章では、3-1 から使用している画面にさらに機能を追加し、画面左の営業所一覧をクリックすると、Google マップの該当営業所のマーカーが真ん中に移動し、かつ拡大して表示する方法について説明する。

この章での変更はソースのみとなる。まずは Google マップを表示する HTML を【ソース 11】に従って変更する。

【ソース 11 の変更点①】

引数で受け取った緯度と経度を Google マップの中心にし、拡大表示する JavaScript 「moveMapCenter」を追加実装する。

次に、【ソース 12】に従って TDBGrid コンポーネントの OnClick イベントを実装する。【ソース 11】で実装した moveMapCenter にサンプルファイルの緯度と経度を引数として渡して実行する。

完成した画面を実行すると完成イメージ【図 17】のように、画面左の営業所一覧をクリックすると Google マップの該当営業所のマーカーが真ん中に移動し、かつ拡大して表示される。

以上で「Delphi/400 で Google Maps Platform を使用したアプリケーション開発テクニック」の説明は終了となる。

4.最後に

初めに述べたように、Google Maps Platform のサービス開始に伴い、非公開の社内用 Web アプリケーションや C/S 型アプリケーションでもプレミアムプランの契約なしに、API を用いた Google マップが利用できるようになった。今後は、そのプレミアムプラン契約がネックとなって利用を諦めていた非公開環境での活用も増えてくるだろう。

ただし、無料で利用できる枠には限度があり、無料で利用できる枠は使用するサービスによって異なるため、Google Maps Platform を利用するには注意

ソース1 Googleマップ表示用HTML

```

cGoogleMap = '<!DOCTYPE html>'
+ '<html>'
+ '<head>'
+ '<title>Simple Map</title>'
+ '<meta name="viewport" content="initial-scale=1.0">'
+ '<meta charset="utf-8">'
+ '<style>'
+ '/* Always set the map height explicitly to define the size of the div'
+ ' * element that contains the map. */'
+ '#map {'
+ '  height: 100%;'
+ '}'
+ '/* Optional: Makes the sample page fill the window. */'
+ 'html, body {'
+ '  height: 100%;'
+ '  margin: 0;'
+ '  padding: 0;'
+ '}'
+ '</style>'
+ '</head>'
+ '<body>'
+ '<div id="map"></div>'
+ '<script>'
+ '  var map;'
+ '  function initMap() {'
+ '    map = new google.maps.Map(document.getElementById(''map''), {'
+ '      center: {lat: -34.397, lng: 150.644},'
+ '      zoom: 8,'
+ '    });'
+ '  }'
+ '</script>'
+ '<script src="https://maps.googleapis.com/maps/api/js?key=[APIキー]&callback=initMap"'
+ '  async_defer></script>'
+ '</body>'
+ '</html>';
  
```

centerとzoomは
必須オプション

[APIキー]

2章で取得したAPIキーを設定する

ソース2 フォームのOnShowイベント

```

procedure TForm1.FormShow(Sender: TObject);
const
  [ソース1] (Googleマップ表示用HTML)
var
  sl: TStringList;
  ms: TMemoryStream;
begin
  WebBrowser1.Navigate('about:blank');
  while WebBrowser1.ReadyState < READYSTATE_INTERACTIVE do
    Application.ProcessMessages;
  if Assigned(WebBrowser1.Document) then
  begin
    sl := TStringList.Create;
    try
      ms := TMemoryStream.Create;
      try
        sl.Text := cGoogleMap;
        sl.SaveToStream(ms);
        ms.Seek(0, 0);
        (WebBrowser1.Document as IPersistStreamInit).Load(TStreamAdapter.Create(ms));
      finally
        ms.Free;
      end;
    finally
      sl.Free;
    end;
  end;
end;
  
```

usesにWinapi.ActiveXを追加する

ポイント①

ポイント②

ポイント③

「定数」→「文字列リスト」

「文字列リスト」→「ストリーム」

「ストリーム」→「IE」

が必要となる。詳しくは、下記参考サイトで確認していただきたい。

●参考サイト

<https://cloud.google.com/maps-platform/>

(上記 URL のメニューにある “料金” のリンク先)

M

表1 ReadyStateプロパティが返す状態

ReadyStateプロパティの戻り値	状態
READYSTATE_UNINITIALIZED	IEに対する初期化未完了
READYSTATE_LOADING	IEのロード中
READYSTATE_LOADED	IEのロード完了・操作不可能
READYSTATE_INTERACTIVE	IEの操作可能
READYSTATE_COMPLETE	IEの全データ読込完了

図8 Googleマップを表示する③

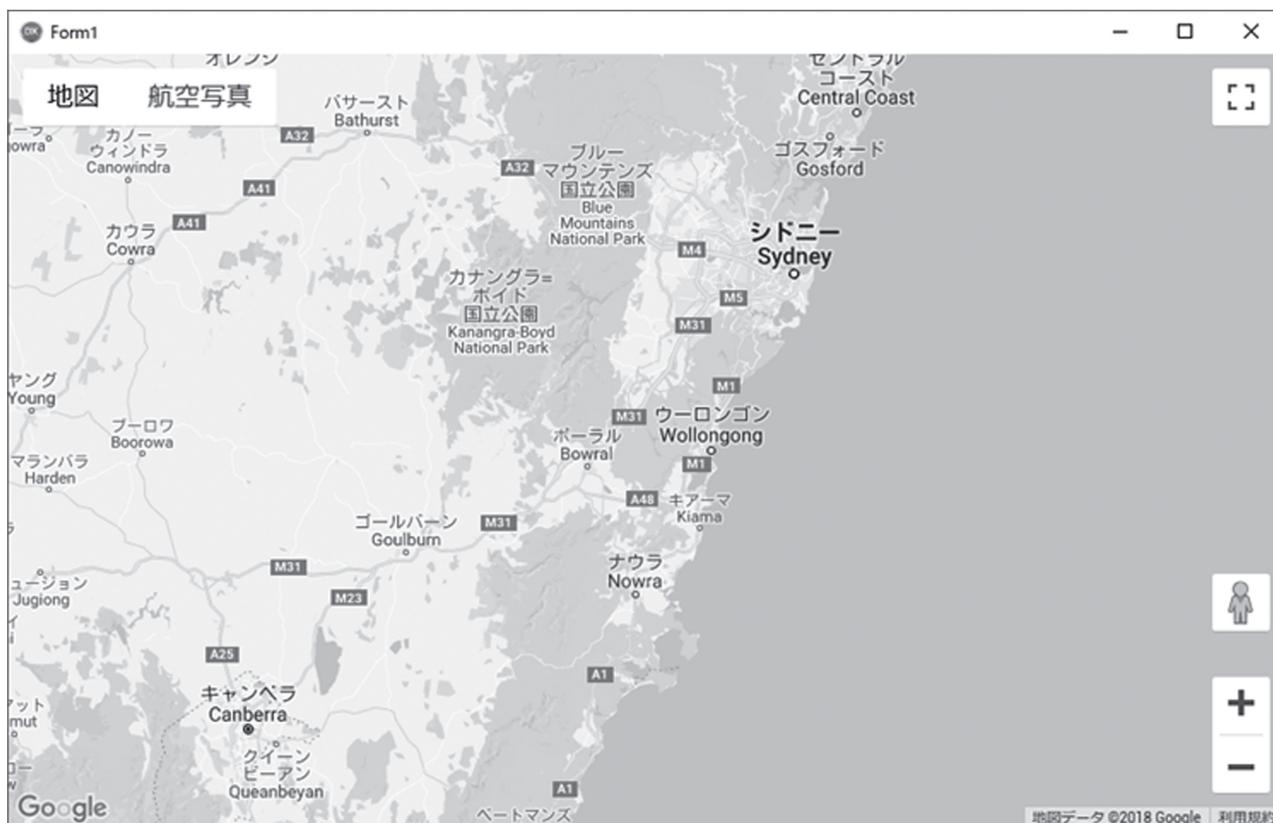


図9 Googleマップにマーカーを立てる①



図10 Googleマップにマーカーを立てる②

サンプルファイルイメージ

No.	Key	項目ID	項目名	属性(桁数)
1	1	SPEGCD	営業所CD	A (6)
2		SPEGNM	営業所名	O (32)
3		SPEGRN	営業所略称	O (22)
4		SPPOST	郵便番号	A (8)
5		SPADR1	住所1	O (42)
6		SPADR2	住所2	O (42)
7		SPTELN	電話番号	A (15)
8		SPFAXN	FAX番号	A (15)
9		SPLATI	緯度	S (10, 7)
10		SPLNGI	経度	S (10, 7)

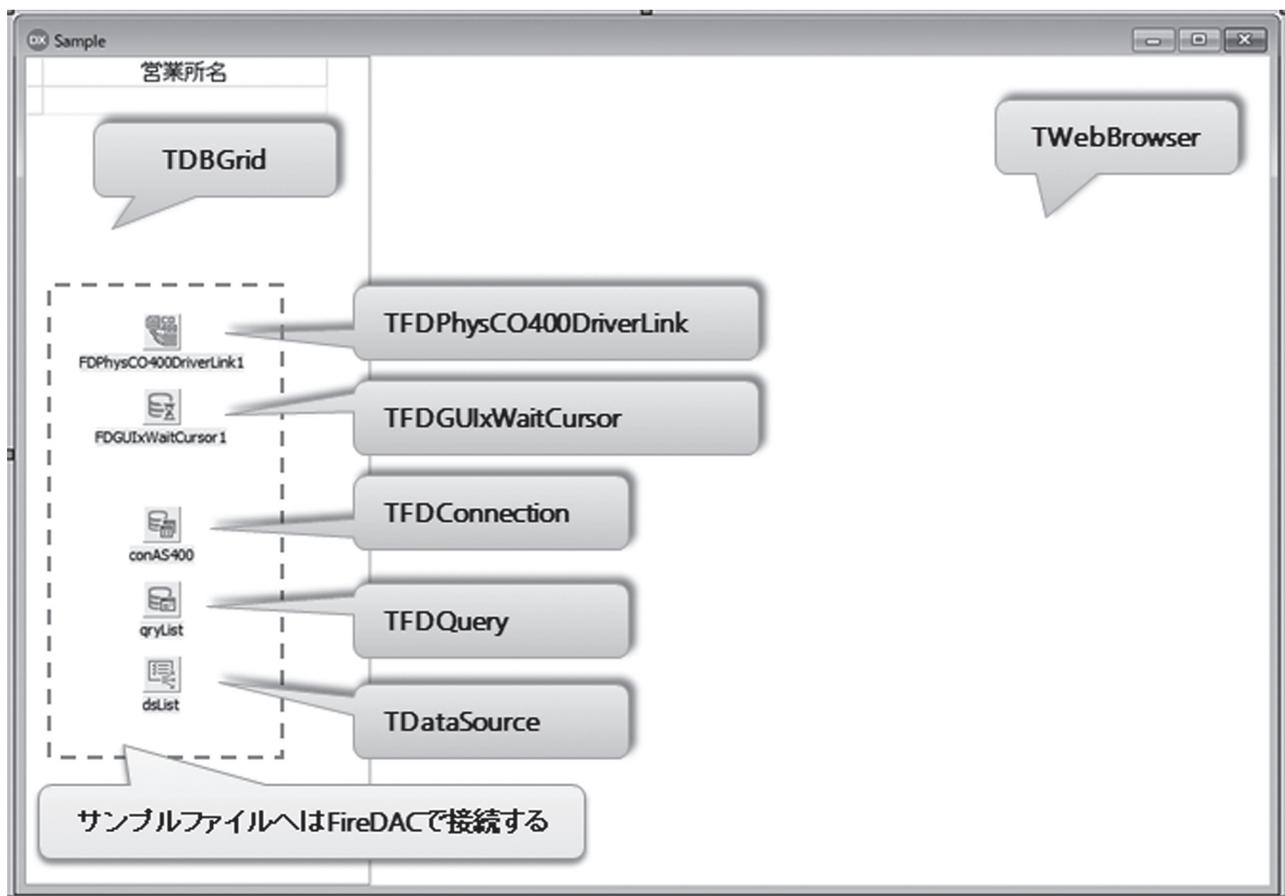
図11 Googleマップにマーカーを立てる③

サンプルデータイメージ

営業所CD	営業所名	営業所略称	郵便番号	住所1	住所2
100001	難波サンケイビル 本社	難波サンケイ	556-0017	大阪府大阪市浪速区湊町～	難波サンケイビル～
200001	霞が関 営業所	霞が関	100-0013	東京都千代田区霞が関～	霞が関東急ビル～

電話番号	FAX番号	緯度	経度
06-6631-8601	06-6631-8603	34.6651420	135.4955550
03-5510-5701	03-5510-5702	35.6714410	139.7447100

図12 Googleマップにマーカーを立てる④



ソース3 Googleマップ表示用HTML①

```
GMapHTML_1
= '<!DOCTYPE html>'
+ '<html>'
+ ' <head>'
+ '   <title>Simple Map</title>'
+ '   <meta name="viewport" content="initial-scale=1.0">'
+ '   <meta charset="utf-8">'
+ '   <style>'
+ '     /* Always set the map height explicitly to define the size of the div'
+ '      * element that contains the map. */'
+ '     #map {'
+ '       height: 100%;'
+ '     }'
+ '     /* Optional: Makes the sample page fill the window. */'
+ '     html, body {'
+ '       height: 100%;'
+ '       margin: 0;'
+ '       padding: 0;'
+ '     }'
+ '   </style>'
+ '   <script type="text/javascript">'
+ ' // マーカーセット用関数
+ '     function setMarker(lat, lng) {'
+ '       var latlng = new google.maps.LatLng(lat, lng);'
+ '       var marker;'
+ '       marker = new google.maps.Marker({map: map, position: latlng});'
+ '     }'
+ '     window.onload = function() {';
```

Google Maps Platformの
APIで、緯度と経度から
マーカーをセットする

ソース4 Googleマップ表示用HTML②

```
GMapHTML_2
= '   };'
+ ' </script>'
+ ' </head>'
+ ' <body>'
+ '   <div id="map"></div>'
+ '   <script>'
+ '     var map;'
+ '     function initMap() {'
+ '       map = new google.maps.Map(document.getElementById("map"), {'
+ '         center: [lat: 34.665160, lng: 135.495414],'
+ '         zoom: 14'
+ '       });'
+ '     }'
+ '   </script>'
+ '   <script src="https://maps.googleapis.com/maps/api/js?key=[APIキー]&callback=initMap"'
+ '     async defer></script>'
+ ' </body>'
+ '</html>';
```

難波サンケイビル 本社の緯度/経度を設定
表示直後、難波サンケイビル 本社を中心に
する

2章で取得したAPIキーを設定する

ソース5 フォームのOnShowイベント

```
procedure TFormSample.FormShow(Sender: TObject);
const
  [ソース3] (Googleマップ表示用HTML①)
  [ソース4] (Googleマップ表示用HTML②)
var
  sl: TStringList;
  ms: TMemoryStream;
  sMap: string;
begin
  // IBMiへ接続
  conAS400.Connected := True;
  // 営業所リスト表示
  qryList.Open;
  // 営業所の緯度と経度でマーカーをセットするJavaScriptをHTMLに埋め込む処理
  sMap := GMapHTML_1;
  qryList.DisableControls;
  try
    qryList.Last;
    while not qryList.Bof do
      begin
        sMap := sMap + ' setMarker('
          + qryList.FieldName('SPLATI').AsString + ','
          + qryList.FieldName('SPLNGI').AsString + ');';
        qryList.Prior;
      end;
    finally
      qryList.EnableControls;
    end;
  ↓ [ソース6]へ続く
```

usesにWinapi.ActiveXを追加する

ポイント①

ポイント②

ソース6 フォームのOnShowイベント

```
↓ [ソース5]の続き
// Googleマップ表示
wbMap.Navigate('about:blank');
while wbMap.ReadyState < READYSTATE_INTERACTIVE do
  Application.ProcessMessages;

if Assigned(wbMap.Document) then
begin
  sl := TStringList.Create;
  try
    ms := TMemoryStream.Create;
    try
      sl.Text := sMap + GMapHTML_2;
      sl.SaveToStream(ms);
      ms.Seek(0, 0);
      (wbMap.Document as IPersistStreamInit).Load(TStreamAdapter.Create(ms));
    finally
      ms.Free;
    end;
  finally
    sl.Free;
  end;
end;
end;
```

ポイント①

図13 Googleマップにマーカーを立てる⑤

Googleマップを表示するHTMLのイメージ

【ソース3】Googleマップ表示用HTML①



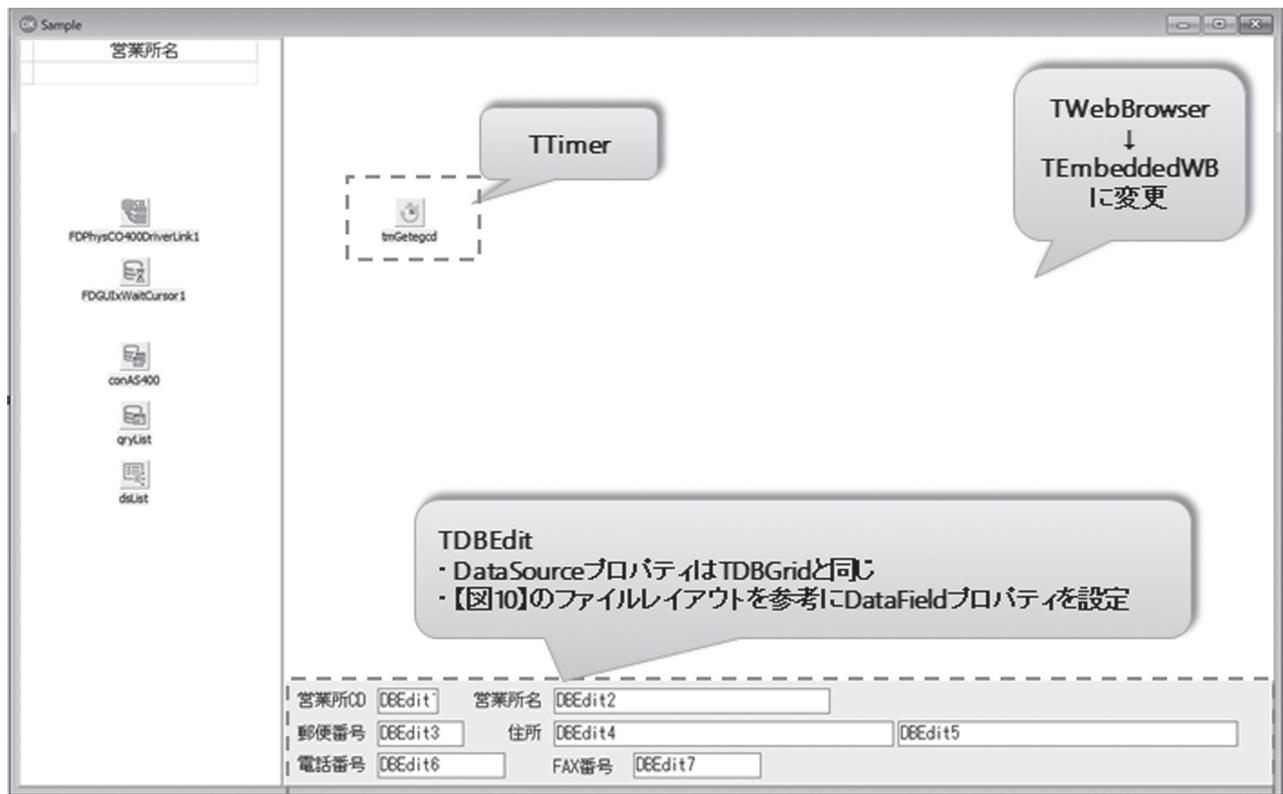
```
setMarker(34.6651420, 135.4955550);  
setMarker(35.6714410, 139.7447100);  
.  
.  
.
```

サンプルファイルに登録されている件数分、マーカーをセットするJavaScriptを【ソース3】の後に結合する。



【ソース4】Googleマップ表示用HTML②

図14 クリックしたマーカーの情報を取得する①



ソース7 Googleマップ表示用HTML①【ソース3】からの変更点

```

GMapHTML_1
= 【ソース3】と同じ～

// マーカーセット用関数
+ ,   function setMarker(lat, lng, egcd) {
+ ,     var latlng = new google.maps.LatLng(lat, lng);
+ ,     var marker;
+ ,     marker = new google.maps.Marker({map: map, position: latlng});

+ ,     marker.egcd = egcd;

+ ,     google.maps.event.addListener(marker, "click", function() {
+ ,       mrkegcd = marker.egcd;

+ ,     });
+ ,   };
+ ,   // 営業所CD (mrkegcd) の値を取得する関数
+ ,   function getmrkegcd() {
+ ,     return (mrkegcd);
+ ,   };
+ ,   window.onload = function() {

```

変更点①
引数に営業所CDを追加

変更点②
引数で受け取った営業所CDをマーカーに紐付けて保管

変更点③
マーカーのクリックイベントを定義し、クリック時にマーカーに保管している営業所CDを変数に退避

変更点④
マーカークリック時に、変数に退避した営業所CDを取得するJavaScript

ソース8 Googleマップ表示用HTML②【ソース4】からの変更点

```

GMapHTML_2
= ,   };
+ ,   </script>
+ ,   </head>
+ ,   <body>
+ ,     <div id="map"></div>
+ ,     <script>
+ ,       var map;
+ ,       var mrkegcd;
+ ,       function initMap() {
+ ,         map = new google.maps.Map(document.getElementById('map'), {
+ ,           center: {lat: 34.665160, lng: 135.495414},
+ ,           zoom: 14
+ ,         });
+ ,       };
+ ,     </script>
+ ,     <script src="https://maps.googleapis.com/maps/api/js?key=[APIキー]&callback=initMap"
+ ,     async defer></script>
+ ,   </body>
+ , </html>;

```

変更点①
マーカークリック時、マーカーに保管している営業所CDを退避する変数を宣言
この変数の値をgetmrkegcd関数で戻り値として返す

ソース9 フォームのOnShowイベント【ソース5】からの変更点

```

procedure TFormSample.FormShow(Sender: TObject);

【ソース5】と同じ～

// 営業所の緯度と経度でマーカーをセットするJavaScriptをHTMLに埋め込む処理
sMap := GMapHTML_1;
qryList.DisableControls;
try
  qryList.Last;
  while not qryList.Bof do
  begin
    sMap := sMap + ' setMarker('
      + qryList.FieldName('SPLATI').AsString + ',';
      + qryList.FieldName('SPLNGL').AsString + ',';
      + QuotedStr(qryList.FieldName('SPEGCD').AsString) + ');';
    qryList.Prior;
  end;
finally
  qryList.EnableControls;
end;

【ソース5】と同じ～

```

変更点①
setMarkerの引数に営業所CDを追加

ソース10 TEmbeddedWBコンポーネントのOnClickイベント

```

procedure TfrmSample.wbMapClick(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  // Googleマップのマーカークリックが後に発生するため、Timerでタイムラグを調整
  tmGetegcd.Enabled := True;
end;
    
```

ポイント①

TTimerコンポーネントのOnTimerイベント

```

procedure TfrmSample.tmGetegcdTimer(Sender: TObject);
var
  vResult: OleVariant;
  segcd: string;
begin
  tmGetegcd.Enabled := False;;

  vResult := wbMap.ExecScriptEx('getmrkegcd', ['']);
  segcd := vResult;

  qryList.Locate('SPEGCD', segcd, []);
end;
    
```

TTimerコンポーネントのIntervalプロパティは200ミリ秒で設定

ポイント②

ポイント③

図15 クリックしたマーカーの情報を取得する②

Googleマップのマーカークリック時の営業所CD受け渡しイメージ



マーカーのクリックイベントで、マーカーで保管している営業所CDをグローバル変数にセット

TEmbeddedWBコンポーネントのクリックイベントで、JavaScript「getmrkegcd」を実行して、営業所CD(グローバル変数)の値を取得する
 ※ getmrkegcdはTTimerにて時間差実行



図16 クリックしたマーカーの情報を取得する③



マーカーをクリックする毎に
該当営業所にレコード移動

マーカーをクリックする毎に
該当営業所の詳細情報表示

ソース11 Googleマップ表示用HTML①JavaScriptの追加

```
GMapHTML_1
= 【ソース7】と同じ～

// 営業所CD (mrkegcd) の値を取得する関数
+ '    function getmrkegcd(){'
+ '        return (mrkegcd);'
+ '    }'

// 経度と緯度をMapの中心にする関数
+ '    function moveMapCenter(lat, lng){'
+ '        var latlng = new google.maps.LatLng(lat, lng);'
+ '        map.setCenter(latlng);'
+ '        map.setZoom(18);'
+ '    }'

+ '    window.onload = function() {';
```

変更点①
引数で受け取った緯度と経度を
Googleマップの中心にして、拡大
表示するJavaScript

緯度と経度を中心にする

拡大表示

ソース12 TDBGridコンポーネントのOnClickイベント

```

procedure TFormSample.dbgListCellClick(Column: TColumn);
begin
  wbMap.ExecScript('moveMapCenter('
    + qryList.FieldByName('SPLATI').AsString + ','
    + qryList.FieldByName('SPLNGI').AsString + ')', 'JavaScript');
end;

```

【ソース11】で実装したmoveMapCenterを実行

図17 Googleマップの移動と拡大①



