

【セッションNo.1】

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2023

株式会社ミガロ。
プロダクト事業部 技術支援課
佐田 雄一



■ Migaro. 技術Tipsについて

- 毎月Delphi/400・Valence・SP4iの技術Tipsを公開中
<https://www.migaro.co.jp/tips/>



今回はサポートでよく頂くご質問に加えて、
この技術Tipsで公開した機能についてもご紹介



【目次】

- WebAPIアクセスを行うアプリケーションを効率よく開発する方法
- 通知機能の実装方法
- 複数バージョンのDelphi/400開発環境を入替する方法
- Edge対応の最新情報



NEW TOPICS

■ WebAPIアクセスを行うアプリケーションを効率よく開発する方法



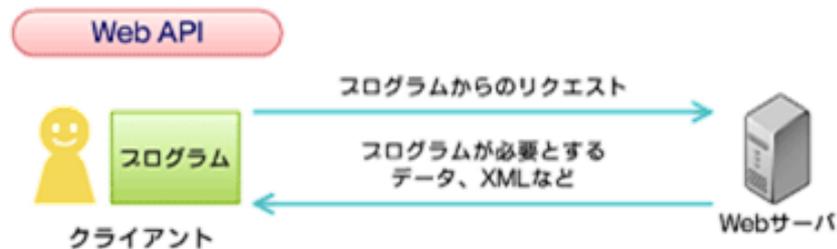
■ WebAPIアクセスを行うアプリケーション

• WebAPIとは？

- コンピュータプログラムの提供する機能を外部の別のプログラムから呼び出して、利用するための手順・規約
HTTPなどWebの技術を使用して構築されたもの

• RESTサービスとは？

• JSONとは？



■ WebAPIアクセスを行うアプリケーション

- WebAPIとは？
- **RESTサービスとは？** (REpresentational State Transfer)
 - Webサービスの設計モデルの1つで、ネットワーク上のデータ（リソース）を一意的なURIで表す
 - サービスのURIにHTTPメソッドでアクセスすることでデータの送受信を行う
 - パラメータを指定して特定のURIにアクセスするとJSON(またはXML)で応答される
- JSONとは？



■ WebAPIアクセスを行うアプリケーション

- WebAPIとは？
- RESTサービスとは？
- **JSONとは？** (JavaScript Object Notation)
 - テキストベースで軽量なデータ交換フォーマット
 - キーと値のペアでデータを管理
({"key": "Value"})
 - 配列の使用もできる ([]で表現)
- 今回はこれらをコンポーネントの定義から行うのではなく、簡単に利用できる方法をご紹介します！

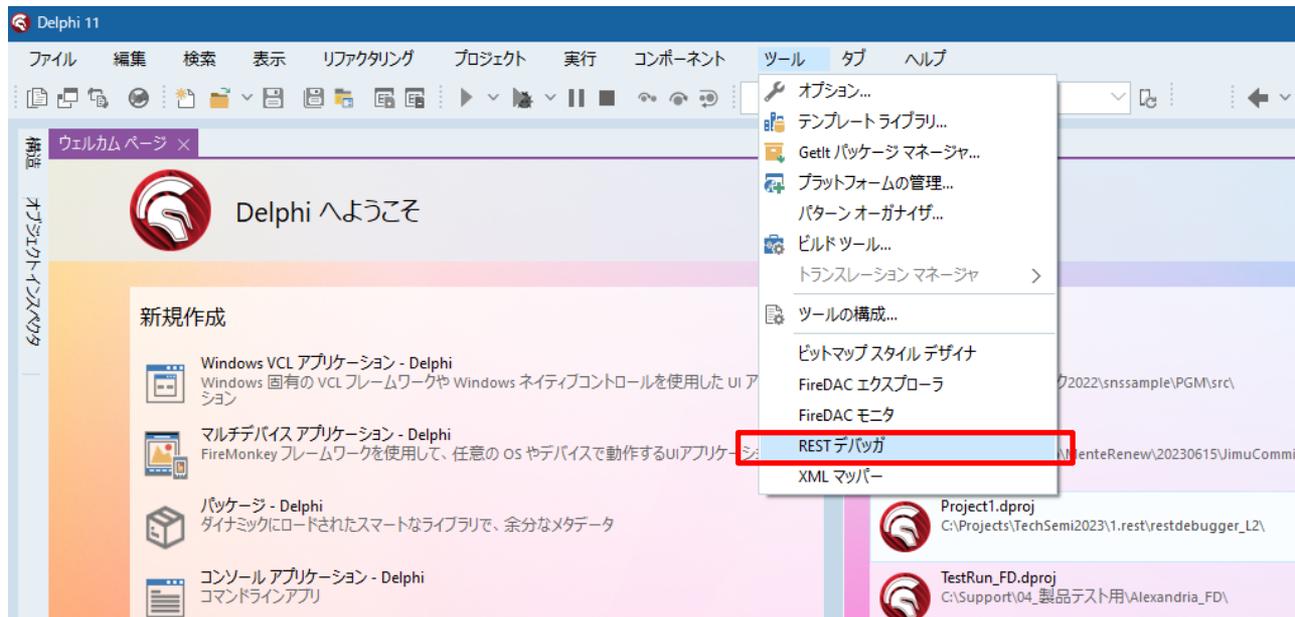
JSON例

```
{  
  "result": [  
    {  
      "subject": "english",  
      "score": 80,  
      "state": "good"  
    },  
    {  
      "subject": "math",  
      "score": 70,  
      "state": "normal"  
    }  
  ]  
}
```



■ WebAPIアクセスを行うアプリケーション

- RESTデバッガを起動



■ WebAPIアクセスを行うアプリケーション

- 郵便番号検索APIを使用



郵便番号データ配信サービス
zipcloud
powered by Google App Engine

トップ > 郵便番号検索API

郵便番号検索API

郵便番号検索APIとは

郵便番号検索APIは、日本郵便が公開している郵便番号データを検索する機能をRESTで提供しています。現在使用しているデータは、「2023年10月31日更新分の全国一括データ（加工済バージョン）」です。

» 全国一括データ（加工済バージョン）とは

リクエストURL

ベースとなるURLは以下になります。

`https://zipcloud.ibsnet.co.jp/api/search`

このURLにリクエストパラメータを加え、リクエストを行います。

リクエストパラメータ

パラメータ名	項目名	必須	備考
zipcode	郵便番号	○	7桁の数字。ハイフン付きでも可。完全一致検索。
callback	コールバック関数名	-	JSONPとして出力する際のコールバック関数名。UTF-8でURLエンコードした文字列。
limit	最大件数	-	同一の郵便番号で複数件のデータが存在する場合に返される件数の上限値（数字） ※デフォルト：20

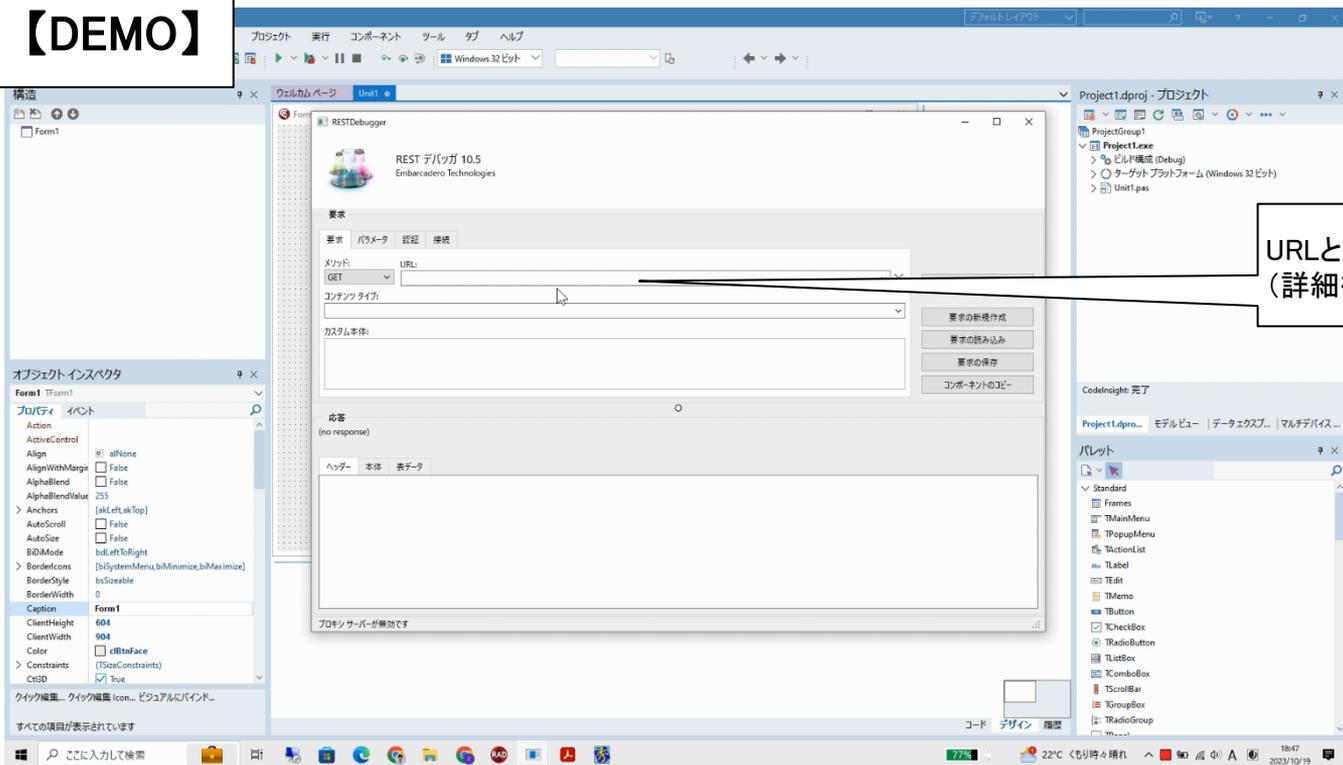
【例】郵便番号「7830060」で検索する場合
`https://zipcloud.ibsnet.co.jp/api/search?zipcode=7830060`

【郵便番号検索API】

<https://zipcloud.ibsnet.co.jp/>

■ WebAPIアクセスを行うアプリケーション

【DEMO】



URLとパラメータを設定
(詳細後述)



WebAPIアクセスを行うアプリケーション

【DEMO】

REST デバッガ 10.5
Embarcadero Technologies

概要

メソッド: GET
URL: https://zipcloud.ibsnet.co.jp/api/search

コンテナタイプ:

リクエスト本文:

応答
https://zipcloud.ibsnet.co.jp/api/search?zipcode=5560017
200: OK - 返されたデータは 302 バイト、計測情報: 着発時間: 0 ミリ秒、実行: 438 ミリ秒、後処理: 140 ミリ秒、合計: 578 ミリ秒

ヘッダー 本体 表データ

内容が有効な JSON です

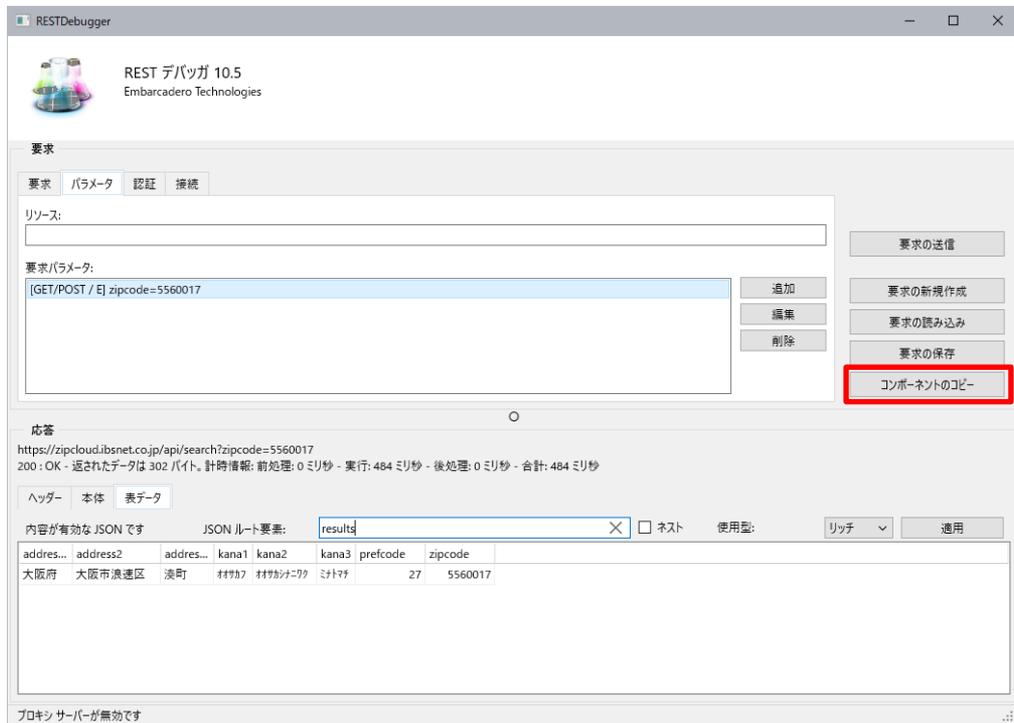
```
JSON ルート要素:  
{  
  "message": null,  
  "results": [  
    {  
      "address": "大医所",  
      "address": "大医所最寄区",  
      "address": "大医所",  
      "kana": "ダイショ",  
      "kana": "ダイショウ",  
      "kana": "ダイショウ" } ]  
}
```

要求の送信を押すと、
結果がJSONで取得される！



■ WebAPIアクセスを行うアプリケーション作成

- この状態のコンポーネントをコピーできる



次はここを押すと……

■ WebAPIアクセスを行うアプリケーション作成

- この状態のコンポーネントをコピーできる



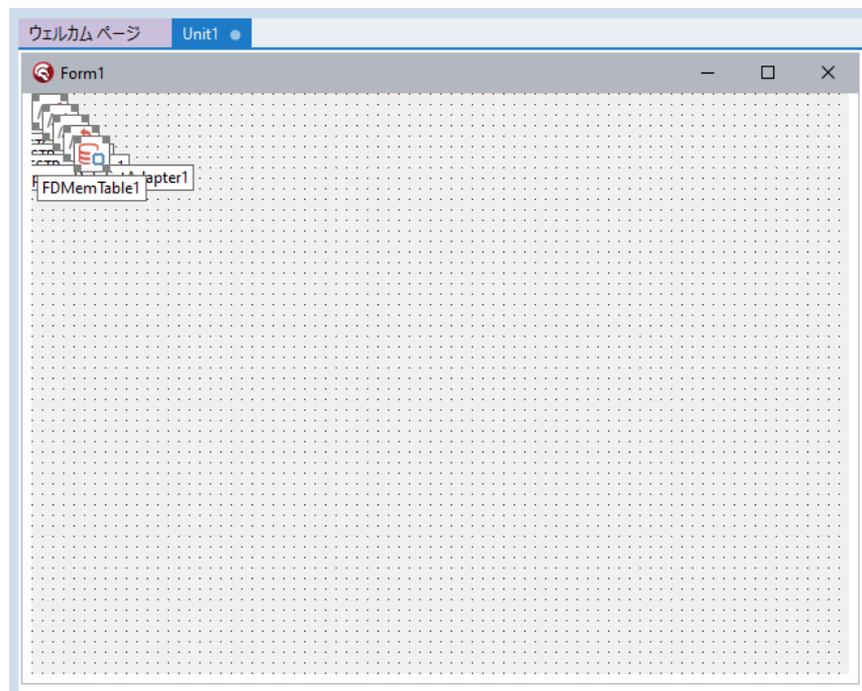
ちゃんとDelphiのコンポーネント情報がクリップボードに入っている

```
1 object TRESTClient1: TRESTClient+
2   BaseURL = 'https://zipcloud.ibsnet.co.jp/api/search'+
3   Params = O+
4   SynchronizedEvents = False+
5 end+
6 object TRESTRequest1: TRESTRequest+
7   AssignedValues = [nConnectTimeout, nReadTimeout]+
8   Client = TRESTClient1+
9   Params = C+
10  item+
11    Name = 'zipcode'+
12    Value = '5560017'+
13  end+
14  Response = TRESTResponse+
15  SynchronizedEvents = False+
16 end+
17 object TRESTResponse1: TRESTResponse+
18   RootElement = 'results'+
19 end+
20 object TRESTResponseDataSetAdapter1: TRESTResponseDataSetAdapter+
21   Dataset = TFDMemTable+
22   FieldDefs = O+
23   Response = TRESTResponse+
24   TypesMode = Rich+
25 end+
26 object TFDMemTable1: TFDMemTable+
27   FetchOptions.AssignedValues = [evMode]+
28   FetchOptions.Mode = fast+
29   ResourceOptions.AssignedValues = [nSilentMode]+
30   ResourceOptions.SilentMode = True+
31   UpdateOptions.AssignedValues = [uUpdateOnFields, uUpdateMode, uLockMode, uLockPoint, uLockWait, uRefreshMode, uFetchGeneratorsPoint, uCheckRequired, uO+
32   UpdateOptions.LockWait = True+
33   UpdateOptions.FetchGeneratorsPoint = gNone+
34   UpdateOptions.CheckRequired = False+
35 end+
36 =
```



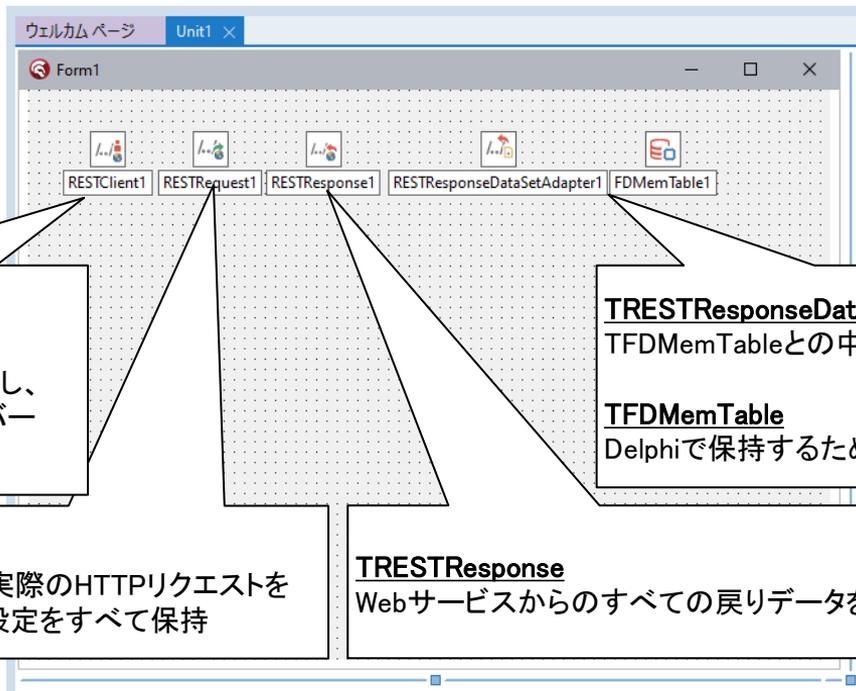
■ WebAPIアクセスを行うアプリケーション作成

- コピーしたコンポーネントを設計画面にペースト



■ WebAPIアクセスを行うアプリケーション作成

- 郵便番号WebAPIを参照するFireDACプログラム作成



TRESTClient

Webサービスへのリクエストを実行
サービスに対する HTTP 接続を管理し、
HTTPヘッダーおよびプロキシ サーバー
を処理し、応答データを受け取る

TRESTRequest

Webサービスに対する実際のHTTPリクエストを
形成するパラメータや設定をすべて保持

TRESTResponse

Webサービスからのすべての戻りデータを保持

TRESTResponseDataSetAdapter

TFDMemTableとの中継役

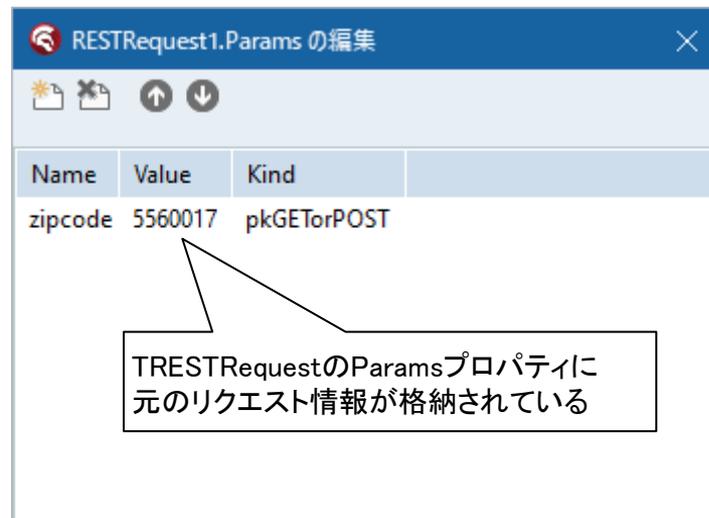
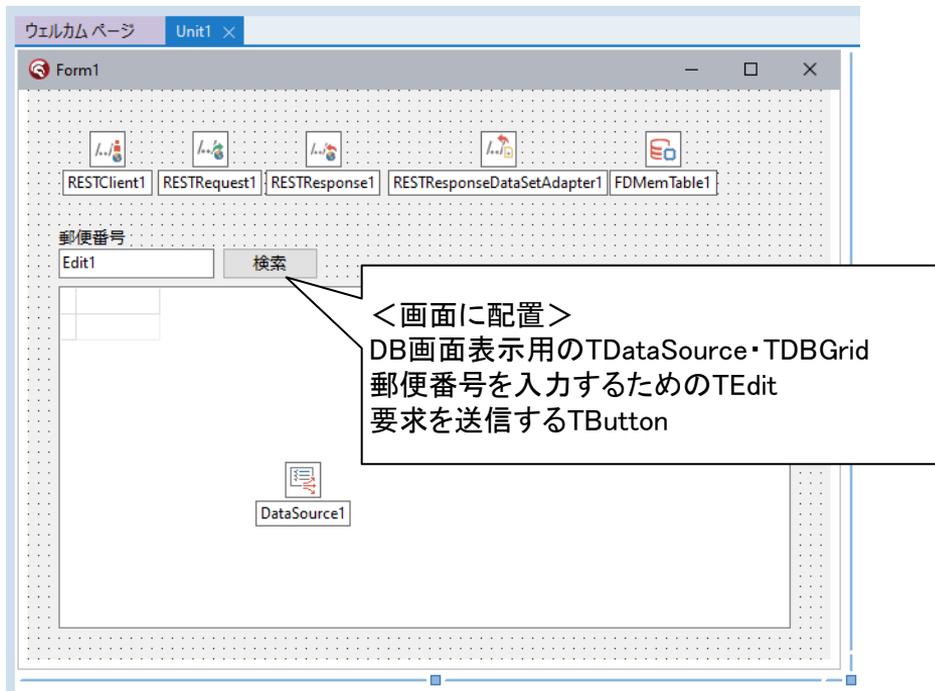
TFDMemTable

Delphiで保持するためのデータセット



■ WebAPIアクセスを行うアプリケーション作成

- 郵便番号WebAPIを参照するFireDACプログラム作成



■ WebAPIアクセスを行うアプリケーション作成

- 郵便番号WebAPIを参照するFireDACプログラム作成

呼び出しロジック例

```
{*****}
目的: 検索ボタン クリック時処理
引数:
戻値:
*****}
procedure TForm1.Button1Click(Sender: TObject);
begin
    // RESTRequestに郵便番号(7桁)を渡して検索リクエストを送信
    RESTRequest1.Params[0].Value := Edit1.Text;
    RESTRequest1.Execute;
end;
```



■ WebAPIアクセスを行うアプリケーション作成

- 郵便番号WebAPIを参照するFireDACプログラム作成

郵便番号

指定した郵便番号の住所情報が
FireDACから簡単に抽出できる！

郵便番号

元がJSONなので、
対象データが無い場合はレコードだけでなく
フィールド情報も無いデータが返る



技術Tips ピックアップ

■ 通知機能の実装方法



■ 通知機能の実装方法

• Migaro. 技術Tips

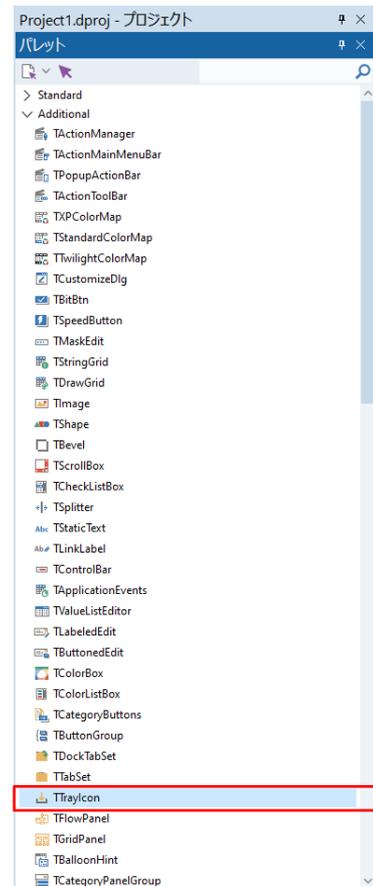
- 技術Tipsにてご紹介した項目から1つピックアップ
<https://www.migaro.co.jp/tips/2389>



■ 通知機能の実装方法

• 方法①：VCLでTTrayIconを使用

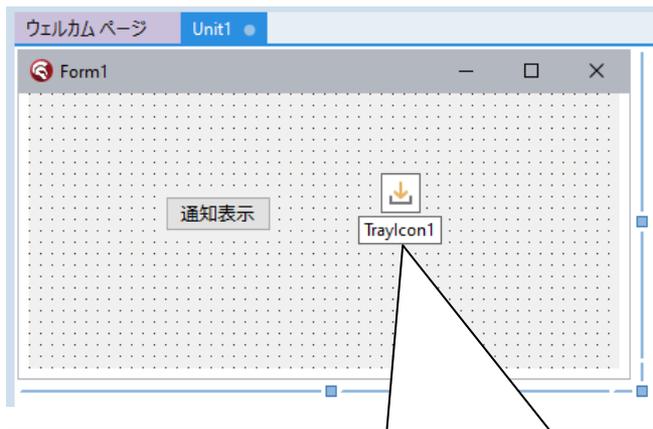
- タスクトレイの右下部分に表示されるアイコンを制御するコンポーネントだがシステムトレイから表示されるポップアップ通知機能のみを活用可能
(Delphi/400 ver2006 ~ 使用可能)



■ 通知機能の実装方法

• 方法①：VCLでTTrayIconを使用

- コンポーネントを配置し、通知表示したい場所でロジックを記述



補足：OnBalloonClickイベントを定義すると、ポップアップがクリックされた時の処理を記述可能

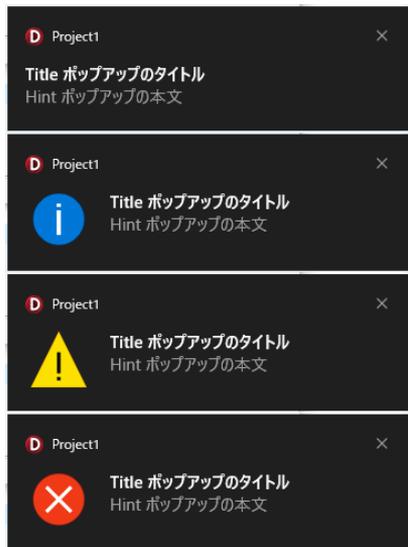
```
{*****  
目的： 通知表示ボタン クリック時処理  
引数：  
戻値：  
*****}  
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    TrayIcon1.Visible      := True; // トレイアイコン表示 (True必須)  
    TrayIcon1.BalloonFlags := bfInfo; // アイコンの種類  
    TrayIcon1.BalloonTitle := 'Title ポップアップのタイトル'; // メッセージ設定1  
    TrayIcon1.BalloonHint  := 'Hint ポップアップの本文'; // メッセージ設定2  
    TrayIcon1.ShowBalloonHint; // メッセージ表示  
end;
```



■ 通知機能の実装方法

• 方法①：VCLでTTrayIconを使用

- BalloonFlagsパラメータを設定すると、アイコンの種類を選べる



bfNone

bfInfo

bfWarning

bfError

BalloonTitleプロパティはポップアップのタイトルを設定
BalloonHintプロパティはポップアップの本文を設定

※BalloonTitleがblankの場合、BalloonHintの値がタイトルとして表示
※BalloonHintがblankの場合はポップアップが表示されません

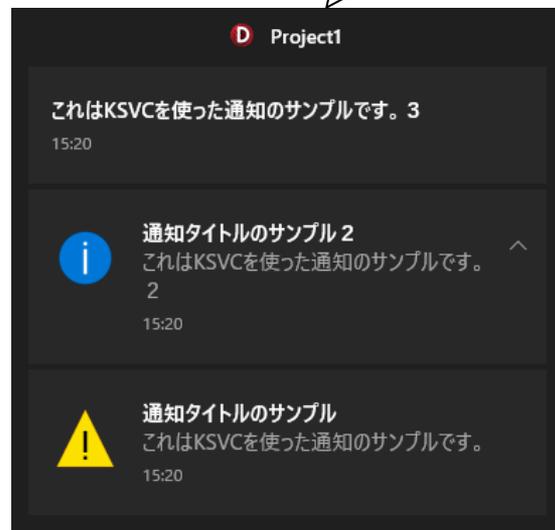


■ 通知機能の実装方法

• 方法② : Bonus KSVC使用

- Bonus KSVCに含まれている TRzTrayIconコンポーネントを使用すると、より簡単なメソッドのみで表示可能
また、複数のポップアップを連続表示可能
(Delphi 10.2 Tokyo ~ 使用可能)
- ※この手順の詳細はこちらのTipsを参照
<https://www.migaro.co.jp/tips/2389/>
- ※Bonus KSVCについてははこちらのTipsを参照
<https://www.migaro.co.jp/tips/1997/>

連続表示させたイメージ



■ 通知機能の実装方法

- 方法③ : FireMonkeyでTNotificationを使用
 - スマートデバイスで通知を表示させたい場合はコレ！
(Delphi 10 Seattle ~ 使用可能)



TNotificationCenterコンポーネントを使用

※VCLにもTnotificationを使って通知は表示できるが、アイコンの設定ができないぶん方法①や方法②の方が有利なため、ここでは割愛



■ 通知機能の実装方法

• 方法③ : FireMonkeyでTNotificationを使用

• Tnotificationの生成・実行ロジック

IncSecondを使用する場合は、uses節にSystem.DateUtilsを追加
iOS・Androidでも「PresentNotification」で通知を即時表示させることは可能

```
{*****  
目的: 検証用 プッシュ通知  
*****}  
procedure TForm1.Button2Click(Sender: TObject);  
var  
    MyNotification: TNotification;  
begin  
    // 通知用クラスを生成  
    MyNotification := NotificationCenter1.CreateNotification;  
    try  
        // 通知内容を設定  
        MyNotification.Name      := 'MIGARO TIPS APP';  
        MyNotification.Title     := '技術TIPS 通知のタイトル';  
        MyNotification.AlertBody := '通知をお知らせします。' + #13#10 +  
            '株式会社ミガロ.の各製品ページへの短縮URLができました!' +  
            '「https://www.migaro.co.jp/」のあとに「valence」「delphi400」 +  
            '「sp4i」と入力すると、それぞれの製品ページへジャンプするよ!!!';  
    // (右へ続く)
```

```
// (左から続く)  
{$IFDEF IOS} // iOS向け処理  
    // 5秒後に通知を表示させる  
    MyNotification.FireDate := IncSecond(Now, 5);  
    NotificationCenter1.ScheduleNotification(MyNotification);  
{$IFDEF Android} // Android向け処理  
    // 5秒後に通知を表示させる  
    MyNotification.FireDate := IncSecond(Now, 5);  
    NotificationCenter1.ScheduleNotification(MyNotification);  
{$ELSE} // それ以外(Windows)向け処理  
    // 通知を表示させる(予約は非対応)  
    NotificationCenter1.PresentNotification(MyNotification);  
{$ENDIF}  
{$ENDIF}  
finally  
    FreeAndNil(MyNotification);  
end;  
end;
```

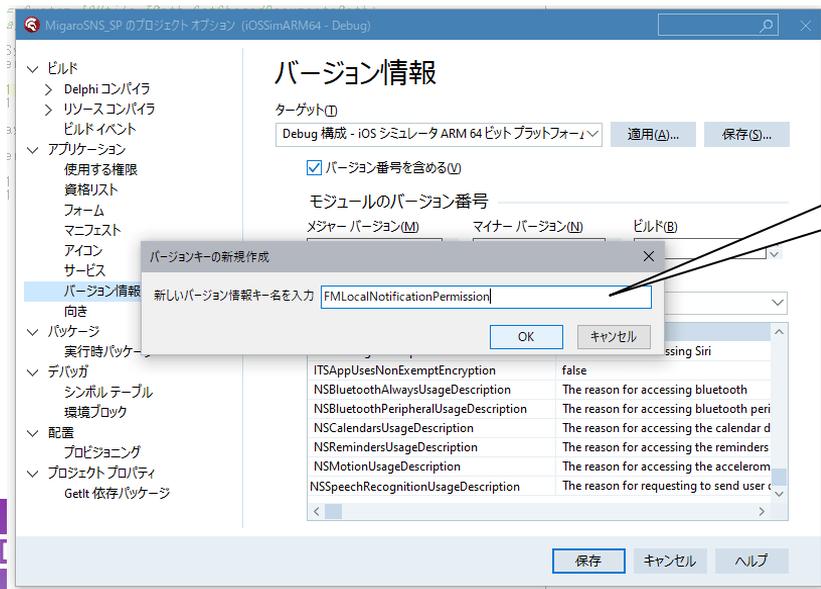


■ 通知機能の実装方法

• 方法③：FireMonkeyでTNotificationを使用

• iOSでは追加手順が必要

- 1. プロジェクトオプションで、新しいバージョンキーを作成
- 2. 事前にロジックで通知の許可を要求



新しいバージョンキー
「FMLocalNotificationPermission」を作成し、
値に「true」を設定

未設定だと
こんなエラーが出る

エラー
通知センターのオペレーションを実行
できません。(RequestPermission) メ
ソッドを使用して権限を要求する必要
があります。 .

OK

■ 通知機能の実装方法

- 方法③ : FireMonkeyでTNotificationを使用
 - iOSでは追加手順が必要
 - 1. プロジェクトオプションで、新しいバージョンキーを作成
 - 2. 事前にロジックで通知の許可を要求

このメソッドを通知表示ロジックより前
(アプリ起動時など)に実行しておく

```
// 通知の許可要求  
NotificationCenter1.RequestPermission;
```



■ 通知機能の実装方法

• 方法③ : FireMonkeyでTNotificationを使用

• 通知の実行イメージ



サポートご質問ランキング上位

■ 複数バージョンの Delphi/400開発環境を入替する方法



■ 複数verのDelphi/400開発環境を入替する方法

- バージョンアップ作業の並行稼働中などでは、同一の開発（運用）端末で複数バージョンのDelphi/400を動かす場合がある



■ 複数verのDelphi/400開発環境を入替する方法

- クライアントPCでは
複数バージョンのDelphi/400は共存できない
(※開発ツール本体であるDelphiは共存可能)

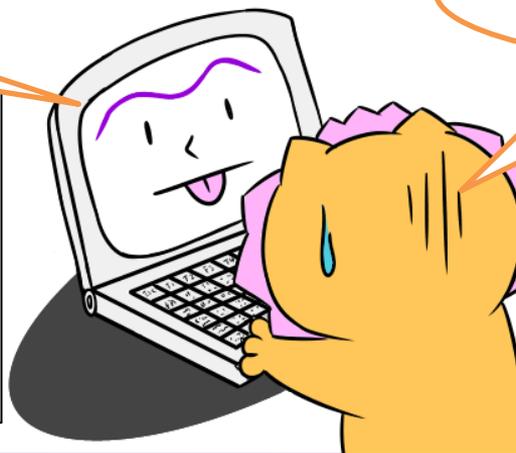
V2010が入っている状態で
11 Alexandriaは動作しません
(逆も同様)

一度アンインストールして
入れ替えるミン...

元のバージョンをアンインストールせずに別のバージョンの
Delphi/400を上書きでインストールした場合、

- ・DLLが同じパスに上書きされたり、
INIファイルの設定値が不正になるため、正常に動作しない
- ・『序数XXXがXXXX.DLLから見つかりませんでした』
といったエラーが発生する

といった事象が発生することがある



■ 複数verのDelphi/400開発環境を入替する方法

- Delphi/400 内部構造の変遷

	V5	V6~V2005	V2006~XE	XE3~XE7	10 Seattle~ 11 Alexandria
設定フォルダ	C:¥co4XX (XX部分はバージョンごとに異なる)				
設定ファイル	C:¥Windows¥win.ini			C:¥co4XX¥CO400.ini	
対応bit数	32bitアプリケーションのみ			32bit・64bit	
dbExpress	なし	DBC0400.DLL	DBC0430.DLL		
FireDAC	なし				FDC0400.DLL
レジストリ	バージョンごとに異なる (詳細後述)				



■ 複数verのDelphi/400開発環境を入替する方法

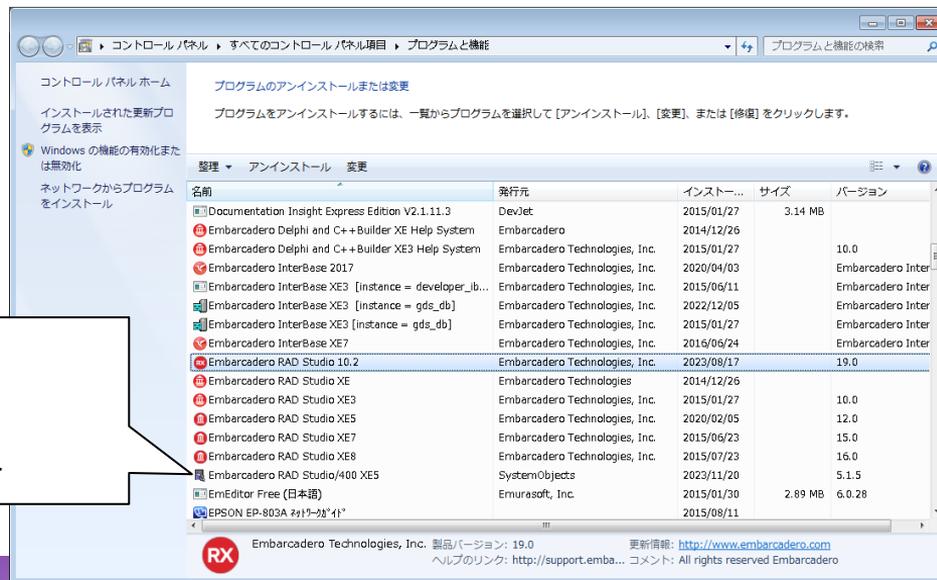
- 手順①：移行元のバージョンをアンインストール

バージョンによって名称が異なるが、基本的には「Delphi/400」「Embarcadero」「RadStudio」のいずれかで始まっている



■ 複数verのDelphi/400開発環境を入替する方法

- 手順①：移行元のバージョンをアンインストール
 - 開発版の場合、
本体となるDelphi (RAD Studio) はアンインストール不要



Delphi (Embarcadero RAD Studio) は
複数バージョン共存可能！

※ただしDelphi/400は同時に1バージョンのみ

■ 複数verのDelphi/400開発環境を入替する方法

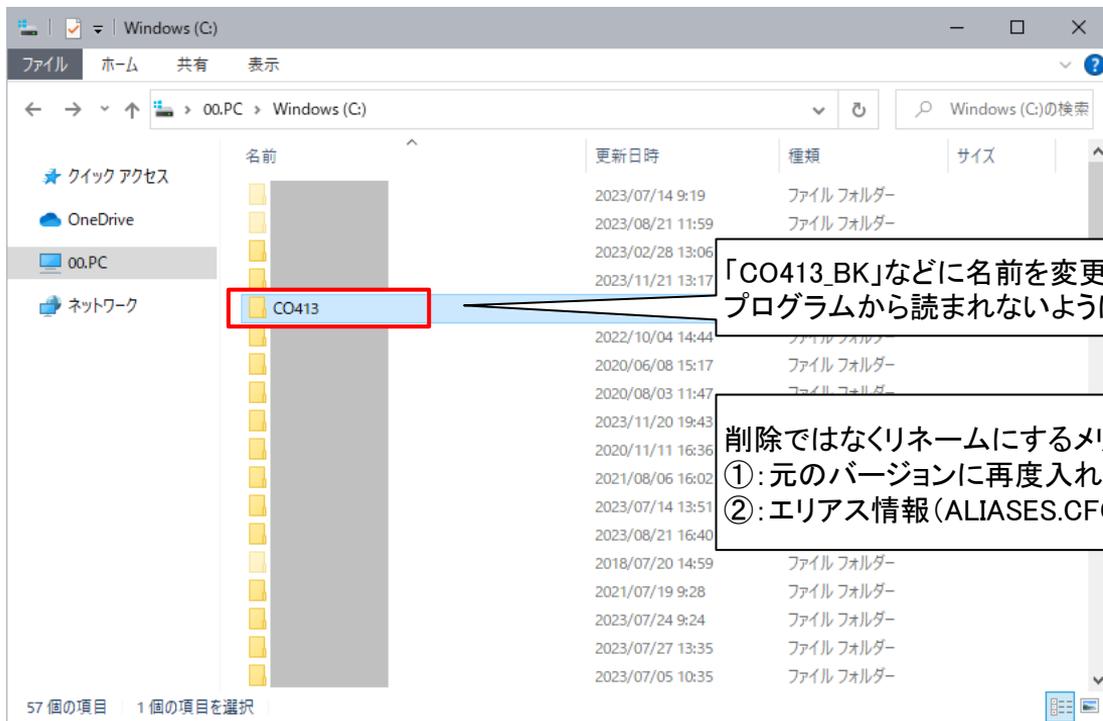
- 手順②：設定フォルダの削除またはリネーム

	V5	V6~V2005	V2006~XE	XE3~XE7	10 Seattle~ 11 Alexandria
設定フォルダ	C:¥co4XX (XX部分はバージョンごとに異なる)				
設定ファイル	C:¥Windows¥win.ini			C:¥co4XX¥CO400.ini	
対応bit数	32bitアプリケーションのみ			32bit・64bit	
dbExpress	なし	DBC0400.DLL	DBC0430.DLL		
FireDAC	なし				FDC0400.DLL
レジストリ	バージョンごとに異なる (詳細後述)				



■ 複数verのDelphi/400開発環境を入替する方法

- 手順②：設定フォルダの削除またはリネーム



■ 複数verのDelphi/400開発環境を入替する方法

• 手順③：関連BPLやDLLの削除

- 必須手順ではないが、古いバージョンに「バージョンダウン」の入替を行う場合は念のためにチェック

32ビットアプリケーション用

(64bit端末:「C:¥Windows¥SysWOW64」 32bit端末:「C:¥Windows¥System32」)

E32TCPIP.DLL

TCRTR32.DLL

CO400LOC.DLL

CO40032.DLL

IDCO400.DLL

ODCO400.DLL

ODCOCFG.DLL

CO400NET.DLL

DBC0400.DLL

DBC0430.DLL

FDCO400.DLL

co400res.dll

SCD400NET.dll

「combo400」で始まるBPI・BPL・DCPファイル

「dclscd400」で始まるBPI・BPL・DCPファイル

「FDCO400」で始まるBPI・BPL・DCPファイル

「scd400」で始まるBPI・BPL・DCPファイル

バージョンによっては無いファイルもあり
具体的なファイルの一覧は、技術Tipsの
<https://www.migaro.co.jp/tips/1438/> にも記載



■ 複数verのDelphi/400開発環境を入替する方法

• 手順③：関連BPLやDLLの削除

- 必須手順ではないが、古いバージョンに「バージョンダウン」の入替を行う場合は念のためにチェック

64ビットアプリケーション用
(64bit端末:「C:¥Windows¥System32」 32bit端末:対象外)

CO400NET.DLL
CO400LOC.DLL
TCRTR32.DLL
E32TCPIP.DLL
DBC0430.DLL
FDGO400.DLL
「scd400」で始まるBPI・BPL・DCPファイル

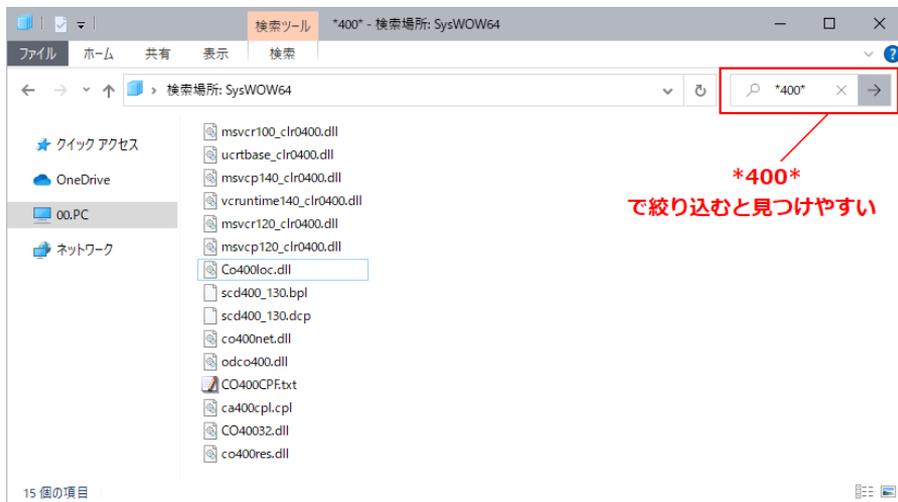
古いWindows 7や一部のWindows 10には
まだ32bit端末が残っている場合あり
「SysWOW64」フォルダの有無や中身で判別できる



■ 複数verのDelphi/400開発環境を入替する方法

• 手順③：関連BPLやDLLの削除

- これらの対象ファイルは、System32やSysWOW64で「*400*」で絞り込むと見つけやすい（DBCO430.DLLなど一部を除く）



■ 複数verのDelphi/400開発環境を入替する方法

- 手順④：設定INIファイルのクリア（XE以前必須）

	V5	V6~V2005	V2006~XE	XE3~XE7	10 Seattle~ 11 Alexandria
設定フォルダ	C:¥co4XX（XX部分はバージョンごとに異なる）				
設定ファイル	C:¥Windows¥win.ini			C:¥co4XX¥CO400.ini	
対応bit数	32bitアプリケーションのみ			32bit・64bit	
dbExpress	なし	DBC0400.DLL	DBC0400.DLL		
FireDAC	なし				FDC0400.DLL
レジストリ	バージョンごとに異なる（詳細後述）				

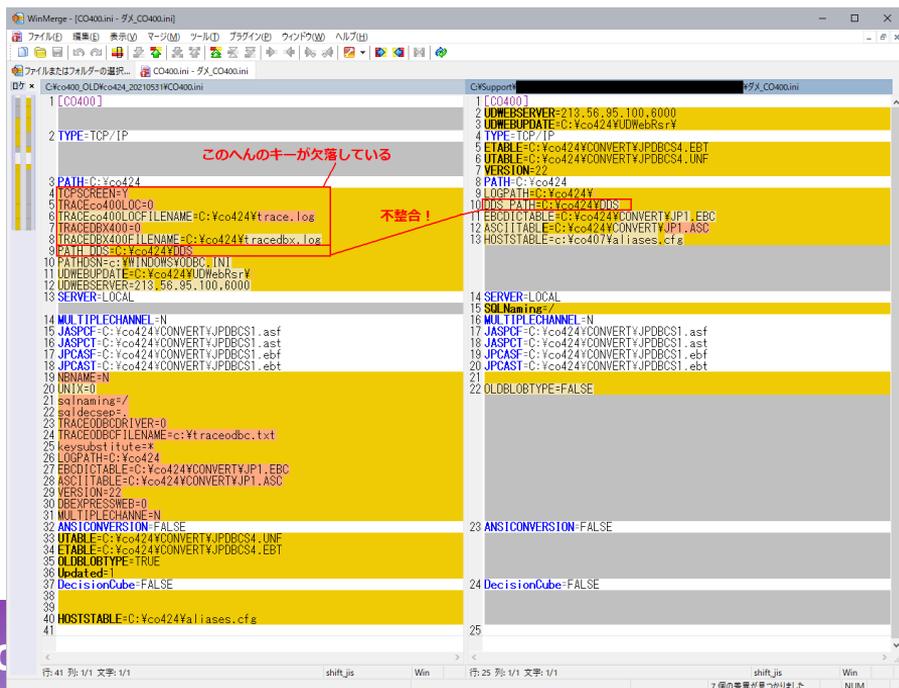
UAC配下にあるC:¥Windowsから設定ファイルを外に出すことになった



■ 複数verのDelphi/400開発環境を入替する方法

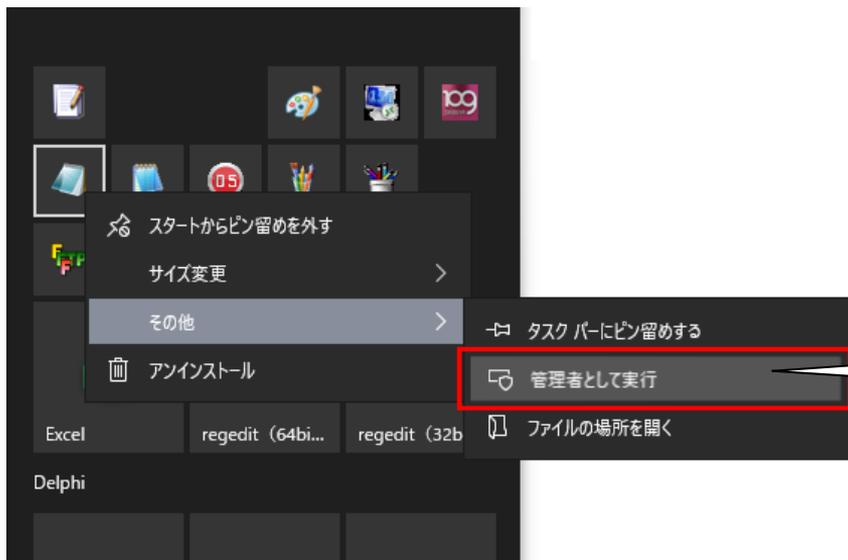
● 手順④：設定INIファイルのクリア（XE以前必須）

- この手順を飛ばしてしまうと、インストール後のCO400.iniの設定値に不整合が起こり、正しく動作しなくなる
- XE3～ではwin.iniを使用していないため、この手順は不要



■ 複数verのDelphi/400開発環境を入替する方法

- 手順④：設定INIファイルのクリア（XE以前必須）
 - メモ帳などテキストエディタを管理者権限で実行



管理者権限で実行していないと、
C:¥Windows 内のファイルを上書き保存できない

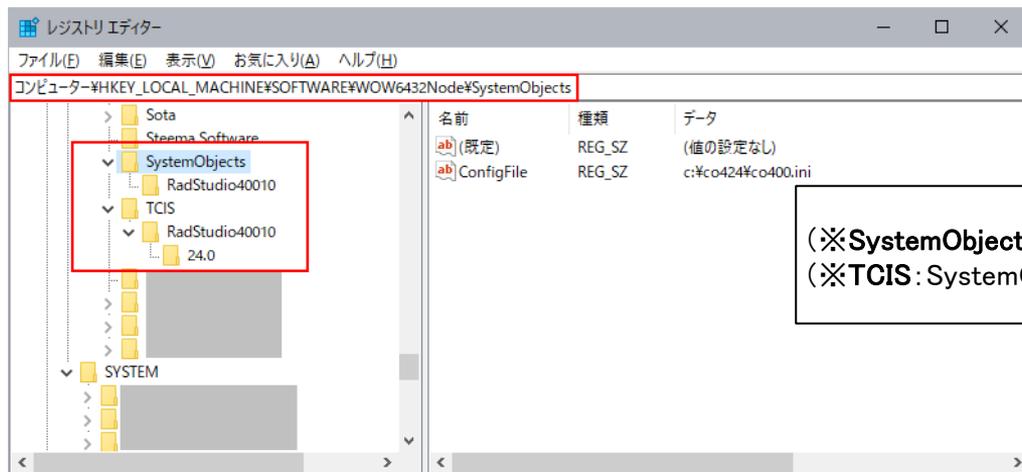


■ 複数verのDelphi/400開発環境を入替する方法

• 手順⑤：設定レジストリのクリア

※PCが32bitの場合は
「Wow6432Node」を読み飛ばす

- HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node
の中に、「SystemObjects」「TCIS」があればキーごと削除

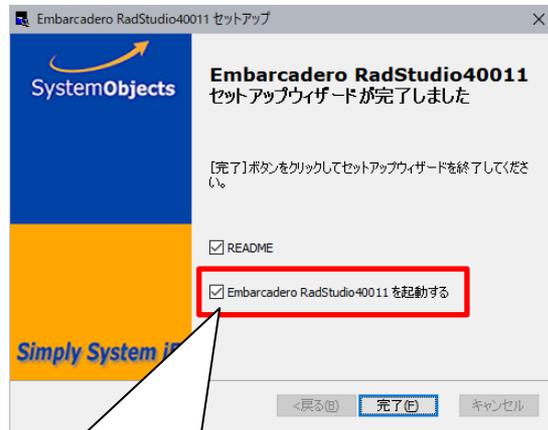
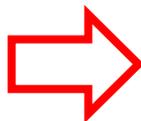
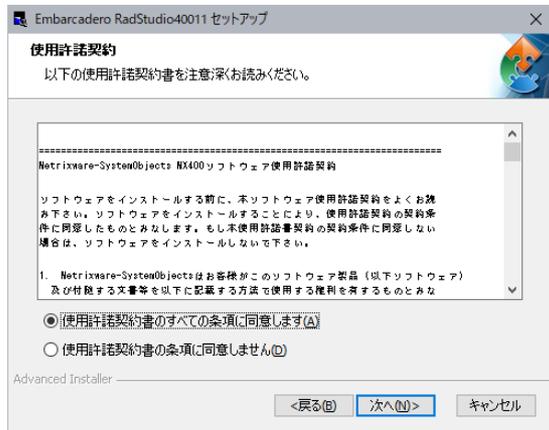


(※SystemObjects: Delphi/400 開発元の名前)
(※TCIS: SystemObjectsの前身の名前)



■ 複数verのDelphi/400開発環境を入替する方法

- 手順⑥：移行先Delphi/400をインストール

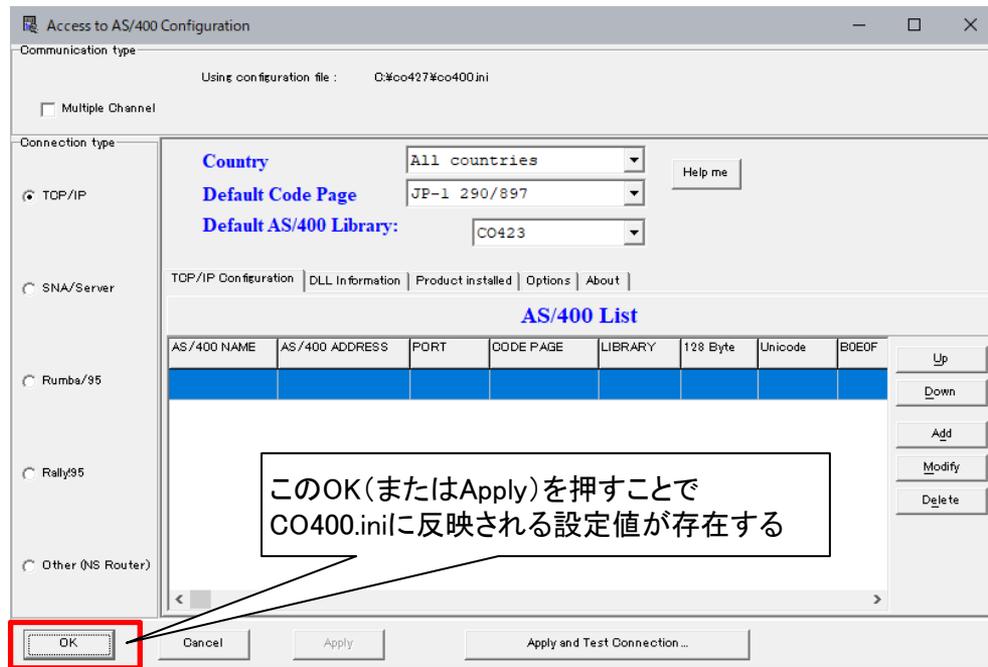


接続テスト(Configuration)は必ず起動する
※サイレントインストールの場合を含む



■ 複数verのDelphi/400開発環境を入替する方法

- 移行先Delphi/400をインストール後



■ 複数verのDelphi/400開発環境を入替する方法

- 移行先Delphi/400をインストール後

OK(設定済)

TAALBM	TAALNO	TAMJMN	TATITL	TATIRK
▶ 2	1	1	XXL	
2	2	1	感情のピクセル	
2	3	1	NATURAL LIPS	
2	4	1	HOROSCOPE	
2	5	1	まわせPDCAサイクル	
2	6	1	電車で聴くと映画の主人公になれる曲 (INTERLUDE)	
2	7	1	OPEN	
2	8	1	観察日記	
2	9	1	SNACK	
2	10	1	鶴川等間隔	
2	11	1	式	
3	1	1	NO TOUCH SERVICE ACE	
3	2	1	からだ	
3	3	1	PTA	
3	4	1	弱者	
3	5	1	なごをやってもらかんわ	
3	6	1	確実こ2分で眠れる睡眠音楽 (INTERLUDE)	
3	7	1	CALCULATED	
3	8	1	OKAZAKI UNREAL HYPOTHESIS	
3	9	1	JACK FROST	
3	10	1	私生活	
3	11	1	龍	
3	12	1	THE ABYSS	
4	1	1	CHAMPIONSHIP	
4	2	1	おっさん	
4	3	1	HOBBY	

NG(未設定)

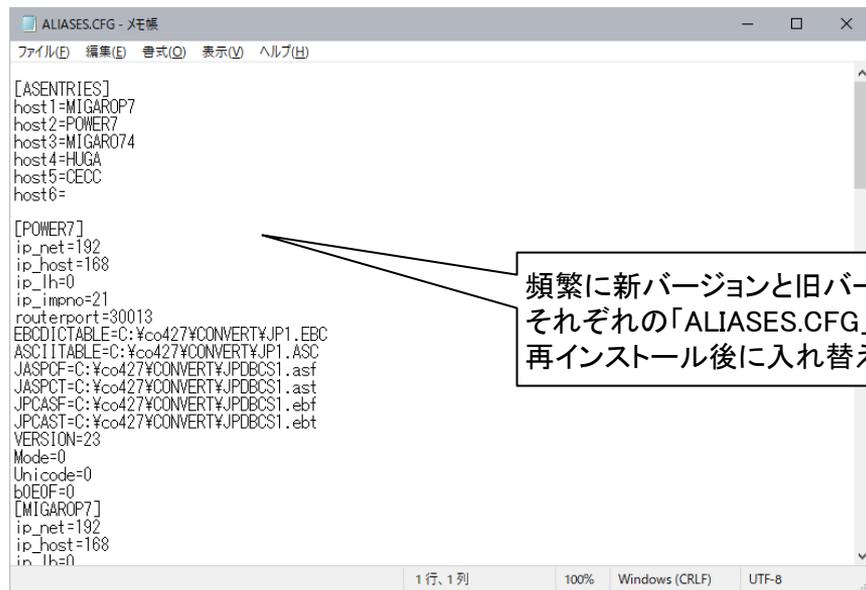
TAALBM	TAALNO	TAMJMN	TATITL	TATIRK
▶ 2	1	1	XXL	
2	2	1		
2	3	1	NATURAL LIPS	
2	4	1	HOROSCOPE	
2	5	1	PDCA	
2	6	1		(INTERLUDE)
2	7	1	OPEN	
2	8	1		
2	9	1	SNACK	
2	10	1		
2	11	1		
3	1	1	NO TOUCH SERVICE ACE	
3	2	1		
3	3	1	PTA	
3	4	1		
3	5	1		
3	6	1	2	(INTERLUDE)
3	7	1	CALCULATED	
3	8	1	OKAZAKI UNREAL HYPOTHESIS	
3	9	1	JACK FROST	
3	10	1		
3	11	1		
3	12	1	THE ABYSS	
4	1	1	CHAMPIONSHIP	

未設定のままだと、
全角文字が空白で読み込まれる



■ 複数verのDelphi/400開発環境を入替する方法

- 移行先Delphi/400をインストール後
 - ALIASES.CFGファイルには IBM i 接続先情報が格納



```
[ASENTRIES]
host1=MIGAROP7
host2=POWER7
host3=MIGAR074
host4=HUGA
host5=CECC
host6=-

[POWER7]
ip_net=192
ip_host=168
ip_lh=0
ip_impro=21
routerport=30013
EBCDICTABLE=C:¥co427¥CONVERT¥JP1_EBC
ASCII1TABLE=C:¥co427¥CONVERT¥JP1_ASC
JASPCF=C:¥co427¥CONVERT¥JPDBC1.asf
JASPC1=C:¥co427¥CONVERT¥JPDBC1.ast
JPCASF=C:¥co427¥CONVERT¥JPDBC1.ebf
JPCAS1=C:¥co427¥CONVERT¥JPDBC1.ebt
VERSION=23
Mode=0
Unicode=0
bEOF=0
[MIGAROP7]
ip_net=192
ip_host=168
ip_lh=0
```

頻りに新バージョンと旧バージョンの入替を行う場合は、それぞれの「ALIASES.CFG」ファイルを保持しておいて再インストール後に入れ替えればOK



以前のセミナーで
ご紹介した件の続報

■ Edge対応の最新情報



■ Edge対応の最新情報

- 過去にもEdge対応についてご案内してきた

■ 新しいVCLコンポーネント

• TEdgeBrowserコンポーネント

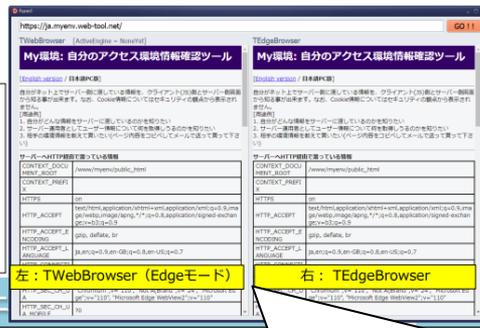
- 従来のTWebBrowserは古いIEベースでWebページを表示していたが、TEdgeBrowserを使えばChromiumのEdgeベースで表示させることが可能
- 従来のTWebBrowserにも、Edgeモードが追加された

<ご注意>

TEdgeBrowserやEdgeモードを有効にする条件

- GetIt パッケージマネージャより『EdgeView2 SDK』を取得し、その中にあるEXEと同じbit数の『WebView2Loader.dll』をEXEと同階層に配置
- Canary Channel VersionのEdgeをインストール

※この条件はFireMonkeyのTWebBrowserでEdgeモードを使用する際も同様



第27回 ミガロ。Delphi/400テクニ

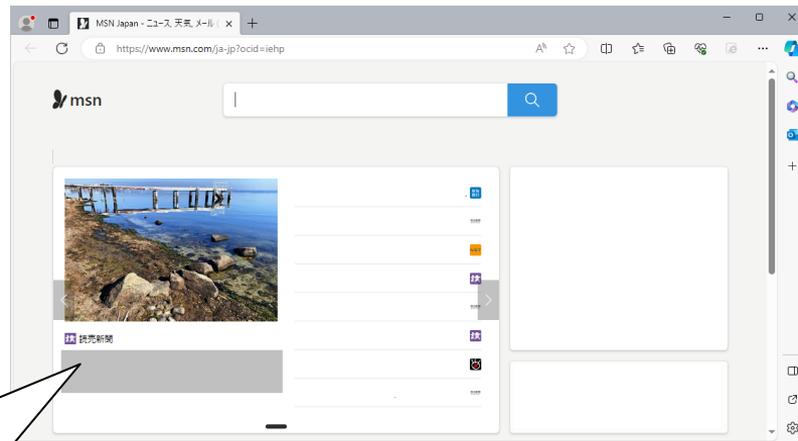
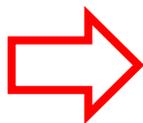
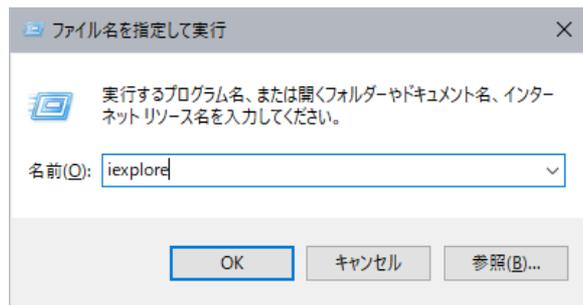
2022年テクニカルセミナーより

<https://www.migaro.co.jp/ts/27th/Session1.pdf>



■ Edge対応の最新情報

- 現在はIEを直接起動できなくなっている

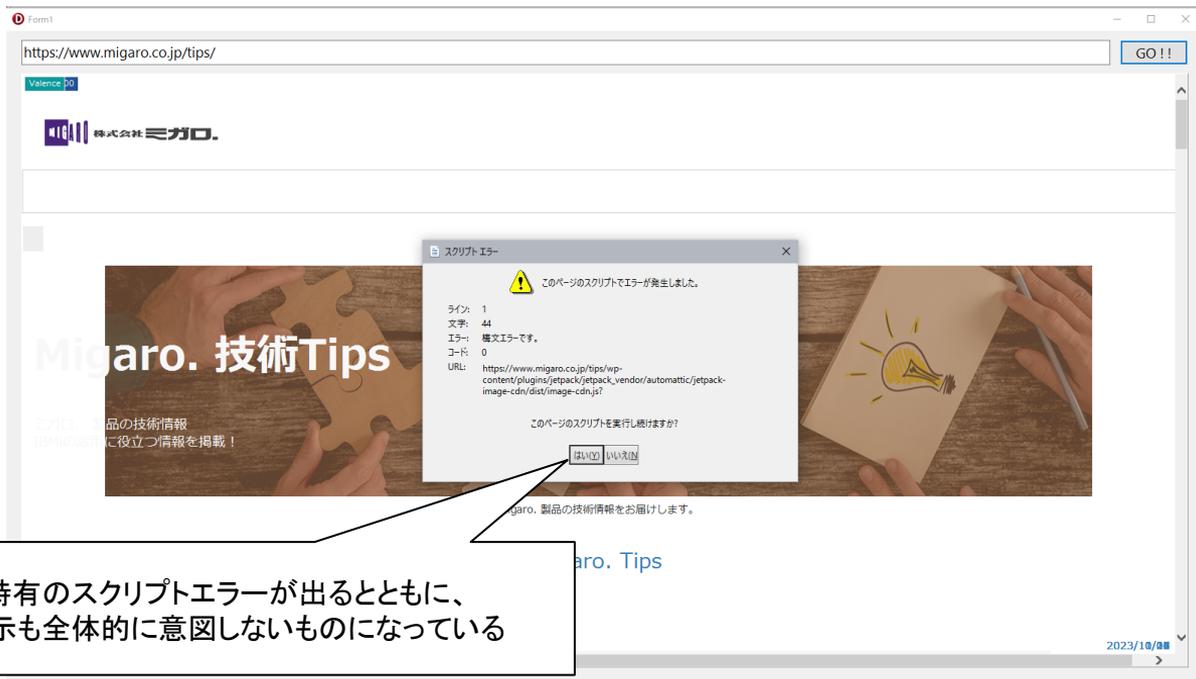


現在は「iexplore」を直接実行しても
起動したIEが閉じてEdgeにリダイレクトするようになった



■ Edge対応の最新情報

- DelphiのTWebBrowserは内部的にIEで動いている



■ Edge対応の最新情報

- DelphiのTWebBrowserは内部的にIEで動いている



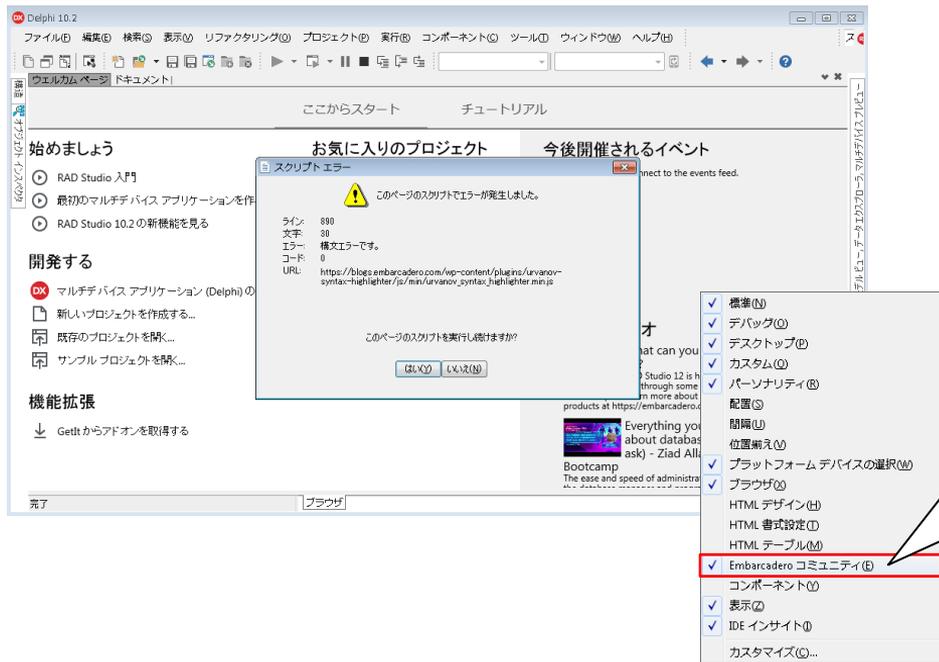
アクセス環境を調査するサイトで調べるとIE7を示す『MSIE 7.0』と表示されている

最新のIE11として実行し続ける
<https://www.migaro.co.jp/tips/1101/>
というTipsは存在するが、今後保証はされない



■ Edge対応の最新情報

• (補足) Delphi IDE起動時のスクリプトエラー回避手順



① 対象バージョン: XE7・10 Seattle

原因: ウェルカムページがHTMLで生成されており、埋め込みのIEブラウザで表示されるため

回避手順: ウェルカムページを非表示の設定でIDEを起動する

<https://www.migaroco.jp/tips/41/>

② 対象バージョン: 10 Seattle・10.2 Tokyo

原因: ツールバーの「Embarcadero コミュニティ」という項目で、IEブラウザが埋め込まれているため

回避手順: ツールバーを右クリックして一覧から「Embarcadero コミュニティ」を非表示に設定する

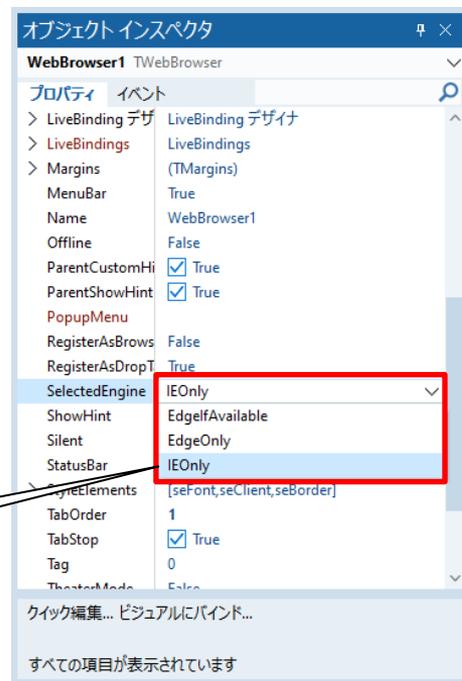
(※レジストリの設定で完全に消すことも可能)



■ Edge対応の最新情報

• Delphi内のブラウザをEdge対応させる条件

- Delphi/400 11 Alexandriaでコンパイル
- ①TEdgeBrowserを使用する
- ②TWebBrowserを継続して使用する場合、SelectedEngineプロパティを「EdgeOnly」または「EdgeIfAvailable」に設定



EdgeIfAvailable 対応していればEdgeで動作
EdgeOnly 常にEdgeで動作を試行
IEOnly 常にIEで動作させる(初期値)



■ Edge対応の最新情報

- Delphi内のブラウザをEdge対応させる手順
 - 『WebView2Loader.dll』をEXEと同階層に配置する
 - 入手方法
 - Office等に付属しているファイルを探す
 - DelphiのGetIt パッケージマネージャから取得する



■ Edge対応の最新情報

- GetItからWebViewを取得

The screenshot shows the GetIt Package Manager window with the 'EdgeView2 SDK 1.0.1054.31' package selected and highlighted with a red box. The package details include the version, author (Microsoft), and a description: 'The Microsoft Edge WebView2 control enables you to host web content in your Windows app.' The installation date is listed as 24 Dec 2021. To the right, a File Explorer window shows the path 'Program Files (x86) \ Embarcadero \ Studio \ 22.0 \ Redist \ win32'. The file list includes various files, with 'WebView2Loader.dll' highlighted in blue. The file details for 'WebView2Loader.dll' are: Name: WebView2Loader.dll, Updated: 2021/11/20 19:06, Type: アプリケーション拡張, Size: 106 KB.

名前	更新日時	種類	サイズ
vclitouch280.de	2023/03/30 0:31	DE ファイル	68 KB
vclitouch280.fr	2023/03/30 0:28	FR ファイル	68 KB
vclitouch280.ja	2023/03/30 0:30	JA ファイル	67 KB
vclWinX280.bpl	2023/03/30 0:25	BPL ファイル	893 KB
vclx280.bpl	2023/03/30 0:25	BPL ファイル	275 KB
vclx280.de	2023/03/30 0:31	DE ファイル	67 KB
vclx280.fr	2023/03/30 0:28	FR ファイル	67 KB
vclx280.ja	2023/03/30 0:30	JA ファイル	67 KB
WebView2Loader.dll	2021/11/20 19:06	アプリケーション拡張	106 KB
xmlrtl280.bpl	2023/03/30 0:25	BPL ファイル	2,286 KB
xmlrtl280.de	2023/03/30 0:31	DE ファイル	62 KB
xmlrtl280.de	2023/03/30 0:28	FR ファイル	62 KB
xmlrtl280.de	2023/03/30 0:30	JA ファイル	59 KB

取得すると、
C:\Program Files (x86)\Embarcadero\Studio\22.0\Redist¥
のwin32・win64フォルダ内にDLLがダウンロードされている



■ ご清聴ありがとうございました。

