尾崎 浩司

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:VCL for the Web活用TIPS紹介

Web アプリケーション開発に役立つ 4 つの TIPS。「VCL for the Web」を用いることで GUI 開発と同様、生産性の高い開発が実現できる。

 VCL for the Webとは
 Ajaxを使用したアプリケーション構築方法
 動的に生成したファイルのダウンロード
 コネクションプーリングを使用した Delphi/400アプリ開発
 スマートフォン対応ページの作成方法
 まとめ



1973 年 08 月 16 日生 1996 年三重大学工学部卒 1999 年 10 月株式会社ミガロ. 入社 1999 年 10 月システム事業部配属

現在の仕事内容 ミガロ . 入社以来、主に Delphi/400 を利用した受託開発を担当している。

1.VCL for the Webとは

「VCL for the Web (旧 IntraWeb)」 とは、Delphi/400 に付属する Web ア プリケーション開発ツールのことであ る。

この VCL for the Web は、通常の VCL フォームアプリケーションと同様 にフォーム上へコンポーネントを貼り付 け、必要に応じプロパティやイベントを 設定することで開発を行えることが特徴 である。

ミガロ.では、Web アプリケーション 開発において積極的にこの VCL for the Web を使用しており、Delphi/400 にお ける Web アプリケーション開発のノウ ハウも蓄積されつつある。

本稿では、これらノウハウの中から役 に立つであろう4つの TIPS について、 具体例を交えながら紹介しようと思う。

※ VCL for the Web をはじめとする Delphi/400 における Web アプリケー ションの基本的な開発手順については、 ミガロ.のホームページに詳しく紹介し ているのでそちらを参照していただきた い。

【Delphi/400 の技術情報(インターネット)に関する Tips】

http://www.migaro.co.jp/contents/ products/delphi400/tips/web/index. html

2.Ajaxを使用した アプリケーション 構築方法

VCL for the Webでは「Ajax」を使 用することが可能である。Ajax とは 「Asynchronous JavaScript + XML」 の略で、Webページのリロードを伴わ ずに、Webサーバーと XML 形式のデー タのやり取りを行って処理を進めていく 対話型 Web アプリケーションのことで ある。

従来の Web アプリケーションは、図 la のように入力後ボタン等を押下する ことで、サブミットを行い、新しいペー ジを要求するという構成であった。【図 1a】

Ajax を使用すると、図 1b のように ページを切り替えることなく、入力中に どんどん最新情報に更新していくことが 可能になるのである。【図 1b】

この仕組みを VCL for the Web では 使用することができる。しかも、JavaScript を使用せず、Delphi 言語をそのまま使 用することが可能であるため、とても魅 力的である。

では、VCL for the Web での Ajax 使 用方法を見てみよう。

図2は、入力用コンポーネントである IWEditのイベント一覧である。【図2】

この中にある「OnAsync 〜」という イベントが、Ajax に関連するものであ る。通常のコンポーネントと同様のイベ ントが用意されていることがわかるであ ろう。つまり、通常のイベントと同じ感 覚で処理を記述することができるのであ る。

具体例として、図3のような仕様を検



討する。【図3】

顧客コード欄(edtCustCd)にフォー カスインしたタイミングとフォーカスア ウトしたタイミングとでイベント処理を 行い、画面上の値が変更された際に顧客 マスターより値を取得すればよいのであ る。【ソース1】

このプログラムを実行すると、図4の ようになる。【図4】

このように Ajax を使用すると、都度 画面をリロードしなくても対話できるア プリケーションが開発できるので、とて も便利なのである。

3.動的に生成した ファイルのダウンロード

Web アプリケーションを作成している と、ファイルをダウンロードさせたい場 面があるだろう。

この際、あらかじめ特定のフォルダに 用意されたファイルをダウンロードする だけならば、WebApplication 変数にある SendFile メソッドを実行すればよいだけ である。

しかしながら、データベースから抽出 した結果等をダウンロードさせるとなる と、動的にファイルを生成しなければな らない。Web アプリケーションはWeb サーバー上で実行されるため、クライア ント側の要求ごとにWeb サーバー側に動 的にファイルを生成するというのはファ イル名の重複等の問題もあり、難しいの ではないだろうか。

では、どうすれば抽出結果から、ファ イルをクライアント側にダウンロードさ せることができるのだろうか。このよう な場合、サーバーではファイルではなく ストリームとして作成し、クライアント ヘデータ転送を行えばよいのである。

具体例を検討してみよう。図5を見て いただきたい。【図5】

これは、データベースから取得した結 果を、IWGrid に表示している例である。 この画面にある CSV 出力リンクをクリッ クすると、画面表示されているグリッド 内容をクライアントに CSV ファイルとし て転送させるのである。

ソースコードを紹介しよう。【ソー ス2】 このプログラムでは、CSV データを 作成するために文字列リスト (TStringList)を使用している。この文 字列リストには、ストリームオブジェク トに書き込む SaveToStream メソッド があるので、メモリーストリームに保存 できる。そうすることで、CSV データ を直接ストリーム化することができるの である。

さらに、作成したストリームは WebApplication 変数の SendStream メ ソッドを使用すれば、クライアントにダ ウンロードさせることができる。

この方式を使用すれば、Web サーバー 側にファイルを生成することが一時的に も生じることなく、クライアント側にダ ウンロードさせることが可能になる。

Web アプリケーションでは、CSV 形 式のダウンロードが要求される場面が多 いと思うので、本ロジックを開発時の参 考にしていただきたい。

なお、このプログラムをそのまま実行 してしまうと、VCL for the Webの画 面ロックの制約により、ダウンロード後、 画面上の動作が一切受け付けられなく なってしまう。これを解消するには、ダ ウンロード対象となる IWLink コン ポーネントの OnScriptEvents イベント に、設定が必要となるので注意してほし い。【図 6】

4.コネクションプーリング を使用したDelphi/ 400アプリ開発

「コネクションプーリング」とは、デー タベースへアクセスの都度、接続(コネ クション)を確立するのではなく、あら かじめ事前に一定数のコネクションを確 立しておき、それを使い回す手法のこと である。データベースアクセスの負荷を 減らすために用いられる。

コネクションプーリングは Web アプ リケーション開発でよく使用する手法だ が、これを VCL for the Web でも使用 することが可能である。

VCL for the Web で、コネクション プーリングを使用する手順を紹介しよ う。

新規作成にて「VCL for the Web Application Wizard」を選択すると、 ウィザードが起動する。この画面で、 [Options] → [Pool Data Connections] のチェックを ON にすればよい。【図 7】 このオプションを設定してプロジェク トを新規作成すると、図 8 のようになる。 【図 8】

ServerController.pas に、Pool と 命 名された IWDataModulePool コンポー ネントが設定されているのがわかる。こ の Pool コンポーネントには、PoolCount というプロパティが用意されており、こ こに指定した数の接続が維持されるので ある。(Delphi/400 の場合、PoolCount に指定した数だけジョブが生成されると 考えればよい。)

では、このコネクションプーリングが 使用できる環境下で、実際にデータベー スにアクセスするにはどうすればよいの だろうか?

データベース接続(コネクション)を 複数の Web セッションで共有して使用 するため、使用開始する際に明示的にコ ネクションをロックし、使用が終了した らロックを解除する必要がある。

実際の使用例をソース3に示すので参 考にしてほしい。【ソース3】

コネクションプーリング制御下でデー タベースにアクセスする場合、アクセス 対象とするデータモジュールをロックす るために、変数を定義する。(ソース3 では、ADataModule という変数を宣 言。)

そして、データベースにアクセスする 直前で LockDataModule 関数を実行す ると、その時点で空きのあるコネクショ ンを探し、ロックを行う。

使用が終了したら、UnlockDataModule 手続きでアンロックする、という流れで ある。(これらの関数/手続きは、 ServerControler.pasに処理が記述され ているため、使用する際には、データモ ジュールユニットと共にユニット参照を 行うこと。)

このようなロジックを追加すること で、Web アプリケーションに対し、コ ネクションプーリング制御ができるので ぜひ検討してみてほしい。

なお、CO400Connection ドライバ (dbExpress 接続) でコネクション プーリングを行う際、データモジュール 生成時 (OnCreate) に、あらかじめ TSQLConnection の接続を行うと正常 に動作しない。初回のデータベースアク

the second se			
朝 ホームペーク ⑤ ProSan D	phot gruns	0	
委注目:		#36	
WAT-N:	- 30000000	concustoname: invitaten	
2.6	1. 11 19 10 10		
用オプジェクトインスパ	29 (F)		
eRCustCD TIVES		F 60.0×41.975	
70/04 1/07		L MOTP LICEN 副答コード 欄(edhOustCD)に値を入力後、フォー	
OnAsyncChange OnAsyncOldk		カスが扱ける物に、顧客マスターを使用する。	
OnAsyncDoubleClick OnAsyncEnter	edicustCDAsynctriter	マスターに通が存在すれば、名称欄に翻客名を表示	
OnlayndDit OnlayndXeyDown	edtCustCDAsyncbat	マスターに進が存在しない場合、エラーメッセージを	
OnAsyndkeyPress OnAsyndkeyLb		PGT-	
OnlayndNuseDown OnAsyndNuseNove			
Onlicynd/kuseOut Onlicynd/kuseOver			
OnAsyndMisuseLip OnMTMLTag			
Ordubmk			
4			
Chilles // localbast/hite2/	No/Sample DIL 48/1310/0/0/ w	nationite that Table 11 inter Table 1	
OQ- Duplante	1-1-1 × 0 + 1-1 × 1-1-1	http://ies.allocit/ies//Perglample DII.dll/CRE070702 reg56x1522	
7HAD 8880 RED	104127040 9-AU AN	Provide and the set of th	
#110//bahat/de2/Pog.	5 · D · D · D · D	2-16-10 編集型 第3-10 2012/01/0 9-16 0 AU209 2-2012/0 (山)	
		語·mail/head 顧客マスターに存在するコードを 3 楽 · ハージジ・ 入せんたける 調査系があっ	
218		CONTRACT OF THE CONTRACT	
B B B C C C		eng.	
44: 10	1	2388:	
		2188: 	
L6:	And Area Bangle DLAW/XII	10 AUTOR AND	
26: () http://acaba () () () () () () () () () () () () () (And Ann Density Contract of the Annual State	またまた。 またまたにおからしたからしてなられたにはのた。 ■本生またのの ■本生またの =本textextextextextextextextextextextextext	
26: (110///seats (20) - (2) 	Auto Anno Annota Coll All All Tanahard 1007 日日 (1) 10日日 1007 日日 (1) 10日日 1007 日日 (1) 10日日 1007 日日 (1) 10日	支注目: ■ 100000006 ■式会社1ガロ。 100000006 ■式会社1ガロ。 100000006 ■式会社1ガロ。 100000006 ■式会社1ガロ。 100000006 ■式会社1ガロ。 100000006 ■式会社1ガロ。 100000006 ■式会社1ガロ。	
L6: Child Marchine Constraints Series Baco Series Sing Bill Series Series Sing Series Sing Series Sing Series Sing Series Sing Series Sing Series Sing	Ant-24700 (Sample El al ATEX Tacabati 1027 日 日 11711 1日日 2015-2018 Art 1127(1・日・ 入力によ 支払 2015-1129(1・日・ 支払	型油目: ■ 0 10000006 ■式会社2月0. 100/0/0/methallocy=12geble1100c ■ ■ 文 × □ 1 me P 1 × □ 1 me P 1 × □ 1 me P 1 (>	
L6: Chitp://incals Orio/D Else Chitp://incals Chitp	And Anno provide Contract Transford (Contract Annual Contract (Contract Annual Contract Annual	ままます。 またまます。 またまます。 たたますたい たまます。 たますたい たますたい たますたい たますたい たますたい たますたい たますたい たますたい たますたい たますたい たますたい たますたい たますたい たますたい たますたい たますたい たますたい たますたい たままたい たまたい たままたい たまたい	
2.6: () http://www. () () () () () () () () () () () () () (Auto 2 (Per) Sample CL 400 (21) The solution 2 (P ● 19) 100 2 (Per) 2 (Per) 2 (Per) 2 (Per) 2 (Per) 2 (Per) 2 (Per	ままま: 100/0000006 ■式会社1ガロ。 100/0000006 ■式会社10000006 ■式会社100000000 100/0000006 ■式会社100000000 100/0000000000000000000000000	
26: (http://acabi (0) (0) (1) (1) (1) (1) (1) (1) (1) (1)	Anno 2010 (Constant Constant) 「Readword 10:07 (日日) (日日) 日日日 10:07 (日日) (日日) 日日日 10:07 (日日) (日日) 日日日 10:07 (日日) 日日日 10:07 (日日) 日日日 10:07 (日日) 日日日 10:07 (日日) 日日日 10:07 (日日) 日日 10:07 (日日 10:07 (日日) 日日 10:07 (日日) 日日 10:07 (日日 10:07 (日日 10:07 (日日) 日日 10:07 (日日 10:07 (日日 10:07 (日) 日日 10:07 (日日 10:07 (日日 10:07 (10:07 (10:07 (10:07	また また また また また また また また また また	
26: (110///acaba 27:042 28:0 27:042 28:0 27:042 28:0 27:042 28: 27:042 28: 28: 28: 28: 28: 28: 28: 28:	Auto Alter processon Control (1997) Texaduct 10077 第一巻 (中) 3 1000 2070;22(19) 第877 2017(19)(1+2)(1+3)(14) 第第一 1017(19)(1+2)(1+3)(14) 1017(19)(1+3)(1+3)(14) 1017(19)(1+3)(1+3)(1+3)(1+3)(1+3)(1+3)(1+3)(1+3	型注目: ■ 100000006 ■式会社1灯口。 100000006 ■式会社1灯口。 × 回 === × 回 === × 回 === × 回 === (> →== (>+>==) →== (>+===) →== (>+====) →== (>+====) →== (>+====) →== (>+====) →== (>+=====) →== (>+=====) →== (>+======) →== (>+=========) →== (>+==================================	
2.6: () 1100//1004 () 2100//1004 () 2100//1004000 () 2100//1004000 () 2100//1004000 () 2100//1004000 () 2100//1004000 () 2100//1004000 () 2100//1004000 () 2100//100400 () 2100//10000 () 2100//100000 () 2100//100000000 () 2100//100000000000000000000000000000000	Anno 2010 (Constant Constant) 「Standard Society (Constant) 「Standard Society (Constant) 「日本日本のです。」「「「「「」」」 「「「」」」」 「「」」」」」 「「」」」」」」 「「」」」」」 「」」」」 「」」」」 「」」」」 「」」」」 「」」 「」」」 「」」 「」」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」」 「」」」 「」」 「」」 「」」」 「」」」 「」」 「」」 「」」 「」」」 「」」 「」」」 「」」 「」」 「」」 「」」 「」」 「」」」 「」」 「」」 「」」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」」 「」」 「」」」 「」」 「」」」 「」」 「」」 「」」 「」」 「」」 「」」」 「」」」 「」」 「」」」 「」」」 「」」 「」」 「」」 「」」 「」」」 「」」」 「」」」 「」」」 「」」 「」」」 「」」 「」」 「」」 「」」」 「」」」 「」」 「」」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 」 」」 「」」」 「」」」」 「」」」 「」」」 」」」 「」」」 「」」」 「」」」」 「」」」 「」」」 「」」」」 」」	支注目: 100000006 ●式会社1月口。 100000006 ●式会社1月口。 1000000006 ●式会社1月日 1000000006 ●式会社1月日 1000000000000000000000000000000000000	
26: (110///weaks 27:04/2 28:0 27:04/2 28:0 28:	And Alter paragraph (1) 411/111 「Andread Inter 7 第 日 何) 1000月10日 (1077 第 日 何) 1000月11日 (1077 前 日 句) 1000月11日 (1077 前	ままたくありません。 ままたくありません。 ままたくありません。 ままたくありません。	
26: () http://iecate () - () - () - () - () - ()	Anno 2 A Trans (Conservation Conservation	2188: 10000000 Brite Br	
	Alex 2 / Yes (Senate Cli di Vili II) Tendent (10 77 日日) El (1)	2:11: 1000000000 #:Kalkifu. 1000000000 #:Kalkifu. Image: State of the first state of th	
2011 2011	And A free provides (1) 400 (101) The advect (101) 7 (10) The advect (101) 7 (101)	2:18: 100000000 #Kdk1ff0. 100000000 #Kdk1ff0. 1000000000 10000000000 10000000000 10000000000 100000000000 10000000000000000 1000000000000000000000000000000000000	
	Anno 2010 (1990)		
26: (1) 110///2018 (2) 010//2018 (2) 110//2018 (2) 110//2018	Anno 1 Mar 1 Stangele Cit (11) (11) The advect for (7) (2) (4) (4) (11) The advect for (7) (2) (4) (4) The advect for (7) (2) (4) The advect for (7) The advect for (7		
201 201 201 201 201 201 201 201	Auto 2 / You piceuple (11 40/ 1201 Troubled 10:07 第 日 1 (1) Troubled 10:07 1 (1)		
	Ansolventie Ansolventie Ansolventie	2010 Bit State 100 100 Bit State 100	
	Anno 2 Mar (Sengele Cit di M 1911 Terradiate 1 60 77 第 日 (1) Terradiate 1 60 77 8 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2018 00000000 Bridskinder 1000000000 Bridskinder Image: State of the state	
20 20 20 20 20 20 20 20 20 20	Auto 2 (1997) (2009) (11 - 0.01 (1997) The solution (10-07 (1)) (2) (1) (1) (1) (1) (1) (1) (1) (1) (1) (1	2018: 00000000 6-K detator. 10000000000 6-K detator. 1000000000000 6-K detator. 1000000000000 6-K detator. 1000000000000000000000000000000000000	
26: 24:00///2008.00 27:00/2008.00 26:00/2008.00 27:00/2008.00 26:00/2008.00 27:00/2008.00 26:00/2008.00 27:00/2008.00 26:00/2008.00 27:00/2008.00 26:00/2008.00 27:00/2008.00 26:00 27:00/2008.00 26:00 20:00/2008.	Anno Alexandre Col Alexandre Anno Alexandre Col Alexandre Anno Alexandre Col Alexandre Anno Anno Alexandre Anno A	2018: 0000000 Brdattan. 10000000 Brdattan. 10000000 Brdattan. 10000000 Brdattan. 10000000 Brdattan. 10000000 Brdattan. 100000000 Brdattan. 100000000 Brdattan. 10000000000 Brdattan. 1000000000 Brdattan. 100000000000 Brdattan. 1000000000000000000000000000000000000	
	And Schweisen (1) (1) (1) The sector of the Property (1) (1) (1) (1) The sector of the Property (1) (1) (1) The sector of the Property (1) (1) The sector of the Property (1) The sector of the Propert	2018: 000000000000000000000000000000000000	
Image: Second	Auto 2 (1997) (2009) (11 - 0.01 (11 - 11 - 11 - 11 - 11 - 11 - 11 - 1	2018: 100000000 B-K 2012 ATD. 200000000 B-K 2012 ATD. 2000000000 B-K 2012 BTD. 2000000000 B-K 2012 BTD. 2000000000 B-K 2012 BTD. 20000000000 B-K 2012 BTD. 20000000000 B-K 2012 BTD. 20000000000 B-K 2012 BTD. 20000000000 B-K 2010 BTD. 20000000000 B-K 2010 BTD. 200000000000 B-K 2010 BTD. 200000000000 B-K 2010 BTD. 2000000000000000000000000000000000000	
26: 24:00///2008.00 27:00/2008.00 27:00/2008.00 27:00/2008.00 27:00/2008.00 27:00/2008.00 27:00/2008.00 27:00/2008.00 27:00/2008.00 27:00/2008.00 27:00/2008.00 27:00/2008.00 28:00 27:00/2008.00 28:00 27:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.00 28:00 20:00/2008.	Anno 2019 (1999)	2018: RX 1000000000000000000000000000000000000	
Image: Second	Auto 2 (Mar (Seeger C) 2 (M / 1 M) The advect (1 + 2 / 1) The advect (218: 10000000 B-K dt H J HD 100000000 B-K dt H J HD 1000000000 B-K dt H J HD 10000000000 B-K dt H J HD 10000000000 B-K dt H J HD 100000000000 B-K dt H J HD 1000000000000000000000000000000000000	
Image: Second	Auto 2010 (1) Sengle (1) 401/101 Totalised (1) 27 (2) (2) (4) (2) (1) Totalised (1) 27 (2) (2) (4) Totalised (1) 27 (2) (2) (4) Totalised (1) 27 (2) (2) (4) Totalised (1) 20 (2) (2) (2) Totalised (1) 20 (2) (2) (2) Totalised (1) 20 (2) (2) (2) Totalised (1) (2) (2) (2) (2) Totalised (1) (2) (2) (2) (2) Totalised (2) (2) (2) (2) (2) Totalised (2) (2) (2) (2) (2) (2) Totalised (2) (2) (2) (2) (2) (2) Totalised (2) (2) (2) (2) (2) (2) (2) Totalised (2) (2) (2) (2) (2) (2) (2) (2) Totalised (2) (2) (2) (2) (2)	218: 0000000 # A BE A BA 10000000 # A BE A BA 10000000 # A BE A BA 100000000 # A BE A BA 10000000 # A BE A BA 100000000 # A BE A BA 10000000 # A BE A BA 1000000000 # A BE A BA 100000000 # A BA 10000000000 # A BA 100000000 # A BA 1000000000000000000000000000000000000	

セス時(初めて TSQLQuery 等がオー プンする時)に、接続されるようにする のがポイントとなるので注意いただきた い。

5.スマートフォン対応 ページの作成方法

近年、スマートフォンがブームとなっ ている。今後業務アプリケーションにも スマートフォン向けサイトの対応を迫ら れることが予想される。

スマートフォンは標準でブラウザ機能 を持っているので、これまでの Web ア プリケーションをそのまま動作させるこ ともできるが、PC と違い、スマートフォ ンでは解像度の制約がある。そのため、 通常のサイトをそのまま一覧表示する と、図9のようにとても細かな表示と なってしまう。【図9】

解決策の1つとして、スマートフォン に最適な画面設計を行えばよいわけだ が、スマートフォンは機種により解像度 が異なるなど、そのままではなかなか調 整が難しいものがある。

この場合役に立つのが、HTML のメ タタグの1つである「ViewPort」である。 このメタタグは、スマートフォンでの可 視領域やズームなどの設定するものであ る。記述例を図 10 に示す。【図 10】

では、このメタタグを VCL for the Web のフォームにどのように組み込め ばよいのだろうか。それは、IWForm にある ExtraHeader プロパティに記述 すればよいのである。【図 11】

そして、メタタグを設定したフォーム を実行すると、図 12 のようにスマート フォンに最適化され、不要な拡大縮小を 伴わない画面が構築できるのである。 【図 12】

実際の画面設計では項目数を制限する など、スマートフォンならではの制約も 出てくるであろうが、メタタグの可視領 域を設定することで、よりスマートフォ ンに最適化された画面が構築できる。

6.まとめ

今回は、VCL for the Web にまつわる 4つの TIPS を紹介した。Web アプリ ケーション開発の一助としてほしい。 また、これまで Web アプリケーショ ンを作成したことがない方も、一度 VCL for the Webによる開発を実践い ただき、GUI 開発と同様に生産性の高 い開発が可能であることを確認いただけ ると幸いである。

Μ

3	
用 おびょうトインス (5)5	
Inflakt Uktor	
Name DVLRki Remet O doubby R True	VGL for the web Event Scripts filter
ForetSortik [false Favfot [false	ondort X Entranscol(); onder Globusticological
P Southerts TWSouth	vents)
Sec.	or Cragenge
il SyletanderCottons (TTWRylet)	en pt offens
Tag 0 100 40	offer(D)
Visible Proe Width 65	orthoad orthoa
22400 (0 behalleds (47) for the Web) 13	B Minut M
7	
VCL for the Web Applicat	ion Witard
intrated The VCL	for the Web Version 10.0.23
11 10.	
application you want to create and	approach without select the type of 3 any additional options
Application Type	Options Fill Park Connections
© Sandhinne Androdom	Create User Session
0	(Indudes DataModule)
	Use ServerManager Lite
O Service Application	Use ISAPI]]/veedPool
	Main Form
O 15AP1 Extension	OPDA HTML 3.2
	© wap
Project Name Project1	
Project Directory C:WTEMPADER	
	Qk Gancel
3	
🖗 Projecti – Delphi 2010 – Serv	erContreller 💽 🖸 🕅
79414 (2) (4) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2	37009255日 カジカトの 東市の エオーシナビ 2-40 00-1000 A4700 [see
2.463 21 20 40 40 40 40 40 40 40 40 40 40 40 40 40	日 (1) (2) D · 田 (3) (2) O · O · O Biserve Cartesian 目のanamatature 日のexect (1) 0日 (2) Prioriti (2010-7072-01)
	D.86 a.
and freed	10. 加加 (日・ 1704) アイム
	20 ProstGrap! ⊕ ⊕ FrejetLase
A 875294-123/09 310	a da ChFMall a i≣ Darandabbit pa
Peal Thebasholdfoil	a 11 ServerCartolerpat ₩ 11 Unit.ces
the second s	1 🖬 🖬 US#TREGORUNT. Jan
Active Poliec	
Active Police	
Active Police Policeust 20 100 100 Version 2,0.0	
Active Police Police Red Police Red Pol	
Active Police Policence 20 Poli	<u>Row</u> - V+7
Active Police Po	8 8 8 10 10 10 10 10 10 10 10 10 10 10 10 10



40	[TINForm1]
-	procedure TIMForm1.edtCustCDAsyncEnter(Sender: TObject; EventParams: TStringList);
	// フォーカス取得時、現時点の値(初期値)を変数に保持 FCustCode := edtCustCD.Text; -end;
: 50 c	<pre>procedure TIMForm1.edtCustCDAsyncExit(Sender: TObject; EventParams: TStringList);</pre>
	// フォーカス終ア時、初期値と値が変わった場合、船径を行う if FCustCode ◇ edtCustCD.Text then basin
	// 顧客名欄を初期化 (クリア) する edtCustCD.BGColor := cl#indow; lblCustName.Caption := ;;
60	// コード値が入力されている場合、顧客マスタ検索を行う。 if edtCustCD.Text 〇 ' then
	try // GotCustName開数で顧客コードより顧客名を取得 IblCustName.Caption := GetCustName(edtCustCD.Text);
	except // エラー (例外生成) 時、エラーメッセージを表示する on E: Exception do
70	edtCustCD.BGColor := clWebAQUA; IblCustName.Caption := E.Message;
1	end: end:
74	end: end:
۲2	procedure TIWForm2.IWLink1Click(Sender: TObject):
72	procedure TIWForm2.IWLink1Click(Sender: TObject): var i, j: Integer: sStr: String: sCSVList: TStringList: // 各列海の文字列を保持するリスト sCSVRemList: TStringList: // 各行毎の信頼を哲理するリスト
72	procedure TIWForm2.IWLink1Click(Sender: TObject): var i. j: Integer: sCSVList: TStringList: // 各列毎の文字列を保持するリスト sCSVRcmList: TStringList: // 各行毎の信頼を哲理するリスト memCSVStream: TMemoryStream; // CSVの方用 メモリーストリーム begin //CSVデータを質提する文字列リストを生成 sCSVList := TStringList.Create:
72	procedure TIWForm2.1MLink1Click(Sender: TObject): var i. j: Integer: sStr: String: sCSVList: TStringList: // 各列毎の文字列を保持するリスト sCSVRowList: TStringList: // 各月毎の覚醒を登録するリスト memCSVStream: TMemoryStream; // CSV的力用 メモリーストリーム begin //CSVデータを管理する文字列リストを生成 sCSVList : TStringList.Create: try //CSV用文字列リストの初期化 otSNList.Clear: //各列毎にある健を管理する文字列リストを生成
	procedure TIWForm2.1WLink1Click(Sender: TObject): var i. j: Integer: sStr: String: sCSVList: TStringList: // 各列海の文字列を保持するリスト sCSVRowList: TStringList: // 各列海の文字列を保持するリスト memCSVRream: TMemoryStream; // CSV的力用 メモリーストリーム begin //CSVデータを管理する文字列リストを生成 sCSVList: IStringList.Create: try //CSVR文字列リストの初期化 sCSVList.clear: //AJWELSOUPTIEList.Create: try //CSVRowList: IStringList.Create: try //CBWのList: IStringList.Create: try //CBWのList: IStringList.Create: try //CBWのList: IStringList.Create:
722 110 120	procedure TIWForm2.IWLink1Click(Sender: TObject): var i. j: Integer: sStr: String: sCSVList: TStringList: // 各列帯の文字列を保持するリスト sCSVRowList: TStringList: // 各列帯の文字列を保持するリスト sCSVList: TStringList: // 各列帯の電機を管理するリスト memCSSTream: TMemoryStream: // CSYDDカ団 メモリーストリーム begin //CSVデータを管理する文字列リストを生成 sCSVList: IStringList.Create: try //CSVR文字列リストの初期化 sCSVList: IStringList.Create: try //BBDグリッドよりデータ取得する for i :: 0 to IMGridL.RowCount - 1 do begin // 新聞の値を保持するリストの初期化 sCSVRowList.Clear: //各別の値を見持するリストの初期化
Z2	procedure TIWForm2.1%Link1Click(Sender: TObject): var i. j: Integer: sStr: String: sCSWList: TStringList: // 各列帯の文字列を保持するリスト SSWMCWList: TStringList: // 各列帯の文字列を保持するリスト sSSWList: TStringList: // 各列帯の文字列を保持するリスト memCSIStream: TMemoryStream; // CSWDカ用 メモリーストリーム besin //CSVデータを賛得する文字列リストを生成 sCSWList: IStringList.Create: try //CSWRowList: IStringList.Create: try //原列節にある健全管理する文字列リストを生成 sCSWRowList: IStringList.Create: try //原面グリッドよりデータ取得する for i := 0 to IWGrid1.RowCount - 1 do besin //所帯の健全保持するリストの初期化 sCSWRowList.Clear: //各列節の健子人たに追加 for j := 0 to IWGrid1.ColumCount - 1 do begin sStr := IMGrid1.Cell[i, j].Text; sCSWRowList.Add(sStr);
Z2	procedure IIWForm2.IWLink1Click(Sender: TObject): var i, j: Integer; sStr: String: sCSVList: TStringList: // 各列車の文中利を保持するリスト sCSVRowList: TStringList: // 各列車の文中利を保持するリスト memCSVStream: TMemoryStream: // CSV面力団 メモリーストリーム begin //CSVデータを管理する文中利リストを生成 sCSVList: IStringList.Create: try // 原列車にある塩を管理する文中利リストを生成 sCSNRowList: TStringList.Create: try // 周囲グリッドよりデータ取得する tor i := 0 to IWGrid1.RowCount - 1 do begin // 活剤の値をリストに通知 for j := 0 to IWGrid1.ColumnCount - 1 do begin sStr := IMGrid1.Cell[i, j].Text; sCSVRowList.Add(sStr); end; // CSV用文中利リストに行借額を作入 sUSVList.Add(sSUSVRowList.CommaText); end;
Z2	<pre>iprocedure TIMForm2.1MLink1Click(Sender: TObject): var i, j: Integer: aStri String; aCSVList: TStringList: // 各列節の文字列を保持するリスト sCSVRowList: TStringList: // 各利節の文字列を保持するリスト memCSVStream: TMemoryStream; // CSVガカボ メモリーストリーム Desin //CSVデータを覚醒する文字列リストを生成 sCSVList: IStringList.Create: try //CSVデータを覚醒する文字列リストを生成 sCSVRowList:: IStringList.Create: try //CMEMグリッドよりデータ取得する tor i := 0 to [MGrid1.RowCount - 1 do besin //A 利の値を保持するリストに通知 tor j := 0 to [MGrid1.ColumnCount - 1 do besin sStr := IMGrid1.Cell[i, j].Text; sCSVRowList.Add(sStr); end; //CSV用文字列リストに行慣欄を任入 sCSVRowList.Add(sStr); end; finally sCSVRowList.Free; end;</pre>
Z2	procedure TIWForm2.1MLink1Click(Sender: TObject): var i, j: Integer: sStrisString; sStrisString; sSSMcodist:TStringList: // 各列線の文字列を保持するリスト sSSMcodist:TStringList: // 各列線の文字列を保持するリスト memISNStream:TMEmoryStream; // CSV加力用 メモリーストリーム besin //CSVF-タを覚得する文字列リストを生成 sCSVList:TStringList.Create: try //CSVM文字列リストの卸期化 eCSVList:TStringList.Create: try //周囲グリッドよりデータ取得する for i:=0 to IMGridl.RomCount - 1 do besin //別柳の値を提指するリストの卸期化 eCSVRowList.Clear: //各例の値を提指するリストの卸期化 eCSVRowList.Clear: //各例の値を提指するリストの卸期化 eCSVRowList.Clear: //各例の値を提指するリストの卸期化 eCSVRowList.Clear: //各例の値を提指するリストの卸期化 eCSVRowList.Clear: //SMOMEを提指するリストの卸期化 eCSVRowList.Clear: //SMOMESURAFCENT for j:=0 to IMGridl.ColumnCount - 1 do begin sStr:=IMGridl.Cell[i, j].Text; sCSVRowList.Add(sSVF); end; //CSVM文字列リストに行借録を行入 sCSVRowList.Add(sSVF); end; //CSVMovList.Free; end; ///ST=2+SUJ-AcDSTR memCSVStream:: TMemoryStream.Create: memCSVStream:: TMemoryStream.Create: memCSVStream:Clear: //CSVmateRUTAFCENT
Z2	procedure TIWForm2.1WLink1Click(Sender: TObject): var i, j: Integer; sStri String; sStSW_ist: TStringList; // 各月海の文字列を保持するリスト SSWM_ist: TStringList; // 各月海の文字列を保持するリスト memSNStream: TMemoryStream; // GSYのカ用 メモリーストリーム besin // SSYデータを覚得する文字列リストを生成 sSWList :: TStringList.Create: try // SNM文字列リストの知難化 eSWList :: TStringList.Create: try // 周囲グリッドよりデータ取得する for i :: 0 to IWSrid1.RowCount - 1 do besin // SMMowList.Clear: // SMMowList.Clear: // SMMowList.Clear: // SMMowList.Clear: // SMMowList.Clear: // SMMowList.Clear: // SMMowList.Add(sStr): end: istr :: IMGrid1.Cell[i, j].Text; sCSWRowList.Add(sStr): end: // SYMIX字列リストに行信録を代入 sCSWList.Free; end: // SYMIX字列リストに行信録を代入 sCSWList.Free; end: // SYMIX字列リストに行信録を代入 sCSWList.Free; end: // SYMIX字列リストをメモリーストリームに保存 sCSWList.SaveloStream(SVStream): // スモリーストリームからのダウンパートの語

