

【A5】Delphiテクニカルセッション



DEVELOPER CAMP

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 3.0

株式会社ミガロ RAD事業部
技術支援課 顧客サポート主任
吉原 泰介

アジェンダ



DEVELOPER CAMP

- ミガロについて
- VCL for the Web基礎知識
- VCL for the Webのよくある問合せQ&A
 - 【Q1】戻るボタンの対応方法
 - 【Q2】Cookieを利用した制御
 - 【Q3】IWTextの右寄せ表示
 - 【Q4】開発モードの変更方法
 - 【Q5】IIS上での動作の違い

お持ち帰り資料

- 【QA】GUIアプリからWEBサイトの呼出
- 【QB】デバッグ実行の判断

株式会社ミガロ – A 100% IBM i Company –

会社情報

所在地: 本社 大阪市浪速区湊町2-1-57
難波サンケイビル13F

東京事業所 東京都港区麻布台1-4-3
エグゼクティブタワー麻布台11F

事業内容

IBM i 向けのソフトウェア・ツール販売および技術サポート

【開発ツール】

- Delphi/400
- JACi400

【災害対策ソフト】

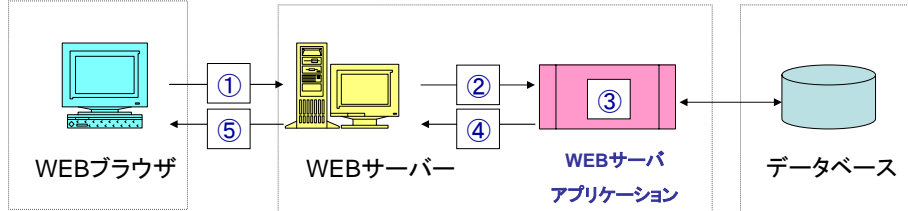
- *noMAX

【アプリケーション・ライフサイクル・マネージメント】

- MKS

- Delphi/400
- DelphiをIBM iに完全対応させたミドルウェア
- 国内約600社、全世界約5,000社の導入実績

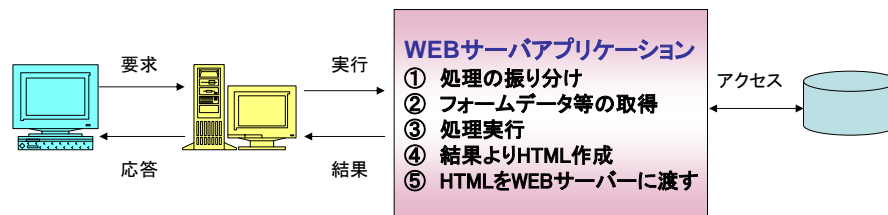
WEBサーバアプリケーションとは



処理手順

- ① WEBブラウザは、URLを指定することにより、検索等の処理などを要求
- ② WEBサーバは、指定されたURLより対象のWEBサーバアプリケーションをCALL
- ③ WEBサーバアプリケーションは処理を実行し、データ等を抽出
- ④ WEBサーバアプリケーションは処理結果をHTMLとしてWEBサーバに返す
- ⑤ WEBサーバは、リクエストしたブラウザに対し、処理結果HTMLを返す

一般的なWEBサーバアプリケーションの仕組み

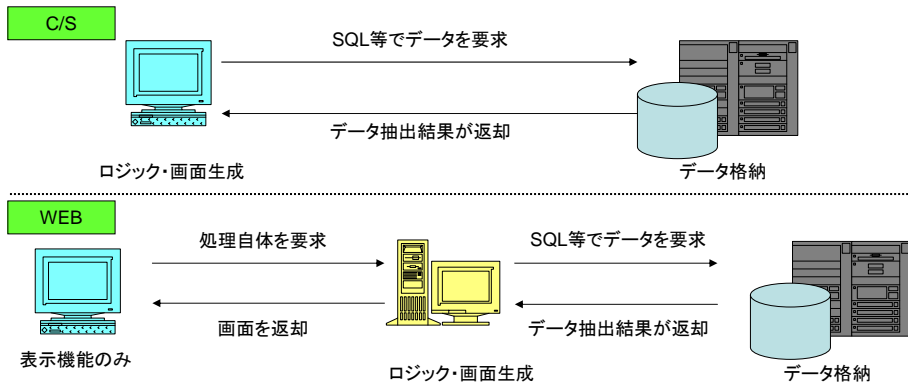


処理手順

- ① 検索処理や登録処理等の各処理をPathInfo部を使用して振り分ける
- ② GETメソッドやPOSTメソッドを使いブラウザから要求された変数を受け取る
- ③ ①と②で得られた情報より処理(データ取得など)を実行する
- ④ 処理結果を元に結果値を含むHTMLを作成する
- ⑤ HTTP応答ヘッダ及び作成したHTMLを標準出力を使ってWEBサーバに渡す

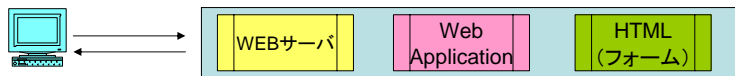
C/SアプリケーションとWEBサーバアプリケーションの違い

C/Sアプリが2層構造なのに対しWEBサーバアプリは3層構造になる

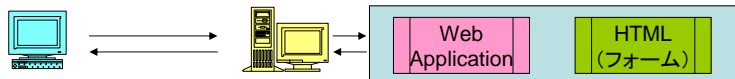


VCL for the Webサーバアプリケーション作成時の3つのモード

- スタンドアロンモード
 - 単体で動作するモード(WEBサーバ自体もVCL for the Webを利用)



- アプリケーションモード
 - 市販のWEBサーバを利用し、WEBアプリ部分の全てをVCL for the Webで作成



- ページモード
 - WebSnapを使ってWEBアプリを構築し、画面生成部分をVCL for the Webで作成



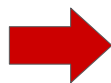
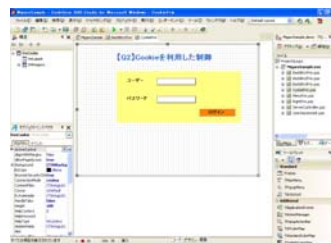
VCL for the Webアプリケーションの構成

- **ServerControllerオブジェクト**
 - VCL for the Webの管理を行うオブジェクト
 - WebApplicationオブジェクトとフォームオブジェクトの管理を行う
- **WebApplicationオブジェクト**
 - WEBブラウザ毎の情報を管理するオブジェクト
 - 複数ページにまたがるデータ(グローバル変数等)の保持を行う
- **Formオブジェクト**
 - ブラウザ画面からの入力を受け付けたり、画面を作成したりするオブジェクト
 - GUIアプリケーションと同様にフォームにプログラムを行う

VCL for the Webアプリケーションの開発手順

スタンドアロン/アプリケーションモード手順

- ① 新規プロジェクトの作成
- ② フォームにコンポーネントを貼り付けてプロパティの設定
- ③ 必要に応じてイベントハンドラを作成
- ④ コンパイル
- ⑤ 実行(テスト)

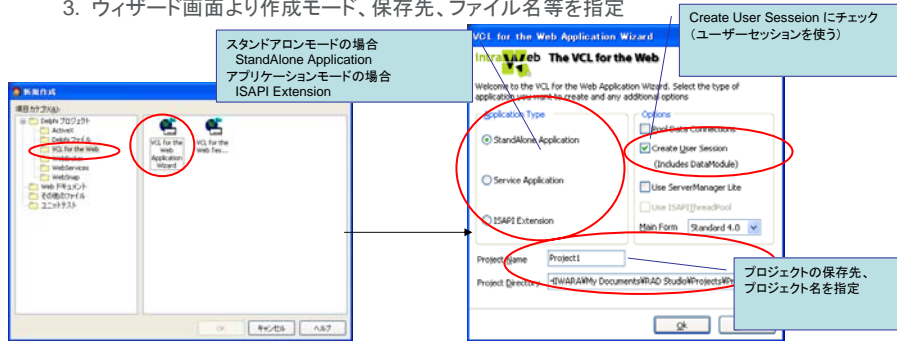


GUIアプリケーションと同じような手順で開発が可能

VCL for the Webプロジェクトの新規作成

VCL for the Webプロジェクトの作成手順

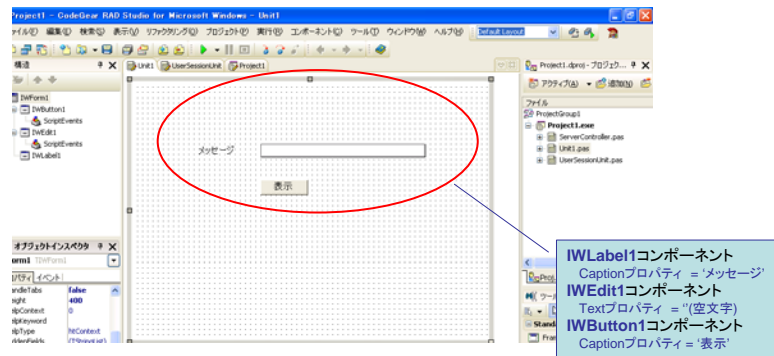
1. ファイル⇒新規作成⇒その他 を選択
2. Delphiプロジェクト
⇒VCL for Web より、[VCL for the Web Application Wizard] を選択
3. ウィザード画面より作成モード、保存先、ファイル名等を指定



コンポーネントの配置

コンポーネントの貼り付けおよびプロパティの設定

1. ウィザードにて作成されたプロジェクトよりUnit1を選択し、フォームを表示
2. [IWStandard]ページより、IWLabel, IWEdit, IWButton をフォームに貼り付け プロパティをセット



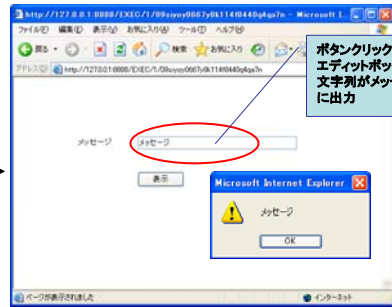
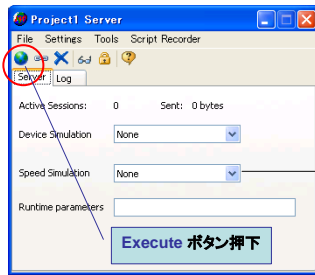
イベントの作成～コンパイル

イベントハンドラの設定とコンパイル/実行

1. IWButton1コンポーネントのonClickイベントハンドラを設定し、下記コードを入力

```
procedure TIWForm1.IWButton1Click(Sender: TObject);
begin
  WebApplication.ShowMessage(IWEdit1.Text);
end;
```

2. コンパイル ⇒ 実行 (起動するフォームからブラウザ起動ボタンを押下し動作確認)

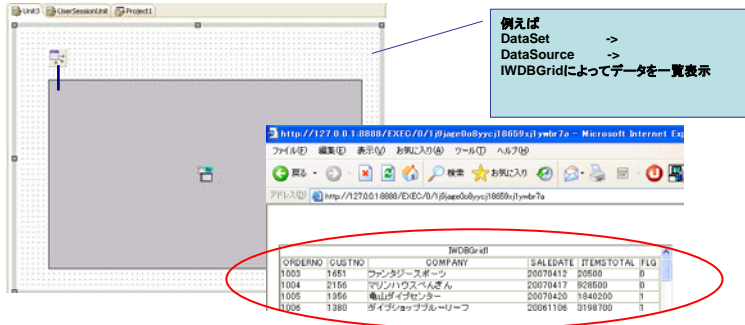


ボタンクリックにより、エディットボックスの入力文字列がメッセージボックスに出力

データベース接続

DB接続用コンポーネント

- データベース接続はDBExpress等、GUIと同様に使用することができます。データベースに連結した画面を作成したい場合[IW Data]ページにコンポーネントが用意されています。(Data Control)ページのコンポーネントに相当)



例えば
DataSet ->
DataSource ->
IWDBGridによってデータを一覧表示

画面遷移

画面の遷移方法

- Delphiプロジェクト⇒ VCL for the Web より、[New Form] を選択して追加
- 次画面呼び出し元の処理
 - **Createメソッド**でオブジェクトを生成
 - **Showメソッド**で遷移
- 次画面の処理
 - 画面終了時に**Releaseメソッド**呼び出しにより、元のフォームに戻る

```
procedure TfrmMain.btnGoFrm2Click(Sender: TObject);
var
    frmDetail: TfrmDetail;
begin
    frmDetail := TfrmDetail.Create(WebApplication);
    frmDetail.Show;
end;
```

フォーム変数はGlobal定義されていないので、ローカル変数として定義

```
procedure TfrmDetail.btnBackClick(Sender: TObject);
begin
    Release;
end;
```

グローバル変数

グローバル変数の扱い

- WEBアプリケーションの場合、**WEBブラウザ単位に変数を保持しなければならない。**
- VCL for the Webの場合、UserSessionユニットのIWUserSessionクラスに変数を定義し、**UserSessionオブジェクトとして利用**する。

```
TWUserSession = class(TIWUserSessionBase)
    sesSession: TSession;
    dbAS: TDatabase;
    qrySelectData: TQuery;
private
    { Private declarations }
public
    { Public declarations }
    LogOnUserID: String;
    LogOnUserName: String;
end;
```

グローバル変数
WEBブラウザ毎に
変数が別々に保持される

// ログオンユーザーID
// ログオンユーザー名

```
procedure TfrmMain.IWAppFormRender(Sender: TObject);
begin
    // ログオンユーザー名の表示
    lblLogOnUser.Caption := UserSession.LogOnUserName;
end;
```


【Q1】戻るボタンの対応方法

!【質問】

IE上の戻るボタンを制御する方法にはどのようなものがあるのでしょうか。

✉【回答】

バージョンにもよりますが、ブラウザの戻るボタンの動作はIWebServerControllerのOnBackButtonイベントで処理を行うことができます。
¥Demos¥IntraWeb¥Win32¥BackButtonにサンプルも用意されています。(バージョンによって異なります。)

【Q1】戻るボタンの対応方法

サンプル #Demos#IntraWeb#Win32#BackButton

```

procedure TIWServerController.IWServerControllerBaseBackButton(
  ASubmittedSequence, ACurrentSequence: Integer; AFormName: String;
  var VHandled, VExecute: Boolean);
const
  DefaultFormName = 'MainForm';
type
  TIWFormClass = class of TIWForm;
var
  LForm : TIWForm;
begin
  // This is a generic back-button handler that can easily be adapted to
  // your application
  VHandled := True;
  VExecute := True;

  if AFormName = "" then
    AFormName := DefaultFormName;
  end;

  if WebApplication.FindComponent(AFormName) <> nil then
    begin
      WebApplication.SetActiveForm(WebApplication.FindComponent(
        AFormName) as
        TIWContainer);
    end else begin
      try
        LForm := TIWFormClass(FindClass('T +
          AFormName')).Create(WebApplication);
        WebApplication.SetActiveForm(LForm);
      except
        VHandled := false;
      end;
    end;
  end;
end;

```

デフォルトにするフォーム

フォーム名が取得できない場合はデフォルトのフォームを指定

指定されたフォーム名をWebApplicationの中で探して存在すればアクティブにする。

指定されたフォーム名をWebApplicationの中で探して存在しなければ生成してアクティブにする。

【Q2】Cookieを利用した制御

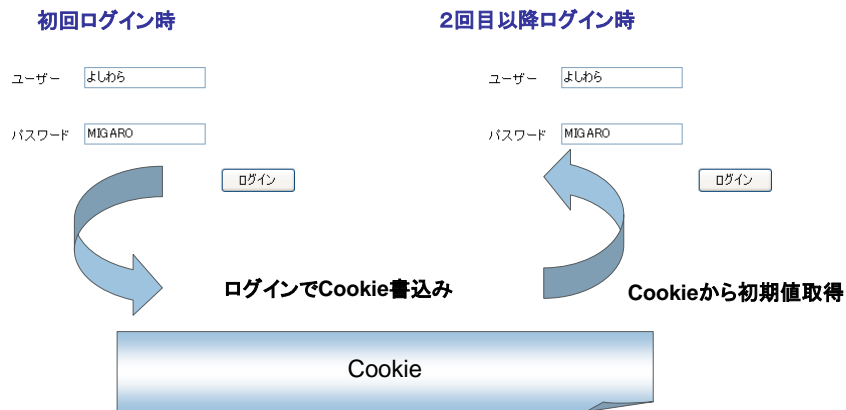
【質問】

Cookieを利用して当日限りで1回目以降のログイン入口に初期値を設定することはできますか？

【回答】

CookieはTWebRequesのCookieFieldsプロパティやTWebResponseのSetCookieFieldメソッドで扱うことができます。
扱えることとは別にセキュリティ的な配慮は必要です。

【Q2】Cookieを利用した制御



【Q2】Cookieを利用した制御

ログイン時にCookieの作成

```

procedure TIWForm1.IWButton1Click(Sender: TObject);
var
  slstCookie: TStringList;
begin
  slstCookie := TStringList.Create;
  try
    with slstCookie do
      begin
        Clear;
        Append('USER=' + IWEdit1.Text);
        Append('PASS=' + IWEdit2.Text);
      end;
      webapplication.Response.SetCookieField(slstCookie, ", ", (Now + 1), False);
    finally
      slstCookie.Free;
    end;
  end;
end;

```




SetCookieFieldメソッドについては
次ページ参照

【Q2】Cookieを利用した制御

SetCookieFieldメソッドとCookieの扱い

```
procedure SetCookieField(Values: TStrings; const ADomain: string; const APath: string; AExpires:
  TDateTime; ASecure: Boolean);
```

 http://Server1/DelphiISAPI/MigaroSampleISAPI.dll/EXEC/1/0ngtrd518zs90w14ga8i61iv9rv9

【パラメータ】

Values: Cookieの格納内容

複数のCookieをまとめて書き込むことができます。

ADomain: Cookie送信先のドメイン名

省略時は使用している場合、Cookie応答を生成したサーバのホスト名です。

APath: Cookie送信先のパス

URLのドメイン以降のパス。

取得時にはこのパスが前方一致で一致するCookieを扱います。

実際にアクセスを行うURLの考慮が必要です。

Aexpires: Cookie有効期限

削除する場合は過去の日付を設定したりもします。

Asecure: セキュリティの確保

セキュリティが確保された接続の使用時にCookieをクライアントによってのみ渡すかどうかを設定します。

例えばこのURLであれば動的生成部分を含めないように配置先を指定する。
'/DelphiISAPI'と指定しておくと同前一致するURL上で格納したCookieを扱うことができます。

【Q2】Cookieを利用した制御

Cookieに格納された値

ユーザー

パスワード

Cookie上の値
USER%82%E6%82%B5%82%ED%82%E7
PASSMIGARO

そのまま取得すると...

ユーザー

パスワード

空白や特殊記号、日本語等の全角文字はHTTPAppユニットのHTTPDecodeでデコードして取得する必要があります。

画面起動時にCookieの取得

```
procedure TIWForm1. IWAppFormCreate(Sender:
  TObject);
begin
  with webapplication.Request.CookieFields do
  begin
    IWEdit1.Text := HttpDecode(Values["USER"]);
    IWEdit2.Text := HttpDecode(Values["PASS"]);
  end;
  デコードして値を取得。
end;
```

Cookieから初期値取得

ユーザー

パスワード

【Q3】IWTextの右寄せ表示

【質問】

IWTextでの表示内容を右寄せで表示したいのですが、可能でしょうか？

【回答】

IWText、IWMemo、IWButton等には残念ながらAlignmentプロパティは存在しません。右寄せに表示を行いたい場合はstyleタグを利用する方法が考えられます。

【Q3】IWTextの右寄せ表示

ExtraTagParamsプロパティにStyleタグを記述

The screenshot shows the Delphi IDE with three components: IWText, IWMemo, and IWButton. The 'Object Inspector' on the left shows the 'ExtraTagParams' property for IWText1, which is set to 'style=text-align:right;'. A callout box titled '文字列リストの記述' (String List Description) shows the same text: '1行' followed by 'style=text-align:right;'. A speech bubble notes: 'Delphi7で付属していたIWEditもAlignmentプロパティがないので同じような対処ができます。' (Delphi 7's IWEdit also doesn't have an Alignment property, so the same workaround can be used.)

【Q4】開発モードの変更方法

!【質問】

スタンドアロンで開発してISAPIアプリケーションに変更することはできますか？

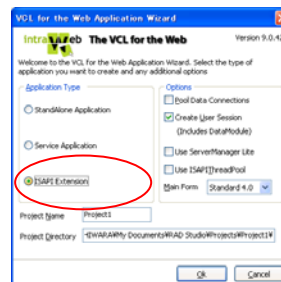
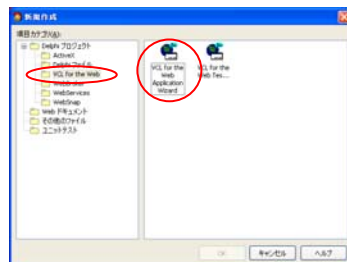
✉【回答】

ISAPIアプリケーションで新規作成したプロジェクトにスタンドアロンで作成モジュールを取り込みを行うことで簡単に作り変えることができます。
テスト/本番用にプロジェクトファイルを用意すると便利です。

【Q4】開発モードの変更方法

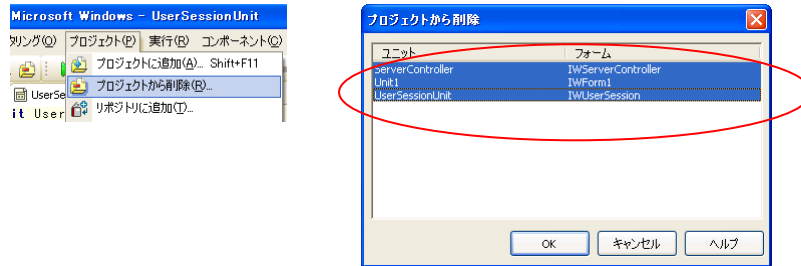
ISAPIへの変更手順

- ①メニューの[ファイル | 新規作成 | その他]を選び、表示されたツリーの中から「Delphi プロジェクト-VCL for the Web」を選択します。
次に表示されたアイコンの中から「VCL for the Web Application Wizard」を選択します。
「Application Type」で「ISAPI Extension」を選択します。



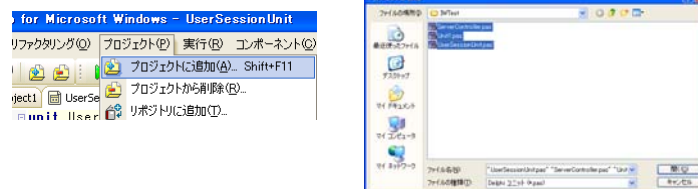
【Q4】開発モードの変更方法

- ② IWForm1、UserSessionUnit、ServerControllerのファイルが作成されます。
デフォルトで作成されたこれらのファイルが不要ならば、メニューの「プロジェクト|プロジェクトから削除」で削除します。



【Q4】開発モードの変更方法

- ③メニューの「プロジェクト|プロジェクトに追加」を選択し、スタンドアロンモードのユニットのあるフォルダを指定し、必要なユニットを追加します。



- ④メニューの「ファイル | すべて保存」を選びます。
- ⑤メニューの「プロジェクト|XXXXX(プロジェクト名)をコンパイル」を行うと、ISAPIのdllファイルが作成されます。

【Q5】IIS上での動作の違い

!【質問】

スタンドアロンモードで動作した処理やタイムアウトの動作が、IIS上ではうまくいきません。

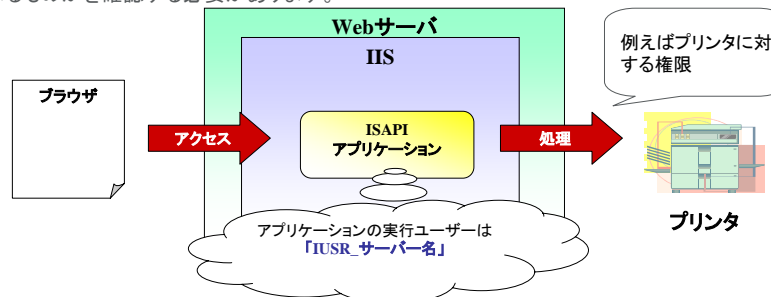
✉【回答】

IIS上で動作を行う場合、スタンドアロンモードと違い、IISのユーザーでプログラムが実行されることとなります。環境上の権限やIISの制御の設定などの実行環境との違いを確認する必要があります。

【Q5】IIS上での動作の違い

実行ユーザーの権限

IISが稼動しているOSはデフォルトでは「IUSR_サーバー名」アカウントのユーザーとして、匿名アクセスを認識しています。ISAPIアプリケーションもこのユーザーで実行されることとなります。この「IUSR_サーバー名」アカウントはデフォルトではローカルログオン権限とゲスト権限しか割り当てられていません。スタンドアロンモードとの動作の違いをISAPIの実行ユーザーの権限によるものかを確認する必要があります。

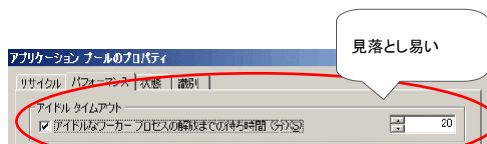


【Q5】IIS上での動作の違い

セッションのタイムアウトについて

【アプリケーション側】

IWServerControllerのSessionTimeoutプロパティで設定することができます。またSessionTimeoutURLプロパティでのURLやファイルを設定しておくことで、セッションアウト時の遷移先とすることもできます。(URLとファイルは片方しか設定できません)



【IIS側】

バージョンによって設定も異なりますが、通常の「接続のタイムアウト」の設定の他、アプリケーションプール上のアイドルタイムアウトなどの設定も管理する必要があります。IIS側でセッションが切断されてしまうと、アプリケーション側もセッションを保つことはできません。

【QA】GUIアプリからWEBサイトの呼出

!【質問】

GUIアプリケーション内で検索Webサイトを呼び出して結果を表示するようなことはできますか？

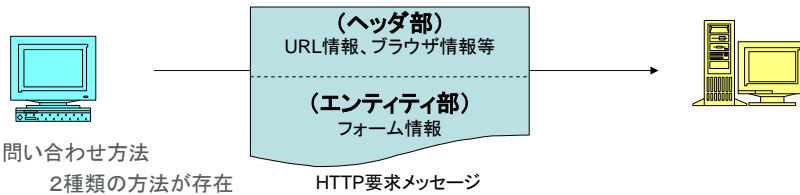
✉【回答】

検索パラメータなどをURLエンコードで変換して、サイトにあわせた指定を行い、呼出を行うことができます。WebBrowserコンポーネントで結果をアプリケーション画面内に取り込むことも可能です。

【QA】GUIアプリからWEBサイトの呼出

WEBサーバーへの問い合わせ

WEBブラウザからWEBサーバーへの問い合わせ
HTTP要求メッセージをWEBブラウザからWEBサーバーへ送る



問い合わせ方法
 2種類の方法が存在
 • GETメソッド
 • POSTメソッド

【QA】GUIアプリからWEBサイトの呼出

GETメソッドとPOSTメソッド

それぞれの特徴

	GETメソッド	POSTメソッド
パラメータセット方法	URL内のQuery部	エンティティ部
WEBブラウザの表示	アドレス部にQuery部が付加されて送信	表示は不変
パラメータの大きさ	大きなデータ送信は不向き	大きなデータを送信可能

POSTメソッドは、フォームデータ内部にデータを持つ為、URLでの指定は不可能だが、GETメソッドの場合、URL自体にパラメータが記述できるため、比較的容易に問い合わせが可能

⇒ GETメソッドの場合、URLにパラメータを指定した問い合わせが可能

【QA】GUIアプリからWEBサイトの呼出

GETメソッドによる問い合わせ例(1)

日本郵便 郵便番号検索 (GETメソッド)

`http://search.post.japanpost.jp/cgi-zip/zipcode.php?zip=5560017`

プロトコル

ホスト名

スクリプト名

問い合わせ
QUERY



【QA】GUIアプリからWEBサイトの呼出

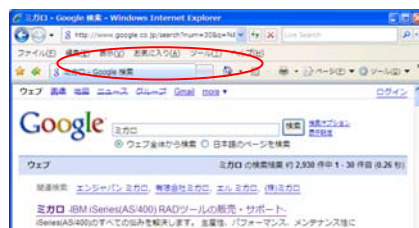
GETメソッドによる問い合わせ例(2)

`http://www.google.co.jp/search?num=30&q=%E3%83%9F%E3%82%AC%E3%83%AD`

google検索 (GETメソッド)

問い合わせQUERY

結果件数=30件 & 検索語=ミガロ



【QA】GUIアプリからWEBサイトの呼出

URLエンコード

ブラウザから送信するデータは、URLエンコードが必要
URLエンコード ... 空白や特殊記号、日本語等の全角文字を
符号化するルール

使用する文字コード体系により値が決定

例) 「ミガロ」

- Shift-JIS 「%83%7E%83K%83%8D」
- UTF-8 「%E3%83%9F%E3%82%AC%E3%83%AD」

Delphiでは、HTTPAppユニットにあるHTTPEncode関数で変換可能

- Shift-JIS の場合 HTTPEncode(Avalue)
- UTF-8 の場合 HTTPEncode(AnsiToUtf8(Avalue))

【QA】GUIアプリからWEBサイトの呼出

GoogleMapの呼出例



Delphiで入力した住所をURLエンコードし、URLを送信

応答されたHTMLをWebBrowserコンポーネント上に表示

```
procedure TFormMain.button1Click(Sender: TObject);
var
  sText: String;
begin
  sText :=
    Format('http://www.google.co.jp/maps?hl=ja&tab=wl&q=%s',
      [HTTPEncode(AnsiToUtf8(editAddress.Text))]);
  WebBrowser1.Navigate(sText);
end;
```

【QB】デバッグ実行の判断

!【質問】

開発環境でテストで実行する際のDB接続のログインを省くことはできませんか？

✉【回答】

実際のアプリケーションの実行とデバッグ実行では System.DebugHook 変数の値が異なります。 System.DebugHook 変数で判断して、ログインの省略を用意しておけば、デバッグ実行の省略ができます。

【QB】デバッグ実行の判断

SQLConnectionのDebugHook制御例

```
with SQLConnection1 do
begin
//デバッグ実行であればユーザーパスワードを自動設定
if DebugHook <> 0 then
begin
LoginPrompt := False;
Params.Values['User_Name'] := 'ユーザー名';
Params.Values['Password'] := 'パスワード';
end
else
begin
LoginPrompt := True;
Params.Values['User_Name'] := '';
Params.Values['Password'] := '';
end;

Connected := True;
end;
```

DebugHook
デバッグ実行時:1
EXE実行時:0

開発環境デバッグ実行時だけ暗黙のログインを行うことで、開発上での手間を省きます。その他デバッグ実行時のみ扱うデータを切り替えるなどにも便利です。