

Migaro.Technical Report Stra.F7=カルレポート

ごあいさつ

Information

Migaro.Technical Award 2018 お客様受賞論文/ミガロ.テクニカルアワード	
【部門1】最優秀賞 Delphi/400 駒田 純也 様●ユサコ株式会社	04
ゴールド賞 SP4iの活用による製品検査チェックシステムの構築 SP4i 八木 秀樹 様●極東産機株式会社	12
【部門 2】 優秀賞 Delphi/400 村上 稔明 様●ライオン流通サービス株式会社	18
優秀賞 Delphi/400 寺西 健一様●大阪シーリング印刷株式会社	22
優秀賞 Delphi/400 による無線ハンディターミナルのデータ集約の仕組みの実装 Delphi/400 時西 健一 様●大阪シーリング印刷株式会社	26
Migaro.Technical Report 2018 SE 論文/ミガロ.テクニカルレポート	
Delphi/400 [初級者向け] 楽師 尚之●システム事業部システム 2 課	32
Delphi/400 [中級者向け] 佐田 雄一●システム事業部システム 1 課	40
Delphi/400 REST による Web サービスを活用した機能拡張テクニック [中級者向け] 尾崎 浩司● RAD 事業部 営業・営業推進課	58
Delphi/400 [上級者向け] Google Maps Platform を使用したアプリケーション開発テクニック 福井 和彦/小杉 智昭●システム事業部 プロジェクト推進室	74
Delphi/400 RAD Server を使った新しい多層アプリケーション構築 [上級者向け] 吉原 泰介● RAD 事業部 技術支援課	94
SP4i JC/400 から SP4i へのマイクレーションノウハウ [初級者向け] 吉原 泰介 / 國元 祐二● RAD 事業部 技術支援課	108
Information 既刊号バックナンバー	122

N T E N I T x S

01

A

ごあいさつ

いつもミガロ.製品をご愛用いただき誠にありがとうございます。

「ミガロ.製品をご利用中の技術者の皆様に、日々の開発に少しでもお役に立つような技術情報をご提供 したい」という思いから 2008 年に創刊した『Migaro.Technical Report』は、このたび第 11 号を発刊する 運びとなりました。これもひとえに、ご多忙中にもかかわらず『Migaro.Technical Award (お客様論文)』 にご寄稿いただいた多くのお客様、ならびに『Migaro.Technical Report』に対して貴重なご意見・ご要望 をお寄せ下さった皆様のご支援の賜物と、心より感謝をしております。

今年 IBM i は、発表から 30 周年を迎えました。弊社では、お客様向けのシステム開発および開発ツール の販売を両輪として、長年にわたり IBM i とともに活動を続けてまいりました。主力製品の Delphi/400 と SP4i は、IBM i をより便利にご利用いただくための GUI 化・Web 化ツールとして、これまで大変多くのお 客様にご利用いただいております。また、昨年より取り扱いを開始した Valence (バレンス) は、超高速 開発を実現する「ローコード開発プラットフォーム」へと進化し、IBM i の強力なツールとして積極的に ご紹介してまいります。

さて、今回の『Migaro.Technical Report』も従来と同様に、第1部は「Migaro.Technical Award 2018 お客様受賞論文」、第2部は「ミガロ.SE 論文」の2部構成としています。

第1部の「Migaro.Technical Award」とは、日々アプリケーションの開発・保守に携わるエンジニアの方々 の努力と創意工夫の成果を顕彰することを目的とし、「Delphi/400」「SP4i」「Valence」などの弊社製品を ご利用中のユーザー様を対象に実践レポート(論文)を公募し、厳正な審査・選考のうえ表彰する制度です。 昨年に引き続き、従来のお客様論文に当たる「部門1」と「業務課題を解決した開発技術・テクニック」 を簡潔にまとめていただく「部門2」の2部門構成といたしました。

今回のお客様論文は、「Excel テンプレートによるカスタマイズ性の高い帳票出力機能の開発」や「iPad による工場での製品検査システム開発」など、創意工夫にあふれる論文を多数ご寄稿いただきました。

第2部「ミガロ.SE 論文」では、弊社 SE による技術論文を掲載しております。今回は、「モバイル開発 などに役立つ新しい RAD Server による多層アプリケーション開発」や「天気予報など各種 Web サービ スとの連携手法」、「Google マップとの連携手法」など、さまざまなテクニックを開発に活かしていただく ための技術情報をご紹介しております。本レポートが少しでも皆様の開発・保守のお役に立てば幸いです。

最後に『Migaro.Technical Report』第11号を発刊するにあたりまして、多くのお客様・パートナー様 にご支援、ご協力をいただきましたことを、この場をお借りして、あらためて厚く御礼を申し上げます。

2018 年秋

株式会社ミガロ. 代表取締役社長 上甲 將隆

1

MIGARO. TECHNICAL

REPORT



Migaro. Technical Award 2018

お客様受賞論文/ミガロ.テクニカルアワード

Migaro.Technical Award 2018 最優秀賞

Excelテンプレートを使用した帳票出力機能の開発 ーセルフサービス化への道

駒田 純也 様

ユサコ株式会社 アカデミア事業部 技術部 開発課



ユサコ株式会社 http://www.usaco.co.jp/

海外の学術雑誌、書籍の輸入販売を 中心に事業を展開している。とくに 医学、薬学等の自然科学分野に強み をもつ。近年、学術情報媒体のIT 化が進み、データベースや各種ソフ トウェアの取り扱いを強化。学術情 報を通じ、知的情報の創造、蓄積、 共有による社会貢献を目指してい る。

Excel テンプレートを使用した帳票出 力機能(以下、Excel テンプレート帳票) は、Excel で自由に帳票テンプレートを 作成し、そのテンプレートファイルを Delphi/400で読み込み、Excel ファイル として帳票出力する機能である。【図1】

開発の経緯

Delphi/400以外で開発された社内シ ステムの開発環境やプログラムの維持管 理が負担になってきたため、今回 Delphi/400に移行することになり、そ のシステム内で独自に開発したExcel スタイルシートを使った帳票出力機能も 何らかの形で移行する必要が出てきた。

基幹システムと同一のプラット フォームになることで、マスタデータを 同期する必要がなくなり、ユーザーもロ グインするシステムが減り、デザインや 操作感が統一されたインターフェースに よる作業効率アップのメリットはある。 しかし Excel スタイルシートを使った 帳票出力は複雑なので、同様の仕組みに すると、ユーザーが自身でテンプレート を作成・編集するのは難しい。

そこで考えたのが、テンプレートとな る Excel のセルに出力キーワードを埋 め込む方法である。この方法であれば、 既存の見積書などの Excel ファイルを そのまま使用してテンプレートを作れる ので、既存資産が無駄にならない。使い 慣れた Excel で帳票をデザインできる ため、新しいことを覚える手間もほとん どかからない。

また1つのファイルだけで済むので、 情報システム部門としても管理が容易で ある。

セルフサービス化の 必要性

個人的な感覚であるが、とくにここ数 年はWebシステムが大きく発展してお り、当社でもセルフサービス型の製品を いくつか導入するに至っている。 Delphi/400で自社開発したシステムで も、「ユーザーが自分自身でカスタマイ ズできる」機能を開発フレームワークに 追加実装し、セルフサービス化を意識し て改善に取り組んでいる。

ユーザーが思ったときに、思ったこと ができれば時間のロスを防げる。管理す べきシステムが増えれば増えるほど、そ れぞれに要する時間は減るが、システム 間連携の設定・構築に時間を取られるの で、情報システム部門としてもセルフサー ビス化の推進は必須だと考えている。

テンプレートをユーザーが 自身で編集して アップデート

テンプレートを共有フォルダ上に配 置することで直接編集したり、いったん ローカルフォルダにコピーしてから編集 しアップデートすることも可能となる。

出力キーワードは「\$ {」で始まり、「} 」 で終わる形式で、その中にテーブル名 (TClientDataSet 名) とフィールド名 の順に「/ (スラッシュ)」で区切り、記 述する。「Header」テーブルのフィール

41	帳票出力結果							
		請求	書			<u>請求Nb:12345</u> 請求日:2018/	56 /8/24	
	<u>駒田 様</u>							
		合計金額	ᥠ28),380 (RL).	<u>}</u>)			
	行No	タイトル	単	●価 数	量	金額		
	1 Delph クオブミ	iテクニック集	3 87 - 1	3,800	1		3,800	
	3 帳票:	デザイナー デザイナー	9),800	2	1	9,600	
2	明細を含む単票形	式の帳票テンプレ	- ト					
0	G12 🔻	$\times \checkmark$	fx =E1:	2*F12				
	AB C	D	E	F		G H		
	3	清步	圭		請	求No \${Header/No	b }	
	4 5				請	求日 \${Header/Da	te}	
	6 <u>\${Header/</u> 7	Atena} 栜						
1	8 9 10 (The	合計並額 ¥	#VALUE	(れ込)	<u>소</u>	农百		
1	1 \${#明細開	レール 始#}		<u>東入王</u>				
1	13	(\${Detail/ itie}	⊅{Detall⁄ A	(mpiDetail/	یا \${	#VALUE #明細終了#}		
		i						
3	リスト形式の帳票テ	・ンプレート						
ſ	C3 ×	: X ./	f. =F	3*F3				
L			в		ם	F	F	1
_	1 注文No	▼ タイト J	l 🔽	金額				
	3 \${ChkLst/Ord	lerNo} \${ChkL	st/Title}	#VALUE		\${ChkLst/Price}	\${ChkLst/Qty} \${#明細終了#}	
	5 <u>集計</u>			#VALUE,				

ド「Atena」を出力したい場合は、「\$ {Header/Atena}」となる。「様」や「請 求 No」などの文字列も、同じセルに自 由に記述してよい。

明細部分は、キーワード「\$ {# 明細 開始 #}」と「\$ {# 明細終了 #}」で囲 む形で定義する。明細キーワード行は出 力時には削除されるので、実際に出力さ れることはない。

図3は価格と数量の計算結果のみを 金額列に出力するようにしており、印刷 範囲として設定した範囲外に価格と数量 を出力し、Excelの計算式を使って金額 を計算している。

このように Excel の機能を利用すれ ば、工夫次第でいろいろな帳票を作成で きる。【図 2】【図 3】

なおフィールド名はアルファベット 表記であり、現場ユーザーにはわかりづ らいため、当社では該当機能で利用可能 なフィールドを検索できる汎用画面を用 意している。実装も簡単で、セルフサー ビス化のハードルを下げる役割もあると 考えている。【図 4】

帳票出力処理の流れ

おおまかな流れは、【図 5】のとおり である。説明の都合上、単票と呼ばれる 【図 1】のような帳票での明細部分以外 を「単票部分」と書いている。

機能全体を1つのデータモジュール として作成しており、出力キーワードを 格納する TClientDataSet (cdsKeywd) を配置。デザイン上であらかじめ必要な フィールドとして、セル位置とその内容、 テーブル名とフィールド名、明細部分の 開始・終了・明細内部であることを示す フラグを格納するためのフィールドが存 在している。【図 6】

帳票出力部分のコーディングでは、 データモジュール内の以下の引数をもつ 関数に適切な引数を渡して呼び出すだけ でよい。【図 7】

- * 引数1 [i]: string 型=テンプレート File のフルパス
- * 引数2 [i]: string 型=保存先 File の フルパス
- * 引数3[i]:TStringList型=テンプレー ト内で指定されたテーブル名 (TClientDataSet名)のリスト

- * 引 数 4 [i]: TForm 型 = データ用 TClientDataSet が定義されている対 象フォーム
- * 引数5 [i]: Boolean 型 = Excel を画 面表示するかどうか

上記のうち、引数1と引数2はそれ ぞれのファイルへのフルパスで、引数1 が存在しない場合はエラーとなる。引数 2の保存については強制的に上書き保存 したり、出力ファイル名に日付や実行 ユーザー名を加えたりと、関数側で動作 を設定している。

引数3は、帳票テンプレートに登場 するテーブル名 (TClientDataSet 名) をすべて列挙した TStringList 型の変数 を渡す。

引数4は、引数3で渡された TClientDataSet名をFindComponent で取得するため、各データセットが定義 されている対象フォームを渡す。

引数5は処理終了時に、それまでバッ クグラウンドで処理していた Excel を 画面表示するかどうかを Boolean 型で 渡す。False の場合、つまり画面表示し ない場合の Excel プロセス終了はソー スコード上で記述する必要があり、 True の場合は Excel 画面が表示される ためユーザーに委ねられる。

関数内部の処理の流れは、テンプレー トファイルを開く、Clone カーソルを作 成する、各シートごとに処理する、といっ た順になっており、シートごとにキー ワード解析、データ差し替え処理を実行 している。【図8】【図9】

ブック全体のキーワード解析を最初 に実行してから、各シートを処理する方 法も考えられるが、速度的にそれほど差 はなさそうなので、実装方法はコーディ ングの好みになるかもしれない。なお呼 び出し元の画面に影響を与えないため に、Clone カーソルを作成している。

キーワード解析は、まず検索を行い、 セル位置や内容を cdsKeywd へ登録す るという手順で実行している。ポイント は、UsedRange プロパティで使用され ているセル範囲を取得してから行うこ と。それにより範囲をあらかじめ絞れる ので、余計なオーバーヘッドを減らせる。 【図 10】

データ差し替え処理は、単票部分と明 細部分を別々にループ処理している。1 つのループでも処理可能だが、ソース コードの可読性を優先した。同じ cdsKeywd内に単票部分と明細部分が 混在するため、Filter プロパティを使い 絞り込んでから、それぞれのループを処 理している。【図 11】

明細部分のデータ差し替え処理に関 する注意点には、Variant型の2次元配 列を使用し、明細範囲を一括で書き込む ことが挙げられる。もしセルを移動しな がら明細を1セルずつ書き込んでいった 場合、セル移動のオーバーヘッドが大き いため、速度が劇的に低下し、実用に耐 えない。

配列を Variant 型で定義することに も理由がある。たとえば文字列型で定義 すると、セルの書式などが自動的に処理 されないためである。【図 12】

実際に出力して 感じたこと

考えていたよりも使い勝手はよさそ うな印象で、もともと利用対象として考 えていた Delphi/400 へ移行するシステ ムだけでなく、応用範囲は広がりそうで ある。とにかく帳票設計が簡単なので、 ユーザーのアイデアを引き出すツールに なるかもしれない。

当社は FastReport も導入している が、この Excel テンプレート帳票があ れば、必要ないかもしれない。

帳票サーバーも導入しており、バッチ 処理で出力される帳票には威力を発揮し ているが、入力チェックリストの類はス プールから出力指示を出さなくても、直 後にデスクトップ画面に Excel として 表示したほうがデータとしても処理でき るので、いろいろと手間を省けるように 思う。

また出力関数を呼び出した際に、シー トごとにキーワード解析とデータ出力を 単票部分と明細部分に分けて実行する仕 様なので、同様のループを複数回実行す ることでオーバーヘッドが速度的に問題 にならないか心配であった。しかし今の ところ、それほど気にならない状況であ る。

今後の課題・計画

課題としては、Excel への依存が挙げ



られる。VBA の仕様が急に大きく変わ るとは考えにくいが、コントロールでき ない部分であるため Office のバージョ ンアップなど大きな変化には注意が必要 である。

今後の計画としては、ユーザー自身で 各画面に対して帳票出力を設定できるよ うな仕組みを考えている。すでに TClientDataSetのフィルタ内容をユー ザー自身で編集し、10個まで保存でき る機能を作っているが【図13】、同様に Excelテンプレートのファイルパスをい くつか登録しておき、選択した帳票を出 力すれば短期間で作成できるはずであ る。

実はこの Excel テンプレート帳票は、 当社で導入しているセルフ型の Web ア プリ開発ツールにインスピレーションを 受け開発したものであるが、そのツール にはテンプレート側とデータベース側 フィールドのマッチングを設定する機能 がある。テンプレート側のキーワード部 分を自由に命名できるので、ユーザーに とって見やすいテンプレート作成が可能 になっている。

現状ではフィールド物理名をキー ワードにしているため、少々見づらい。 そこで、そういったマッチング機能も検 討している。あとは、マスタなどのコー ドからその名称に変換する処理を実行で きるよう、明細部分のレコードごとにイ ベントが発生するような仕組みも検討し ている。

最後に

当社では本稿で紹介した以外にも、セ ルフ化を意識した機能の実装や、 Delphi/400以外でもセルフイノベー ションを支援する製品を導入している。 情報システム部門の人数は限られている ので、現場の意欲を高め、その意欲を打 ち消さないために、そういった環境の重 要性と影響力の高さもひしひしと感じて いる。

今回のようなセルフ化や自動化に よって、現場と情報システム部門の双方 で削減できた時間を創造的な業務を行う 時間へとシフトさせていく。そういった スパイラルへとつなげていきたい。

Μ

図7	呼び出し部分のサンプルコード
Γ	Template File Path ≔'¥¥サーバ¥Excel Template¥チェックリスト.xlsx'; //テンプレート
	SaveFilePath:=\\\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\
	CDSList.=TStringList.Create. try
	CDSList.Add('cdsHeader');
	CDSList.Add('cdsDetail'); isVisible:=False: //Excel をバックグラウンド処理
	sRet=DataModule.ExcelOutput(TemplateFilePath,SaveFilePath,CDSList,Self,isVisible);
	if sRet <> " then ShowMessage(sRet) else ShowMessage('Excel 出力完了。'); finally
	CDSList.Free;
	endi
₹8	Excel OLE利用とテンプレート読み込みのサンプルコード
ſ	vExcel, vWkBook, vWkSheet: Ole Variant;
	hExcelhWnd: THandle;
	vExcel == Create Ole Object('Excel.Application); //Excel OLE 準備
	hExcelhwnd:=vExcel.hwnd: //Excel フロセス終了処理で利用する vWkBook:=vExcel.WorkBooks.Open(Template File Path); //テン プレート読込
L	
<u>8</u> 9	Cloneカーソル作成のサンプルコード
Γ	//Clone を作成し配列に格納する
	SetLength(arCdsCln, CDSList.Count);
	for i=0 to (CDSList.Count · 1) do begin arCdsClp[i]:=TClientDataSet Create(Self):
	arCdsCln[i].CloneCursor(TClientDataSet(Frm.FindComponent(CDSList[i])), False);
	arCdsCln[i].DisableControls;
	arCasCin[1]. Name.=CDS List[1], end:
図10	キーワード検索のサンプルコード
ſ	vWkSheet, vRange, vCellTmp: OleVariant;
	~~~~~~~~~ /ディタが移動されているやり筋固な取得(海白 0-11 あま(何か) ていかぼそすかで得みがまで)
	W/ - ラ Whath Con Constant Constant Cont Constant Consta
	/Range 内でキーワードセルを検索 ※実際のコーディングでは最初に検索した位置までループしている
	vCellTmp:=vRange.Findl'\${*リ; //キーリード情報を登録(単画部分)
	cds Keywd. Append Record([列, 行, セル値, テーブル名, フィールド名, 0, 0, 0]);
L	

vCel	l Tmp: Ole Variant;			
~~~ //cds	~~~~~~~~ Kewwdのループ処理			
//所象 Field	としい値のキーワード部	分をデータベースのフィール daKaywyd ThiNama Valua)	、ド値に差し替える Find Field (adaKeywyd Fid Name Velue)	
vCel	lTmp:=vWkSheet.Cells	cusikeywa Tonyanie, value). [cdsKeywdRowNo.Value, cd	lsKeywdCol No. Value];	
vCel '\${	'+ cdsKeywdTblName.'	tring Replace (cds Reywd For Value + '/ + cds Keywd FldNa	mula.value, ame.Value + '}', Field.DisplayText,[]);	
× Γf	FindClnCDSJ は arCds	sCln 配列から指定した Nam	ae を持つ Clone を TClientData.Set 型で返す関	人物
12 明細	部分の一括データ貼り	すけ処理サンプル		
vWk arMe	Sheet: Ole Variant; sisai: array of array of f	Jariant; /岬細部分のデータね	格納配列	
/胡制 vWk Va	1範囲に配列値を一括で! Sheet.Range[vWkSheet riant(arMeisai);	貼り付ける t.Cells[開始行,列],vWkSheet	t.Cells[終了行, 列]].FormulaR1C1:=	
13 フィル	∕夕内容保存機能			
13 フィル DEPN ※保	・タ内容保存機能 AME Like '%センター 字されたフィルタ内容が	%' が初期値として表示される	5 ~ フィルタ 解除 5	
13 フィル DEPN ※保存	 ✓タ内容保存機能 AME Like '%センター・ すされたフィルタ内容が 	%' が初期値として表示される 部署=センター(Z)	5	
13 フィル DEPN ※保 一般	 /タ内容保存機能 AME Like '%センター 字されたフィルタ内容が 受注 仕ジ 	%' パネル期値として表示される 部署=センター(Z) Slot.2	5	
13 フィル DEPN ※保 一般	・タ内容保存機能 AME Like '%センター 字されたフィルタ内容が 受注 仕ン	%' が7期値として表示される 部署=センター(Z) Slot.2 Slot.3	5 7イルタ解除 5 5 5 2 保存 岡一町 ?	
13 フィル DEPN ※保 一般	 ク内容保存機能 AME Like '%センター? 字されたフィルタ内容が 受注 仕ン 	%' が初期値として表示される 部署=センター(Z) Slot.2 Slot.3 Slot.4	5 5 5 5 5 5 5 5	
13 フィル DEPN ※保 一般	 タ内容保存機能 AME Like '%センター? 字されたフィルタ内容が 受注 仕ン 	%' が初期値として表示される 部署=センター(Z) Slot.2 Slot.3 Slot.4 Slot.5	5 フィルタ 解除 Slot.2 保存 順一覧 ?	
13 フィル DEPN ※保 一般	 ク内容保存機能 AME Like '%センター? 字されたフィルタ内容が 受注 仕ジ 	%' 洋刀期値として表示される 部署=センター(Z) Slot.2 Slot.3 Slot.4 Slot.5 Slot.6	5 フィルタ 解除 Slot.2 保存 順一殿 ?	
13 フィル DEPN ※保 一般	 ク内容保存機能 AME Like '%センター・ 字されたフィルタ内容が 受注 仕ン 	%' 洋刀期値として表示される 部署=センター(Z) Slot.2 Slot.3 Slot.4 Slot.5 Slot.6 Slot.7	5 フィルタ 解除 Slot.2 保存 順一職 ? 第 前除	
13 フィル NEPN ※保 一般	 ク内容保存機能 AME Like '%センター、 字されたフィルタ内容が 受注 仕ン 	%' が可期値として表示される 部署=センター(Z) Slot.2 Slot.3 Slot.4 Slot.5 Slot.6 Slot.7 Slot.8	5 フィルタ 解除 Slot.2 保存 順一職 ? 第 前除	
13 フィル NEPN ※保 一般	 ク内容保存機能 AME Like '%センター、 字されたフィルタ内容が 受注 仕ン 	%' が利期値として表示される 部署=センター(Z) Slot.2 Slot.3 Slot.4 Slot.5 Slot.6 Slot.7 Slot.8 Slot.9	フィルタ 解除 Slot.2 保存 順一毗 ?	



Migaro.Technical Award 2018 ゴールド賞

SP4iの活用による製品検査チェックシステムの構築 -iPadによる検査入力で各工程でのチェック漏れを防止する

八木 秀樹 様

極東産機株式会社 システム開発室 課長



極東産機株式会社 https://www.kyokuto-sanki.co.jp/

製造機器メーカーとして創業以来、 職人さんの快適な職場環境作りと消 費者の豊かな生活空間作りを2 柱として、伝統技術と先端技術の融 合により、ユニークなオリジナル商 品を開発。畳製造機器はもとよりイ ンテリア施行省力機器、カーテン縫 製機器等、幅広く事業を拡大してい る。

業務課題は検査漏れの 撲滅

極東産機は、畳業界・インテリア施工 業界のニーズに合わせたオリジナル製品 を開発・製造している。工場では、畳・ インテリアの施工時に利用する機械を製 品として製造している。製品製造時の検 査段階で、各検査項目の検査結果の記録 を用紙に記入していたが、記入漏れが あった場合でも誤って完成処理されてし まうケースが見られた。

このため社内では、「検査チェック漏 れまたは不合格の項目があれば、次の工 程へ進まないようなシステムはできない か」「検査記録をタブレットなどでその 場で入力したい」「検査項目の追加や改 定を迅速に行いたい」「検査記録結果を 画面で確認したい」といった要望が多 かった。

以前、取引先との Web-EDI を JC/400 で構築したこともあり、JC/400 の後継 製品でタブレットにも対応している SmartPad4i(以下、SP4iと記載)の採 用を決めた。

採用決定理由は、次のとおりである。

- JC/400のスキルをそのまま活かせる
- (2) iPad などモバイル端末のシステム を開発できる
- (3) IBM i とリアルタイムにデータ処理 できることで、検査チェック漏れが あれば、次の工程へ進まないシステ ムの構築が可能である

システム構築・運用に おける問題点と対策

SP4iと iPad による工場での製品検査 システム構築に際して、以下の課題を洗 い出し、その対策を行った。

- 【課題1】タブレットを工場内の各検査 場所で使用したい。
 【対策1】工場内に無線LAN環境を増
- 【対東1】 上場内に 無線 LAN 東境を増 強した。

- 【課題 2】 タブレットは iPad を採用した が、作業者が iPad 操作に不慣 れで馴染めない。
- 【対策2】 iPad に馴染むため、若手中心 で iPad の使用方法の講習会を 開催した。教育を受けたメン バーから、他のメンバーへ指 導してもらうこととした。
- 【課題3】製品(機械)の種類(以下、 機種)によって検査項目が異 なる。
- 【対策3】 機種ごとの検査項目を IBM i のマスタに登録する。ユーザー 自身が PC 端末から IBM i に 登録できるようにした。
- 【課題4】検査によっては数値の記入が 必要なものがある。
- 【対策4】数値入力にも対応できるよう にした。また入力した数値に よって、自動的に合否を判定 できるようにした。



	【課題 5】	すべての機種の検査について、
		紙のチェックリストから iPad
		でのチェック入力に変更でき
		るか。
	【対策5】	機種ごとに具体的な検査項目
		が違うため、まずは主要1機
		種に限定し iPad に切り替えて
		運用することにした。その後、
		計画的に iPad で検査入力する
		機種を増やすことでスムーズ
		に移行できた。
	【課題6】	検査チェックシステムの承認
		機能を iPad システムでどのよ
		うに実装するか。
	【対策6】	紙で運用していた場合は検査
		担当者、確認者、承認者と検
		印していた。iPad ではログイ
		ン ID によって、メニューを分
		けることで対応した【図1】。
		入力時の情報(いつ・誰が)
		は自動記録されるので、入力
		した際の情報の照会・確認が
		可能になった。
	【課題7】	重点検査項目についてはダブ
		ルチェック(他の担当が再度
		チェックして記入する)を行
		う必要があるが、同じ運用が
		可能か。
	【対策7】	各機種の検査項目の登録で、
		ダブルチェックが必要な項目
		を検査項目として登録する。
		基本的な検査チェックの入力
		項目は、1回目のチェックで
		入力し、別の担当者が2回目
		のダブルチェック項目だけを
		入力することとした。【図2】
	【課題8】	検査項目によっては、合格時
		の写真を見たい場合があるが
		対応は可能か。
	【対策 8】	入力画面に写直右無の項目を
		設置]、写直「右」をタップ
		は という 学具「行」と シックククククククククククククククククククククククククククククククククククク
		サムニシシーカ 【回り】
	【課題9】	紙の運用では特定の機種の検
	_	査時に、図面に検査結果を記
		入している。iPad で同様の対
		応が可能か。
	【対策 9】	従来の紙の図面と同等のレイ
14		

アウトを iPad の入力画面上に 表示し、その画面に検査結果 を入力することとした。【図3】

スムーズな稼働へ 向けての工夫

今回の iPad 検査システム開発プロ ジェクトを進めるにあたり、考慮した点 を記載する。

- (1) 検査対象製品を1機種限定で運用開 始することとし、そのために必要な システムの要望事項を絞り込み、シ ステム構築を進めた。
- (2) 検査項目のマスタ化を関連部門と進 め、合否判定や数値入力、写真添付、 図面添付などデータ整備を進めた。 また、類似の機種は検査項目が共通 である場合が多いので、検査項目の 登録機能も共通化できるようにし た。
- (3) iPad の操作方法などシステム運用 に詳しい現場メンバーを数名、事前 に教育した。
- (4) iPad からの入力をできるだけコー ド化することにより手入力を簡素 化し、入力作業が負担にならないよ うにした。【図4】
- (5) システム作成における疑問点を解消 する。SP4iの開発テキストには、 よく使用する機能の解説や事例が あるため、テキストを読むだけでほ とんどの問題点を解決できた。エ ラーの原因が推測できない場合は、 ミガロ.のテクニカルサポートを積 極的に活用した。このサポートによ り、計画を遅らせることなく進めら れた。

SP4iの活用による 製品検査チェックシステ ム本稼働の所感

「検査チェック漏れ防止」という目標の 達成

検査担当 (iPad) →確認担当 (iPad) →承認者(iPad)→製品完成の入力(PC 端末)のワークフローが確立し、検査 チェックの入力漏れや検査結果が不合格 であれば、次の段階へ進めないようにし

た。結果として、検査合格品の場合にの み製品完成の入力が可能となる仕組みづ くりができた。

直感的な操作性によるスムーズな運用

直感的に操作できる画面であるのに 加え、iPad の操作説明会を事前に何回 か実施したこともあり、運用開始後の問 い合わせはほとんどなかった。担当者、 確認者、承認者別のメニューも、各メ ニューの用途が一目でわかるようにコメ ントなどを工夫した。【図1】

またスマホの操作に慣れていたメン バーが多いことも、順調に運用が開始で きた要因と考えられる。

iPad の利点

検査の都度、所定のパソコン設置場所 へ登録しに行く必要がなく、iPad だけ で入力できる【図2】。また検査結果入 力後に、誰が検査したかを手元の iPad からすぐに確認できる【図5】。このため、 iPad を利用した本システムは現場から は好評だった。

今後の展望

iPad を活用した生産現場の在庫管理 システム (IBM i) を SP4i で構築済み だが、現在、このシステムのリニューア ルに取り組んでいる。改善のポイントは、 ハンディターミナルでは実現できなかっ た画面の情報量と操作性を iPad と iPad 用のバーコードスキャナにより実現する こと。これにより、在庫管理をさらに効 率化する予定である。

今後とも、タブレット活用による生産 現場での作業効率向上と見える化できる システム改善をさらに推進していきた 120

Μ

8 #	8111435 4 4 2091
1 王程コード	10001 影響 確認工程 担当者コード 7947
管理番号	FA - 13 - 43 変更函数 3 寸法・運転チェック入力 🌘 有
持記事項	
	メンノすると、小広・運転チェンジベルへ
(THE 1977)	
	JE-S-25VD
Dy Ha	16918-2A 程 番 2091 易 番 8111435
 裁断寸法チェック 	ウは3週ポードで行うこと <u>丈尺 管理番号</u>
・表の上から順に表	大曲 管理番号
	登録する 登録しない
★運送位で記入 入力寸法	【既 新 寸 法 詳 差 】 入力寸法 第とし 丈尺 丈尺 大 曲 大 曲 備客区分
(*尺)	(*寸) 寸法 上期 下期 下期左 下期右 (*寸)
6	4 2
6	2 2
6	
5	8 3 .
5	5 3
5	
8. II 	
2000 C C C C C C C C C C C C C C C C C C	
必須入	カとなる部分を強調
必須入	力となる部分を強調
必須入	
_{必須入} 検査チェン	^{、カとなる部分を強調} ック入力状況照会
_{必須入} 検査チェッ	カヒなる部分を強調
_{必須入} 検査チェッ	カビなる部分を強調
必須入 検査チェッ ロット番号	カとなる部分を強調 かク入力状況照会 16918-2A エ程コード ロード検索
必須入 検査チェン ロット番号	カとなる部分を強調 アク入力状況照会 16918-2A エ程コード ロード検索
必須入 検査チェッ ロット番号	カとなる部分を強調 アク入力状況照会 16918-2A エ程コード ード検索
必須入 検査チェン ロット番号	かク入力状況照会 16918-2A エ程コード ード検索
必須入 検査チェン ロット番号	yク入力状況照会 16918-2A エ程コード 「住口ード 」 「住口ード」」 「住口ード」」 「住口ード」」 「「快来」」 「し」 「し」 「し」 「し」 「し」 「し」 「し」 「
必須入 検査チェン ロット番号	yク入力状況照会 16918-2A I程コード - F検索 選択 コード エ程名
必須入 検査チェン ロット番号	
必須入 検査チェン ロット番号	ウク入力状況照会 16918-2A エ程コード レード 正程コード レード 正規 コード 正規 1001 断断 確認工程
必須入 検査チェッ ロット番号	x2 x2 x2 x2 x2 x2 x2 x2 x2 x2
必須入 検査チェン ロット番号	ウク入力状況照会 16918-2A エモコード エモコード レード 正世コード レード 正世コード レード エモコード レード エモコード ロード エモロード ロロロ1 BFE 確認工程 ロロロ2 ロロロ2 レロロ1 レロロ2 レロロ1 レロロ2 レロロ1 レロロ2 レロロ1 レロロ2 レロロ1 レロ1
必須入 検査チェン ロット番号	クク入力状況照会 16918-2A エ催コード レード 正確コード レード 工程名 10001 新邦 確認工程 10001 新邦 確認工程 10001 10001 新邦 確認工程 10001 10001 新邦 確認工程 10001 新邦 確認工程 10001 新邦 確認工程 10001 新邦 確認工程 10001 1001 10101 10101 10101 10101 10101
必須入 検査チェン ロット番号	
必須入 検査チェン ロット番号	ADA ADA ADA ADA ADA ADA ADA ADA
必須入 検査チェン □ット番号	
必須入 検査チェン □ット番号	ウク入力状況照会 10318-2A 10319-2A 10319-2A

図5 検査結果の詳細照会

	FEYGU				
 ž	811 36	(* ; * ; * ; *	т@⊐_К	10001	_

	811	-36	r	· · · · ·		Ⅰ程コード	10001	剧部 確	認工程
機	5052	1	凹者	F001	戚. *	뽌	第2担当者	8068	щі «Д
確認者		1601	杉::	ļ±					
チェッ	ク項目			判定基準	.	備考	(測定値等)	検査数値	判定結果
(1)脚部の開き	見合	<u>8</u>	『上側内寸と支 (上側) = 5 m	点板部内寸測定 m~15mm	(下側)			5.00	合 格
(2)原反芯梅、 認	上蓋の取付	確無	墅なく取り付く	こと 。					合 格



配車支援システムを Delphi/400で再構築

ライオン流通サービス株式会社 事業部



部門 2

ライオン流通サービス株式会社 http://www.lion-logi-s.co.jp/

ライオン株式会社100%出資の物 流子会社として、全国のグループ物 流拠点、および協力物流事業者への 委託業務を統括。倉庫管理・在庫管 理・輪配送管理など、グループの物 流業務全般を担っている。輪配送 における CO2 削減など物流業務改 善への積極的な取り組みを行ってい る。

業務課題

ライオングループの物流を担う当社 では、配車支援システムにより、翌日の 運送計画を立案している。

従来の Visual Basic で作成した配車 支援システムには、以下の問題点があっ た。

- (1) IBM i上の各データを毎回 PC ヘダ ウンロードする必要があるため、処 理に時間がかかる。
- (2) 共有の PC で稼働しているため、別 の担当者が操作中は、操作ができな い
- (3)メイン処理の「運行計画編集処理」 では、ドラッグ&ドロップで使用で きず操作しにくい。

これらの問題点を解決するために、 IBM iのデータを直接参照し、各自の PCから操作可能な Delphi/400 を使用 して、システムを再構築することにした。

画面機能の詳細説明

メイン処理の「運行計画編集処理」を Delphi/400 で新規作成した。

表示ボタンを押下すると、IBM iの運 行関連 DBをSQLで抽出し画面に表示す る。【図 1】

一番左の「車番」の列には、車両ごと の拠点(事業者)、予定/実績費用、可 能積載重量が表示される。

第1運行以降の列には、配送拠点 (FROM-TO)、到着/出発予定時間、編 成番号、積載予定重量を表示する。当初 スケジュールしていた車両から別の車両 に運行計画を変更する場合、ドラッグ& ドロップによる運行編集が可能となっ た。

たとえば単一編成Naは、ドラッグ&ド ロップにより運行編集できる【図2】。 また同一編成Naの複数の運行計画をまと めて、ドラッグ&ドロップにより運行編 集できる。【図3】

画面による業務課題の 解決

各業務課題は、以下のとおり解決した。

- Delphi/400 は、IBM iの DBから処 理に必要なデータのみ読み込むた め、全件ダウンロードによる処理時 間は不要になった。
- (2) 配車担当者は座席を移動することな く、各自のPCから配車支援システ ムの入力が可能となった。
- (3)運行計画のドラッグ&ドロップにより、新しい車両への割り付けが可能となった。さらに再計算処理(一時保管)ボタンの機能や、編成No検索機能を新たに追加することにより、1日当たり数十分の時間短縮を実現した。

Μ

81	E B 2018/07/25	·示		缩小表示	編成処理	表示順変更	商計算処理(一時(早安)	<u>i</u> t mos
N0.		36 1 al (1	11 E 2 M 17	560 M (1	第4運行	965 il 17	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	第7運行
1	1304 北開東 12,200 出日27.850円 町日 10,000円 月千切 10,000円	00:30 09:51 09:50 09:11 (30-9593) (9,024)	千葉中 → 03:51 13:14 10:11 13:34 (34-0129) (6,264)					
2	1525 北間東 12,500 山田20,100円 町日10,000円 月平均 10,000円	東流波→ M0L 00:30 10:40 00:50 11:00 (30-9679) (9,402)	平置中→ 11:40 15:05 12:00 15:25 (34-0130) (6,264)					
3	1500 北間支 12,000 山日40,200円 村日10,000円 月平12 10,000円	千葉中→ 北関東 00:30 11:55 00:50 52:15 (34-0136) (0,256)	平旗中→ 紙志面 15:20 16:00 15:40 16:20 (34-9731) (6,264)					
4	6983 北関東 12,600 皆百6.756円 封日 10.000円 日子切 16.000円	平葉中 → 紙市東 00:30 09:10 00:50 09:30 (34-0104) (7,307)	∓ (⊈ Φ → 09:50 19:15 10:10 13:35 (34-0131) (6,264)					
5	7726 北関東 12,500 由日17,500円 町日10,500円 月平均 10,000円	千葉中→ 紙志恵重 00:30 03:10 08:50 03:50 (34-0106) (7,907)	〒菜中→紙売車 09:50 10:30 10:10 10:55 (34-9732) (6,264)					
+	1235 北関支 12,000 出日33,650円 封日 10,000円 月平均 10,000円	千葉中→ 紙市面 00:30 09:10 08:50 09:20 (34-0107) (7,387)	千葉中→紙市版 09:50 10:30 10:10 10:50 (34-9733) (9,024)	MiL → 北間 11:50 14:30 12:10 14:50 (38-9858) (8,345)	×			
7	8776 北間変 12,700 当日 33,450円 町日 10,000円 月平町 10,000円	ギ重中 → 紙売車 08:30 09:10 08:50 09:30 (34-9108) (7,387)	千葉中→紙恵頂 09:50 10:20 10:10 10:50 (34-3734) (3,024)	P №3 → MGL 12:20 14:50 13:20 15:10 (38-3588) (3,293)				
	0344 東京 18,800 町日 10,000円 月平均 10,000円	千葉中 中 括 7 続 08:30 03:20 08:50 03:50 (34-9720) (14,603)	千葉中 ⇒ 風密道 10:30 11:10 10:50 11:30 (34-9722) (14,609)	千葉中⇒紙用書 11:50 12:50 12:10 13:20 (34-9723) (14,€00	 千葉中 寺山五川 13:40 14:00 14:00 14:20 14:20 14:20 14:20 14:20 			
	6408 발표 18,800 방원 10,000円 시구13 10,000円	千葉中 → 話 ヶ頃 08:50 05:50 08:50 03:50 (34-9721) (14,608)	千葉中 中 紙西車 10:30 11:10 10:50 11:30 (34-9724) (14,608)	千葉中 中山五子 11:50 12:28 12:10 18:20 (34-5725) (14,688	* 千葉中 中山五県 13:30 14:00 13:50 14:20 (34-9727) (14,600)			
10	4045 東京 12,200 山日14,550円 町日10,010円 月平均 10,000円	$ \begin{array}{cccc} W \rightarrow HiL \\ 08:30 & 08:51 \\ 08:50 & 09:11 \\ (30-9839) & (7,491) \end{array} $	M6L → 座部 05:11 15:01 09:31 11:21 (38-9395) (8,390)	1:11 - ±03 12:14 18:55 13:20 17:15 (45-9769) (7,274)	1・田里 → 其州田 15:00 17:09 19:09 17:00 19:09 17:00	1 神俗ス + 北朝東 15:05 17:00 (15:20 17:20 (15:20 17:20		
11	4213 重点 12,500 山日40,200円 村日10,000円 月平均 10,000円	千葉中 → 北間東 08:30 11:55 08:50 12:15 (34-9359) (8,131)	千葉中 → 純市直 15:20 16:00 15:40 16:20 (34-3625) (10,291)		(1.5.420.00) (1.2.41)	- Contraction - Contraction		
12	5000 東京 12,000 山日 64,450円 朝日 10,000円 月平均 16,000円	手展中→北間東 00:30 11:55 00:50 12:15 (34-9778) (0,256)	千葉中→紙恵庫 15:20 16:09 15:40 16:20 (34-3626) (10,291)	MGL → 建築 11150 10:40 12:10 14:00 (30-9417) (1,071)	総領33 → 産額 11251 13220 12211 12240 (08-5417) (1,071)	8		
13	5707 東京 12,500 毎日35,599円 町日 10,000円 月平均 10,000円	 	∓ St Φ → 12:40 16:25 19:20 16:45 (04-0122) (7,960)					
14	1457 東京 12,500 山日本0,200円 町日10,000円 月平45 10,000円	千葉中→ 北国東 00:30 11:55 00:50 12:15 (04-9779) (0,250)	千葉中→紙志恵 15:20 16:00 15:40 16:20 (04-9627) (10,939)					

図2 運行計画ドラッグ&ドロップ(単一編成)

出	戦日 2018/07/25 2	更示		縮小表示	編成処理		表示順変更	再計算処理(一時(常管)		រាំ ៣៩៩៣
Ni.		8 1 8 1	第 2 運 fī	第3運行	- 1	第4運行	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	第6道fī	第7课行	1 35
1	12,200 出日27,850円 町日11,000円 月平均 13,000円	08:30 08:51 08:50 09:11 (20-9583) (9.024)	+ K Φ → 09:51 13:18 10:11 13:36 (34-0129) (6,264)							
2	1525 北間東 12,500 始日20,100円 期日10,000円 月平均 10,000円	90 53 28 → MOL 08:30 10:40 08:50 11:00 (30-3679) (3,407)	+∰ ↔ → 11:40 15:05 12:00 15:25 (34-0130) (6,264)							
3	8588 開東 12,800 山田42,350円 町日10,000円 月平均 10,000円	千葉中→北間東 08:30 11:55 08:50 12:15 (34-0196) (8,256)	千葉中→紙市夏 15:20 16:00 15:40 16:20 (34-9731) (6,264)	M5 → ± 11:50 14: 12:10 14: (30-5050) (0,34	M 9 30 50 5)					
4	6963 北間東 12,600 山田4,750円 町日11,000円 月平均 16,000円	午届中 → 純市重 08:30 09:10 08:50 09:30 (34-0104) (7,987)	∓ ∦ 中 → 05:50 13:15 10:10 13:25 (24-0131) (6,264)			ポ	5週1700617 ラッグ&ペ			
5	7728 北間東 12,500 山日17,500円 町日10,000円 月平均 10,000円	千葉中 → 純市里 08:30 09:10 08:50 09:30 (34-0106) (7,987)	千葉中→ 城市厦 09:50 10:30 10:10 10:50 (34-9732) (6,264)	1	<u> </u>					
6	0235 北間東 12,000 山田17,500円 町日11,000円 月平均 11,000円	千葉中 → 城市重 08:30 09:10 09:50 09:30 (34-0107) (7,987)	+ M 中 → M.市 M 09:50 10:30 10:10 10:50 (34-9723) (9,024)							
7	8776 北間東 12,700 山日33,450円 村日11,000円 月平均 11,000円	千葉中 → 純市庫 08:30 09:16 08:50 09:18 (34-0108) (7,987)	千葉中→ 純市重 09:50 10:30 10:10 10:55 (34-9734) (9,024)	P 55 → MOL 12:20 14: 13:20 15: (08-9688) (3,28	50 10 2)		1			
	0344 東京 18,000 第日10,000円 月平均 10,000円	∓XΦ ⇒ 85 746 09:30 09:20 08:50 09:50 (34-3720) (14,608)	千葉中 ⇒ 紙市重 10:30 11:10 10:50 11:30 (34-3722) (14,698)	千葉中 ⇒ M.R 11:50 12: 12:10 13: (34-9723) (14,6)	5.版 千葉(20 13:44 20 14:00 09) (34-97)	# ⇒ 山五詳 14:10 14:20 (14,608)				
9	0408 東京 19,000 前日10,000円 月平均 10,000円	千葉中 中 勝 7 明 00:30 09:30 00:50 09:50 (34-3721) (14,600)	千葉中 ⇒ 城市庫 10:20 11:10 10:50 11:20 (34-9724) (14,600)	ギ第中⇒山王 11:50 12: 12:10 13: (34-9725) (14,6)	(井 千葉) 20 13:30 20 13:55 08) (34-97)	申 山五井 14:00 14:20 (14,600)				
10	4045 東京 12,200 出日96,550円 前日10,000円 月平均 10,000円	W → HGL 08:30 08:51 08:50 09:11 (30-3933) (7,491)	MGL → 度留 09:11 11:01 09:31 11:21 (38-9395) (6,390)	1-⊞I → ±0 12:36 15 13:20 17 (45-5766) (7,27	55 1500 15 1520 40 (45-57	E → 算用描 1760 1720	神計賞 → 北前東 1500 1700 1529 1720			
.,	4213 東京 12,500 山田40,200円 町日16,000円 月平均 16,000円	千葉中 → 土間東 08:30 11:55 08:50 12:15 (34-3359) (0,131)	千葉中 → 城市重 15:20 16:00 15:40 16:20 (34-9625) (10,291)			and formal	Construction (Construction			
12	5200 章政 12,600 山田 64,435円 町田 16,63円 月平均 16,855円	千葉中→ 北間東 08:30 11:55 08:50 12:15 (34-9778) (9,256)	千葉中 → 紙市重 15:20 16:00 15:40 16:20 (34-3626) (10,231)	MEL → 建 11:50 18: 12:10 14: (80-9417) (1.07	鼠 船橋 40 1115 00 1211 1) (38-94	数→ 建間 13:20 13:40 (7) (1,671)				
13	5707 東京 12,500 第日35,555円 第日10,000円 月平均 10,000円	● ¥ X 中 → hbc 04:40 08:30 04:40 08:50 (34-0001) (0)	∓ ⊈Φ → 12:40 16:25 13:20 16:45 (24-0122) (7,960)							
14	8457 東京 12,500 山日40,200円 町日11,000円 月平均 11,000円	千葉中 → 北関東 08:30 11:55 08:50 12:15 (34-9779) (8,256)	千葉中 → 紙市頁 15:20 16:00 15:40 16:20 (34-9627) (10,339)							-
-	· 新秋秋安下 運行数 119	未配車運行数 2	F-her	CIRNETIANES	10 Ph2851	-	星成和技士		HARDING BR	最終確定

図3 運行計画ドラッグ&ドロップ(複数運行)

🍘 運行計画構構处理

لارى	# CI (2018/07/25	COV.		440 1 320 1	180.0%7057至	1	政小朋友丈	MIST \$2,5532 (59(\$)		T[moo()
No	10	第1運行	第2運行	第3運行	t ()	第4運行	1865選行	第6運行	第7運行	38
1	6304 北間東 12,200 出日27,850円 町日10,000円 月平均 10,000円	▲ → MGL 08:30 08:51 08:50 09:11 (30-9583) (5,024)	+≍Ф → 03:51 13:16 10:11 13:36 (34-0125) (6,264)						-	
2	6525 北間東 12,500 山日20,100円 町日10,000円 月平均 10,000円	実筑道→ MG4 08:30 10:40 08:50 11:00 (30-9679) (9,407)	千葉中 → 11:40 15:05 12:00 15:25 (34-0130) (6,264)				第3~第5道	重行の		
9	8588 國東 12,800 出日 62,350円 期日12,000円 月平均 18,000円	千葉中 → 北間東 00:30 11:55 08:50 12:15 (34-0136) (8,256)	千葉中→風市原 15:20 16:00 15:40 16:20 (34-9731) (6,264)	MG → 3 11:50 14 12:10 14 (38-9858) (8,3	上間東 130 150 45)		017日から ミラッグ & ^	一店で ペースト		
4	5383 北間東 12,600 山田 9,750円 町日10,000円 月平均 10,000円	千葉中 → 紙市重 00:30 09:10 08:50 03:30 (34-0104) (7,987)	千葉中 → 09:50 13:15 10:10 13:35 (34-0131) (6,264)			<u> </u>				
5	7728 北関度 12,500 出日53,800円 町日10,000円 月平均 10,000円	千葉中 → 風市東 08:30 09:10 08:50 09:20 (34-0106) (7,987)	千葉中 → M.市厦 03:50 10:30 10:10 10:50 (34-9732) (6,264)	小田工 → 北 12:38 16 13:20 17 (45-9769) (7,2	開東 小田 (55 150 (15 152) 74) (45-9	意→ 武州拓 0 17:00 0 17:20 768) (7:274)	神辞度 → 之間度 15:00 17:00 15:20 17:20 (45-9758) (7.274)			
6	0235 北関東 12,000 山田17,500円 町日10,000円 月平均 10,000円	千篤中 → 紙市原 08:30 03:10 08:50 03:30 (34-0107) (7,387)	千葉中→紙市原 09:50 10:30 10:10 10:50 (34-9733) (9,024)			\wedge	Los Prate United			
7	8776 北関東 12,700 出日33,450円 終日10,000円 月平均 10,000円	千葉中 → 紙市重 08:30 03:10 08:50 03:30 (34-0108) (7,987)	千葉中→ 紙市原 09:50 10:30 10:10 10:50 (34-9724) (9,024)	P 第3 → MG 12:20 14 13:20 15 (38-5688) (3,2	(Z 150 110 88)	1 Г				
0	0344 1東京 18,900 第日10,000円 月平均 10,000円	千葉中 ゆ 師ヶ崎 08:30 09:30 00:50 09:50 (34-9720) (14,608)	千葉中 中 風赤原 10:30 11:10 10:50 11:30 (34-9722) (14,608)	千葉中 歩 ML 11:50 12 12:10 13 (34-9723) (14,	市蔵 千葉 130 13: 120 14: 608) (34-9					
9	0408 東京 18,800 第日10,000円 月平均 10,000円	千葉中 中 勝ヶ崎 00:30 03:30 00:50 03:50 (34-9721)(14,608)	千葉中 ⇒ 風市原 10:30 11:10 10:50 11:30 (34-9724) (14,608)	千葉中 歩山 11:50 12 12:10 13 (34-9725) (14,	五井 千郎 (20 13: (20 13: (34-9) (34-9)					
10	40.45 東京 12,200 出日50,250円 町日10,000円 月平均 10,000円	W → MGL 08:30 08:51 00:50 09:11 (30-9839) (7,491)	MiL → 度間 09:11 11:01 09:31 11:21 (38-9395) (6,390)							
11	4213 東京 12,500 出日40,200円 単日10,000円 月平均 10,000円	千葉中 → 北関東 00:30 11:55 00:50 12:15 (34-9259) (8,131)	千葉中→ M.市原 15:20 16:00 15:40 16:20 (34-9625) (10,291)							
12	5200 東京 12,600 出日64,450円 月平均 10,000円	千葉中 → 北間東 00:30 11:55 00:50 12:15 (34-9778) (0,256)	千葉中→ M.市庫 15:20 16:00 15:40 16:20 (34-9626) (10,291)	HGL → 1 11:50 13 12:10 14 (38-9417) (1.0	25日 相同報 140 111: 100 12: 71) (18-5	資 → 度冒 1 13:20 1 13:40 1(7) (1,671)				
13	5707 東京 12,500 山日35,533円 町日10,000円 月平均 10,000円	● 千葉中 → RDC 04:40 08:30 04:40 08:50 (34-0001) (0)	千篇中 → 12:40 16:25 13:20 16:45 (34-0122) (7,968)							
14	6457 東京 12,500 出日40,200円 时日10,000円 月平均 18,000円	千葉中 → 北関東 08:30 11:55 08:50 12:15 (34-9779) (8,256)	千葉中 → M.市度 15:20 16:00 15:40 16:20 (34-9627) (10,339)							
2.2										
								CRIME ROOM		
	霸状態支更 運行数 119	未配車運行数 2	データ修正	(重新情報) 休車状	】 【棚 陸送トレ	-5-	ISSN®後常 检索	千葉 四田勝	大阪・堺	最終確定



ー般シール受注入力業務の Delphi/400化

大阪シーリング印刷株式会社 IT 推進部 情報システム課 主査



部門 2

大阪シーリング印刷株式会社 http://www.osp.co.jp/

1927年創業以来、加工業としての 本業に徹した堅実経営を買き、主に 凸版印刷を中心とした原紙製造から 印刷までの一貫生産工程を軸に、全 国をオンラインで結ぶ営業・生産 ネットワークを活用。シール業界の リーディングカンパニーとして、印 刷の枠を越えた総合パッケージメー カーとして事業を展開している。

業務課題

従来、一般シールの受注入力を PC の 5250 画面で行っていたが、画面項目が 約 150 まで増加したため、以下のような 問題により入力が困難となった。【図 1】

- ・タブが使えないので、すべての項目を
 1 画面で入力する必要がある。
- ・画面項目を長い間追加してきた経緯により、項目の並びが業務の流れに沿った順番になっていない。
- ・画面上に説明が少ないため、慣れた ユーザーしか入力ができない。
- ・連続入力時に5250セッションのマクロ機能を使用しているが、マクロ入力がずれた項目に登録され、画面がかたまる問題が指摘されていた。

そこで、GUI 化による問題解決に向 けて、Delphi/400 で受注入力画面を作 成・開発した。

技術課題

入力者が従来の画面に慣れているの で、Delphi/400 画面にも同様の動きを 組み込むことが課題となった。

具体的には、以下の3つの要望があった。

- (a) 右 Ctrl キーによる実行動作
- (b) タブによる複数画面の項目展開
- (c) コピー新規登録、ひな形からの入力

また今回の GUI 化をきっかけに、以下の課題にも取り組んだ。

- (d)入力者ごとのデータ表示
- (e) 難しい項目や専門用語などに対する ヘルプ表示

技術課題の解決策

物理ファイルの構成を QTEMP では なくメンバーを使用するようにしたうえ で、上記の課題に対し、以下のように解 決した。

(a) 右 Ctrl キーによる実行動作
 Tform のイベントのキーボード操作
 に VK_CONTROL (仮想キー) を組み
 込み、5250 セッションによる右コントー
 ルキーを有効にした。【図 2】

(b) タブによる複数画面の項目展開
 TPageControl を配置し、複数タブの
 入力項目を配置するように対応した。

(c) コピー新規登録、ひな形からの入力 コピー機能はメンバー指定からの データのみを使い、そのままフィールド をコピーし、その値を埋める形で実現さ せた。

(d) 入力者ごとのデータ表示
 メンバーに保存したデータを使用し
 て、入力者ごとにデータを見られるよう
 にした。具体的には、TTable コンポー
 ネントでメンバー指定を行った。【図 3】

 	従来の5250画面 ***********************************	
	を またいましたいでは、 () () () () () () () () () () () () ()	
図3	end; 3 物理ファイルのメンバー指定 // <i>LBDL1F (入力中データ)</i> Tb1LBDL1F.TableName := cWorkLib + 'LBDL1F(' + FileMember + ')'; // WriteLeg('Tb11EDL1F TableName');	

(e)難しい項目や専門用語などに対する		
ヘルプ表示		
 や専門用語をクリックして表示できるよ		
 うにした。		
業教理 顕叙法 レ 効 甲		
未物味趣肝人て別木		
元成した回面は、未務の流れに行うて タブごとに入力項目が整理されたこと		
で、1件当たりの入力時間が5分から2 分配へと、土幅に工物を削減できた		
またヘルプ画面を表示させることに より、入力方法に関する知識をもたたく		
ても入力できる画面を実現した。【図5】		
 今後も機能を追加し、より効率的に受 注業務が行えるように進めていきたい。		
 M		

T R	してIPTIT/ すいでくみる文/エハンビー単直 (#1114## (+#)-4/1-98/21+)
1	A2004 受法入力中 2018/00/12 (相違) 18:38 音 き a Ver 2.1 (1975年7月) - 加受注入力画面 Ver 2.1 (1975年7月) - 100 (1975711) - 100 (19757111) - 100 (19757111) - 100 (197571111) - 100 (1975711111) - 100 (197571111111111111111111111111111111111
ĺ	日本の2015年間のシージャンブル(出荷指示)製版(日前・加工(仕上(折)) 相包(特記) 第25 小 基本 一 受注(サンブル(出荷指示)製版(日前・加工(仕上(折)) 相包(特記) 第55 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
	2010 100 199502 社用 (外注・受注管理部) - ● 株衆 100 検索 7 特に運送後1 禁止運送後1 禁止運送後1 禁止運送後2 禁止運送後3
ļ	BSRe 産産((ロル) コピー(新現(F)) ダータ(単存(TS)) ダータ(単存(TS)) (水へ(T/T)) 水へ(T/T) 水へ(T/T) (ホー(T/T)))
1	
)	ヘルプをクリックして詳細なヘルプ画面を呼び出し
101	【使注葉濃葉稿 (一谷シール・8 F) ■/(一ジョン2.1 ■) LM2304 蒸発 (- 古) - カーカー
	受け 一般シール受注入力画面 Ver 2.1 青色下線の表示は画面をクリックするとヘルプ画面が表示されます
E	版区分 基本 受注 サンブル 出荷指示 製版 印刷・加工 仕上 (巻) 梱包 特記 美口 (またっゴ) 20/06/15 時期 20/06/14 売注教員 5,000 会計教員 5,000 ほう・ブル 会員
	■日本金平3.57 50/50/13 新州 50/50/14 文元数量 3,500 日計数量 3,500 万クラル日言 出荷指示1 出荷指示2
	5.2% # 得意先 納入先出し人 数量 受注単位 着日 配連区分 同梱 行単位備考欄 送り状記載事項 納品先名 5.5% 000 fb tr 5.000 f
	C F5961 B88 D 100 1 枚 F F5961 B88 D 100 1 枚 F
	※着日欄がブランクの場合、工場一任とします 確認 ※「厚る」ボタンを押すと、指定・禁止運送使が知識値に厚ります。 スペープ ※サンプルタブと連接した明顯の数量は変更できません。
L	サンプルタブに買って確正してください。
12	
	LAN2304 出荷指示(複数納入先)の入力の仕方 2018/08/23 (木曜日) 12:32
	画面構成 基本 サンブルの場合 同梱指示の場合 運送便の指定方法 入力チェック内容
	2
	版区分 基本 受注 出荷指示 製版 印刷・加工 仕上 (巻) 福己 特記
	第日18年キメファ 23/12/31 新用 23/12/21 (東注版業 20,000) 谷計版業 20,000 为275 待要先前入先出し人 数量 受注単位 着日 配適区分 同期名学期の価単態 送りは70割実施
	② ② ② ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ⑤ ⑦ ⑤ ⑤ ⑤ ⑦ ⑤ ⑦ ⑦ ⑦ ⑦ ⑦ ⑤ ⑦
	→ ₩ 55202 004 5 100 1.42 → 29/12/31 →
	2 サンブル別連頭送 ※着日欄がブランクの1
	マロークワンの別部別送 ※着日欄がフランクの* <入力方法(同梱)> ※1 同梱区分は 1~5、までの値を入れて1対としてください 1. 同梱区分を入力します ※2 製品とサンプルを同梱したい場合はサンプルのチェック 2. 他の項目を入力 と同梱区分を入れてください ※3 同梱の場合は、数量・5年単の供告を増加的は合うとうに
	ア サンブル別期開送 ※着目機がブランクの* く入力方法(同梱)> ※1 同梱区分は 1~5、までの値を入れて1対としてください 1. 同梱区分を入力します ※2 製品とサンプルを同梱したい場合はサンプルのチェック 2. 他の項目を入力 ※3 同梱の場合は、数量・行単位備考欄以外は合うように 指定してください

Migaro.Technical Award 2018 優秀賞

Delphi/400による無線ハンディ ターミナルのデータ集約の仕組みの実装

寺西 健一 様

大阪シーリング印刷株式会社 IT 推進部 情報システム課 主査



部門 2

大阪シーリング印刷株式会社 http://www.osp.co.jp/

1927年創業以来、加工業としての 本業に徹した堅実経営を買き、主に 凸版印刷を中心とした原紙製造から 印刷までの一貫生産工程を軸に、全 国をオンラインで結ぶ営業・生産 ネットワークを活用。シール業界の リーディングカンパニーとして、印 刷の枠を越えた総合パッケージメー カーとして事業を展開している。

業務課題

印刷工場において、印刷に用いる「版」 の入庫を確認し、版管理データとして IBM i 基幹システムに登録している。

従来の登録業務は、工場内の各作業場 所でハンディターミナル (バッチ式) に よるデータ読み取り・蓄積の後、ハンディ ターミナルを PC に設置した置台(通信 ユニット) に置く、という2段階でデータ 転送していたため、手間がかかっていた。

そこで、各作業場所から直接データの 収集・登録を行うために、バッチ式に変 えて無線ハンディターミナルを導入する こととした。

技術課題

従来は1台の PC だけで実装された仕 組みだったので、実績データのファイル 名を固定させて FTP による通信を行っ ていた。しかし無線による通信に変わっ たため、任意のタイミングでデータを送 信する仕組みの実装が必要となった。 具体的には、以下の3点を仕様とし て組み込む必要があった。

- (a) IBM i へのデータ送信を定期的に自 動処理する(注1)
- (b) ハンディターミナルからの実績デー タを1つにまとめて IBM i に送信
- (c) ハンディターミナルからのファイル 名競合の回避(注2)

(注1) ハンディターミナルのデータ は、中継用のサーバー区画(シンクライ アント)を経由してIBM iに登録し、サー バー区画上で Delphi/400 アプリケー ション(待ち受けアプリ)を起動する仕 組みとした。中継用に、PCの代わりに シンクライアント区画を利用することに より、セキュリティ面と環境管理に関し ても考慮した。【図1】

(注2) 複数のハンディターミナルから 待ち受けアプリに同時にデータを送信す る場合、Delphi/400の待ち受けアプリ 側で受信ファイル名が競合して、処理が ロックする問題があった。ハンディター ミナル側でも待ち状態になり、ロックが 解除できなくなる。

技術課題の解決策

前述の技術課題に対して、Delphi/400 の待ち受けアプリにて、次のように解決 した。

(a) IBM i へのデータ送信を定期的に
 反復処理

ハンディから実績データを IBM i へ 送信する頻度を画面上で設定可能とし た。具体的には、Timer を実行させる 時間間隔を指定できるようにした。 【図 2】

 (b) ハンディターミナルからの実績デー タを1つにまとめて IBM i に送信 Delphi アプリケーションでは、ハン ディから FTP 送信されたフォルダを、 指定された時間間隔で検索(ファイル*. txt) し、ファイルがあれば、1つのテ



キストデータにまとめる。まとめたデー		
タは Ttable コンポーネントを経由して、		
IBM i へ送信する。【図 3】		
(c) ハンディターミナルからのファイル		
 名競合の回避 ハンディターミナル側のアプリケー		
ションでは、待ち受けアプリに送信する		
実績データを、 端末 ID +年月日.txt」 という形式にして可変のファイル名とす		
ることで、同一ファイル名が競合するこ トを回避した		
こで回避した。		
業務課題解決と効果		
従来、置台式にて転送処理を行ってい		
るたびに転送を設定する必要があった		
が、無線化により環境整備の手間がなく なった。		
また、現場の担当者からも置台まで行		
て動作が省略できてよび、この計画が奇 せられた。具体的には、1日当たり30		
分/人の工数短縮という大きな効果を挙 げることができた。		
M		

110	□procedure IFormI.limerZlimer(Sender: IUbject);
112	sICSV: TstringList;
	slLine: TStringList;
	sIGCSV: TstringList;
-	i: Integer;
	SR: ISearchRec;
	/// ND · INLEGET ,
	begin FindFirst命令により、* txtの内容のファイルを
120	sICSV :=TStringList.Create; すべて読みながら、一つのファイルへ集約
	sicist :=TStringList Create:
	sidoor lottimgerst.oreate,
127	<pre>if FindFirst('C:\FtpTrans\LocalUser\FtpSiage*.txt',FaAnyFile,SR) =0 then begin</pre>
	repeat
130	//datからの読み込み
	sICSV.LoadFromFile('C:¥FtpTrans¥LocalUser¥FtpSiage¥'+ SR.Name); //
(34.) 	except
-	begin
	Abort; // サイレント例外
	end; end;
140	//レコードの行数分(Count-1)処理を繰り返す。
140	begin
	slLine.CommaText:=(slCSV[i]);
÷.	end:
2	//ファイルを保存(Basho.txt) // ShowMessage('Basho.txtファイルを保管'):
1.0	sIGCSV.SaveToFile('c:¥Siage¥Siage.txt');
1	//処理した件数を回面へ表示 Labe19.Caption:=IntToStr(sIGCSV.Count):
100	//処理した時刻を画面へ表示
100	stcTockZ.Caption := FormatUateTime(yyyy/mm/dd (aaaa) nn:nn:ss ,Now()); //ファイルがなくなったら探すのを止める。
1	until FindNext(SR) <> 0;
- G	end;
	9へし記み終わるとUIOSE町市により用しる
	try $(c c v = - v + t = - t)$
	AssignFile(CSVFile, 'C:¥Siage¥Siage.txt');
	Reset(CSVFile);
	try while (eof(CSVEile) = false) do
	begin
270	// CSVの読み込み
-	readLn(USVFIIe, str); // USVの1行を読み込む st.CommaText := str: // 女空別を公判する
3	Number := StrToIntDef(st.Strings[1],0);
*	case Number of
	// 位上げナエックの場合 1: begin
	//文字列をテーブルに書き込む
*	//Table1. Append: //レコードの追加
280	」:= 0; TblSGCHKF1.Insert: // レコードの挿入
	//for~to~doは初めのパラメータが2つ目のパラメータの
	//値になるまで実行。
	//この場合、ファイルの件数ガ1able1(15)にアータを //追加する。
-	for j := 0 To st.Count-1 do
÷	TblSGCHKF1.Fields[j].AsString := st.Strings[j];
	まとめたファイルは Ttableコンボーネント
	栓田で10510万でアータを直接書き出し



Migaro. Technical Report 2018

ミガロ.SE 論文/ミガロ. テクニカルレポート

株式会社ミガロ.

システム事業部 システム2課

[Delphi/400] OLEを利用したExcel出力の パフォーマンス向上手法

1. はじめに

- 2. OLE を利用した基本的な Excel 出力
- 3. Excel の出力パフォーマンス
- 4. OLE バリアント配列を利用した実装
- 5. 最後に



1985 年 11 月 22 日生まれ 2008 年 3 月 阪南大学 流通学部卒 業 2008 年 4 月 株式会社ミガロ. 入社 2008 年 4 月 システム事業部配属

現在の仕事内容:

Delphi/400を利用したシステムの 受託開発を担当し、基本設計から納 品・フォロー、保守作業に至るまで、 システム開発全般に携わっている。

1.はじめに

Delphi/400 にバンドルされる帳票 ツールを利用して帳票機能を開発するこ とが多いが、Delphi/400 のバージョン アップによって帳票ツールが変更された 場合、帳票機能を移行または作り直しの 検討が必要となる (バンドルされる帳票 ツールは、Delphi/400 Version 5 ~ 7 が「QuickReport」、7 ~ XE が「Rave Reports」、XE 3 以降は「FastReport」 である)。

このような帳票ツールに依存した変 更を解決する方法の1つとして、 OLE(*)を利用しExcelをベースに した帳票機能を実装することもできる。

OLE での Excel 利用は、Delphi/400 から比較的簡単に実現できるが、大量 データの処理には向いておらず、パ フォーマンスが落ちることもある。この 課題についてはプログラムロジックを工 夫することで解決が可能である。本稿で は、OLE を利用した基本的な Excel 出 力方法から、大量データを出力する場合 のパフォーマンス向上手法を解説する。

2.0LEを利用した 基本的なExcel出力

本章では、Excel 出力プログラムの作 成例を題材に、OLE の基本的な操作方 法について解説する。

Excel 出力プログラムの概要

- IBM iより売上情報を取得し、明細を Excel に出力する。
- ・明細出力時、各営業所単位で合計金額
 を出力する。
- Excel 出力後に保存ダイアログを表示 し保存する。

開発環境:Delphi/400 10.2 Tokyo および dbExpress

なお本稿では、帳票テンプレートは事 前に Excel で作成したものを利用する。 Delphi/400 から罫線や書式設定の操作 を行うことは可能だが、その回数が多く なるほどパフォーマンスの低下に繋が る。そのため、本稿ではあらかじめ Excel テンプレート【図1】を作成し、 Delphi/400 からの操作回数を極力減ら している。また Excel でテンプレート を作成することで、帳票項目の書式設定 も Excel 側に持たせることができるた め、そうした書式変換のプログラミング も不要となる。

【図 1】の Excel テンプレートを利用 した Excel 出力のロジックが【ソース 1】、【ソース 2】となる。

*OLE: Object Linking and Embedding の略称

マイクロソフトが提供する機能の1 つで、複数のアプリケーション間でデー タの転送や共有を行うための仕組みを指 す。Excel は OLE サーバーとなり、他 のアプリケーションから操作可能(操作 のためのメソッドが用意されている)。 実行のためには、Excel が導入されてい る環境が前提となる。

図1 Excel テンプレート(例)



ソース1

procedure TForm1. Button1Click(Sender: TObject); var ovExcel: OleVariant; ovWorkBook: OleVariant; ovWorkSheet: OleVariant; sFileName: String: iRow: Integer; sFL1: String; cShokei: Currency; begin ^{egin} //出力データ取得 ...① SQLQuery1.Close; SQLQuery1.SQL.Text := 'SELECT * FROM TRURI ORDER BY URIFL1'; SQLQuery1. Open: try // Excelオブジェクト生成 ----② ovExcel := CreateOleObject('Excel.Application'); try // Excel非表示 ...3 ovExcel.Visible := False; // フォーマット読み込み④ sFileName := IncludeTrailingPathDelimiter(ExtractFileDir(Application.ExeName)) + 'Format.xlsx'; ovWorkBook := ovExcel.Workbooks.Open(sFileName); ovWorkSheet := ovWorkBook.WorkSheets[1]: // 出力準備 iRow := 3; cShokei := 0; sFL1 := SQLQuery1.FieldByName('URIFL1').AsString; // 小計出力用営業所 // データ出力 while not SQLQuery1. Eof do begin agin // セルヘ転送5 ovWorkSheet.Cells[iRow, 1].Value := SQLQuery1.FieldByName('URIFL2').AsString: //名称 // 石桥 // 住所1 // 住所2 // 電話番号 // FAX番号 ovWorkSheet.Cells[iRow, 1], Value := SQLQuery1.FieldByName('URIFL2').AsString: ovWorkSheet.Cells[iRow, 13].Value := SQLQuery1.FieldByName('URIFL3').AsString; ovWorkSheet.Cells[iRow, 13].Value := SQLQuery1.FieldByName('URIFL4').AsString; ovWorkSheet.Cells[iRow, 21].Value := SQLQuery1.FieldByName('URIFL5').AsString; ovWorkSheet.Cells[iRow, 21].Value := SQLQuery1.FieldByName('URIFL6').AsString; ovWorkSheet.Cells[iRow, 24].Value := SQLQuery1.FieldByName('URIFL7').AsCurrency: // 金額 // 小計を計算 cShokei := cShokei + SQLQuery1.FieldByName('URIFL7').AsCurrency:

①出力データ取得

TSQLQueryを利用してIBM iのファ イルより Excel 出力用のデータを取得 する。

② Excel オブジェクト生成

CreateOleObject の 引 数 に「Excel. Application」を指定し、Excel のオブ ジェクトを生成する。また、変数 ovExcel に代入することで生成したオブ ジェクトを OleVariant 型で操作可能に する。前提として、Delphi/400 で OLE の各メソッドを利用するには uses 節に 「ComObj」を追加する必要がある。

③ Excel 非表示

Excel を表示したままにすると、プロ グラムの Excel 操作が画面ですべて表 示されてしまいパフォーマンスも低下す るため、ovExcel.Visible を False にし て非表示にする。

④フォーマット読込

Excel テンプレートのパスとファイル 名を取得(変数sFileName)し、 Workbooks.Openで開いたブックを変 数 ovWorkBookに代入する。 ovWorkBookのWorkSheetsプロパティ でブックのシート番号を指定し、シート オブジェクトを取得する。これで、①と 同様に、ブックとそのシートを OleVariant型で操作可能にする。

⑤セルヘ転送

ここで Excel のセルへ値を出力する。 ovWorkSheet の Cells プロパティで セル位置(行および列のインデックス) を指定し、IBM i のデータベース・ファ イルより取得した値を代入する。

代入した値はセル側に設定されてい る書式が適用される。

⑥ Excel 保存

TSaveDialogを配置して、Excel 出 力後に保存ダイアログを開く。そのダイ アログで指定したパスを SaveAs メソッ ドの引数に渡すことで任意の場所にファ イルを保存できる。また Filter プロパ ティの設定により、保存時に選択できる 拡張子の制御が可能である。本稿では 「*.xlsx,*.xls」の2種類を指定可能にし ている。【図 2】 ファイル保存時の注意点を補足して おく。

ダイアログ上で「.xls」を指定して保 存した場合、保存したファイルを開く際 に拡張子が正しくない旨の警告が表示さ れる。【図 3】

これは、Office のバージョン 2007 以 降、規定のデータ保存フォーマットが変 わり、過去の Office とは互換性のない 形式で保存されることになったためであ る。対応方法としては、拡張子に「.xls」 が指定された場合、SaveAs メソッドの 引数にファイル形式を表す定数「56」 (Excel 97-2003 ブック)を指定するこ とで後方互換に対応可能となる。これに より保存したファイルは、開いた際に警 告が表示されなくなる。この定数はほか にも PDF や CSV 形式で出力可能であ り、その一覧は以下の Web サイトにて 紹介されているので、参考にしてほしい。

●参考 URL:

https://msdn.microsoft.com/ja-jp/ vba/excel-vba/articles/xlfileformatenumeration-excel

(Google で "XlFileFormat 列挙"を 検索すると上位に表示される。)

⑦ Excel 表示

処理終了後に ovExcel.Visible を True にして Excel を表示する。データ出力 が完了した状態で Excel が表示される。

⑧オブジェクト解放
 生成したOleVariant型の変数を
 Unassignedで解放する。

これで Excel 出力処理は完成である。 このプログラムで実際に Excel での帳 票出力を行った結果が【図 4】である。

ソースを見ればわかるとおり、OLE を利用した Excel の出力自体は比較的 簡単に実装できる。しかし、この方法で は、大量のデータを出力する際にパ フォーマンスがかなり悪くなってしま う。その原因としては、Delphi/400 か らセルに値をセットする際にアプリケー ション間で通信が発生する(以下、通信 と表記)からだが、実はこれが処理時間 が長くなる大きな要因となっている。

本稿のテンプレートを例にすれば、1 明細あたり6項目存在するため、1行出 力する度に6回の通信が発生する。これ が数百、数千件と、扱うデータ件数や項 目数が増えると、その分通信が繰り返さ れるため、パフォーマンスに影響するの は明白である。パフォーマンスを向上さ せるには、いかに Delphi/400 と Excel との通信回数を減らすかが重要である。 その手法については次章で触れる。

3.Excelの 出力パフォーマンス

2章では基本的な Excel 出力の手法を 説明した。本章ではパフォーマンスを向 上させる手法、つまり Excel との通信 回数を低減する方法を紹介する。

Excel に値を出力する際、各項目を1 セルずつ出力するのではなく、出力する 値を2次元配列などに記憶させ、特定の タイミング(改ページ時など)で一括出 力することにより、通信回数を格段に低 減させることができる。

その方法は2種類あり、クリップボー ド、もしくは OLE バリアント配列を利 用することで実現できる。

しかし、前者のクリップボードについ ては以下の課題点がある。

●クリップボードを利用する上での課題点

- ①クリップボードの内容がプログラムで 書き換えられてしまう
- WindowsVista 以降、クリップボードの動作が不安定

①の課題点

クリップボードは、列ごとに「#9」(タ ブコード)、改行ごとに「#13#10」(改 行コード)のリテラルを挿入することで、 複数行・列の内容を格納でき、またその 内容を Excel に一括で出力できる。し かし、この手法はユーザーの意図しない ところでアプリケーション側からクリッ プボードの内容が書き換えられてしま い、ユーザーのコピー&ペースト操作な どに影響を与えてしまうことがある。

②の課題点

Windows では、Excel 上でコピー& ペーストを繰り返し実施しているとク リップボードのエラーが発生するという 事象がある。

発生する条件は不定だが、特に
```
ソース2
                    // 次データへ
SQLQuery1.Next:
                     // 行数をインクリメント
                    Inc(iRow):
// 営業所が変更されたタイミングまたは最終行で小計を出力
if (sFL1 ◇ SQLQuery1.FieldByName('URIFL1').AsString) or (SQLQuery1.Eof) then
                    begin
                       // 小計を出力
                      //小計を出力
ovWorkSheet.Cells[iRow.20].Value := sFL1 + '小計:';
ovWorkSheet.Cells[iRow, 24].Value := cShokei;
// 営業所を保持
sFL1 := SQLOuery1.FieldByName('URIFL1').AsString:
                       // 小計をクリア
                       cShokei := 0;
// 行数をインクリメント
Inc(iRow);
                    end:
                  end:
                  // Excel保存 ....⑥
if SaveDialog1. Execute then
                 begin
                    if (LowerCase(ExtractFileExt(SaveDialog1.FileName)) = '.xls') then
                    begin
                      ovWorkBook. SaveAs (SaveDialog1. FileName, 56);
                    end
                    else
                    begin
                      ovWorkBook. SaveAs (SaveDialog1. FileName) :
                    end;
                    ovWorkBook. Saved := True;
                 end:
                 // Excel表示 ...⑦
ovExcel.Visible := True;
               inally
//オプジェクト開放 ....⑧
cvWorkSheet := Unassigned:
cvWorkBook := Unassigned:
                 ovExcel := Unassigned;
            end:
finally
               SQLQuery1.Close:
            end:
          end:
図2 TSaveDialog設定
```

つ検索			ř	
プロパティ イベン	N		フィルタの設定	
CtI3D DefaultExt FileName	True True	Î	フィルタ名 Excel ブック (*.xisx) Excel 97-2003 ブック (*.xis)	フィルタ *.xlsx *.xls
Filter) *.xlsx Excel 97-2003 ブック (*.xls) *	.xls	-	
FilterIndex	1			
HelpContext InitialDir	0			
LiveBinding デサ	「イLiveBinding デザイナ			
Name	SaveDialog1			
Options	[ofHideReadOnly,ofEnableSizing]		OK	キャンセル ヘルプ
ofReadOnly	False			
ofOverwritePr	or 🔲 False			
ofHideReadOr	nly 🗹 True			
ofNoChangeD	Dir 🗌 False			
ofShowHelp	False			
ALV 211				

Windows10 では動作が安定しないこと が多い。現時点 (2018 年 8 月現在) では、 マイクロソフトより解決方法は明示され ておらず、Windows Update による修 正も実施されていない。

Delphi/400からクリップボードを操 作する際もこの影響を受ける可能性があ るため、本稿ではクリップボードの利用、 解説は割愛する。

後者のOLEバリアント配列は、2次 元のバリアント配列を生成し、配列に順 番に値をセットすることで、複数列・行 の情報を一括でExcelに出力する手法 である。

この手法であれば、クリップボードの 課題点の影響を受けずにパフォーマンス 向上を実現することができるため、本稿 では OLE バリアント配列を利用した Excel 出力の方法を解説する。

4.OLEバリアント配列 を利用した実装

本章では、実際に OLE バリアント配 列を利用したプログラムの実装例とし て、2章のソースとの相違点を中心に解 説する。(【ソース 3】、【ソース 4】)

2次元配列用 OLE バリアント変数を 定義

変数 ovArray を OleVariant 型で宣 言する。

②明細転送用の配列を準備

VarArrayCreate 関数で、変数 ovArrayに明細転送用の配列を設定す る。1番目の引数は配列の要素(行およ び列のインデックス)を指定する。2番 目の引数は配列の要素型(varVariant) を指定する。ここではExcelテンプレー ト1ページ分(行:24、列:25)で定 義している。

③配列へ格納

ここではExcelのセルに直接値をセッ トするのではなく、②で準備した配列に 格納する。変数 ovArray に行および列 のインデックスを指定(Excelのセル位 置に該当する箇所)し、IBM iのデータ ベースより取得した値を順番に配列へ格 納する。

④配列より Excel に転送

ovWorkSheet の Range プロパティ で配列の行・列の数に合わせて Excel のセル範囲を指定し、変数 ovArray を 代入することで、③で配列に格納した値 を一括で出力することができる。これに より、複数行・列の出力を1回の通信で 完了させることが可能である。

通常の静的配列では Range で指定し た範囲に配列を代入することができない (型違いでコンパイルエラーとなる)。静 的配列でも1行単位であれば出力可能だ が、その場合は Excel との通信回数が 多くなってしまう。そのため、配列は必 ず OleVariant 型で定義する必要がある。

⑤配列を解放

VarClear で生成した OLE バリアン ト配列 (ovArray) を解放する。

これで、OLE バリアント配列を利用 した Excel 出力処理が完成した。出力 結果としては【図 4】と同じになる。

本章の①~⑤で解説した内容が、2章 のプログラムと異なる点である。別途、 改ページが必要になる場合はテンプレー トのシートを1ページ分コピーして最終 行からペーストを行うか、もしくはあら かじめ2ページ目以降をテンプレート内 に作成しておくことで対応が可能であ る。

また、配列に格納するデータ量に応じ てメモリを消費するので、大量に格納し てメモリ不足に陥らないためにも、一定 のタイミングで出力するよう注意した い。

続いて、本章で作成したプログラムが どの程度パフォーマンスを向上させられ たかを検証するため、実際に Excel 出 力にかかる時間を計測、比較している。

検証用にデータを 2000 件準備し、そ れに伴いロジックを一部変更した(【ソー ス 3】の②で定義している配列の要素を 行: 2000 に変更)。

そして、2章と本章のプログラム共に Excel オブジェクト生成からオブジェク ト解放まで(*ソース内コメント参照) にかかる時間を計測した結果が【図5】 である。

「セル単位で転送」ボタンの右側に、 第2章のセル単位に出力した場合の計測 値、「配列で一括転送」ボタンの右側には、 本章の OLE バリアント配列を利用して 出力した場合の計測値を表示している。

<実行結果> 「セル単位で転送」 = 7.866 秒 「配列利用」 = 0.964 秒

結果の差からわかるとおり、Excel と の通信回数を減らすことで処理時間に明 確な効果が出ている。

2章のセル単位に出力する方法では、 単純計算で明細6項目×2000行で 12,000回もExcelと通信を行う。これ に対して、本章で解説したOLEバリア ント配列を利用したプログラムならば通 信回数が1回で済むため、パフォーマン スの向上にかなり貢献していることがわ かる。

1回の通信にかかる時間は微々たるも のだが、扱うデータの件数や項目数が増 えれば増えるほど、この差は顕著になっ てくるので、本章のテクニックが有効と なる。

5.最後に

本稿では、OLE を利用した基本的な Excel 出力の方法と、Excel 出力のパ フォーマンスを低下させる要因として、 Excel 操作に伴って発生する通信がある ことを解説した。

OLE バリアント配列を利用する場合 としない場合の計測値を比較すれば、 Excel との通信回数の低減がパフォーマ ンス向上に役立つことがご理解いただけ たと思う。これは、Excel を操作するプ ログラム全般で有効なテクニックであ る。今後、帳票機能を OLE で実装する 際は、本稿で解説したパフォーマンス向 上テクニックを役立てていただきたく思 う。

Μ

	Microsoft	t Excel			
は(火) (火)支(1) ヘルプ(1)		間こうとしているファイル 'asaxis' は、実際にはその拡張子が行	ミすファイル形式ではありません。このファイルを聞く前し	こ、ファイルが破損していないこと、信頼できる発行元からのファ	(ルであることを確認してください。ファイルを今すく聞きます
The second se			(\$1.12) (11.12	(N) ~117(H)	
2016年8月19月12月25日、10月11日		この情報部は役に立ちましたか?			

図4 出力結果

	売」	上一覧表(2018年)			
名称	住所1	住所2	電話番号	FAX番号	金額
株式会社足立商店	東京都足立区	1 - 1 - 2	XXX-XXXX-XXXX	XXX-XXXX-XXXX	1,000
株式会社足立興業	東京都足立区	1 - 2 - 3	XXX-XXXX-XXXX	XXX-XXXX-XXXX	2,000
株式会社荒川商店	東京都荒川区	2 - 2 - 3	XXX-XXXX-XXXX	XXX-XXXX-XXX	3,000
株式会社荒川興業	東京都荒川区	1 - 2 - 1	XXX-XXXX-XXX	XXX-XXXX-XXX	5,000
株式会社板橋商店	東京都板橋区	5 - 2 - 3	XXX-XXXX-XXXX	XXX-XXXX-XXXX	10,000
			東京	営業所 小計:	21,000
株式会社池田商店	大阪府池田市	1 - 2 - 3	XXX-XXXX-XXXX	XXX-XXXX-XXXX	1,000
株式会社泉大津商店	大阪府泉大津市	12-2-3	XXX-XXXX-XXXX	XXX-XXXX-XXX	2,000
株式会社県佐野商店	大阪府泉佐野市	13-2-3	XXX-XXXX-XXXX	XXX-XXXX-XXXX	3,000
株式会社岸和田裔店	大阪府岸和田市	14-2-1	XXX-XXXX-XXXX	XXX-XXXX-XXX	4,000
			大阪	営業所 小計:	10,000
八幡商店株式会社	京都府八幡市	1 - 5	XXX-XXXX-XXXX	XXX-XXXX-XXXX	3,600
株式会社亀岡	京都府亀岡市	5 - 2 - 3	XXX-XXXX-XXXX	XXX-XXXX-XXXX	5,500
			京都	営業所 小計:	9,100
株式会社広島商店	広島県広島市	1 - 1 - 1	xxx-xxxx-xxxx	XXX-XXXX-XXXX	1,000
株式会社福山商店	広島県福山市	1 - 2 - 2	XXX-XXXX-XXXX	XXX-XXXX-XXXX	3,000
株式会社尾道商店	広島県尾道市	3 - 1	XXX-XXXX-XXXX	XXX-XXXX-XXXX	2,500
			広島	営業所 小計:	6,500
株式会社津軽商店	青森県津軽市	1 - 2 - 2	XXX-XXXX-XXXX	XXX-XXXX-XXXX	1,500
株式会社北津醛商店	青森県きた津軽市	1 - 3 - 2	XXX-XXXX-XXXX	XXX-XXXX-XXXX	3,300
			青窈	営業所 小計:	4,800

ソース3 procedure TForm1.Button2Click(Sender: TObject); var ~途中省略~(ソース1と同様) begin // 出力データ取得 SQLQuery1. Close; SQLQuery1. Open: try // Excel非表示 // 出力準備 iRow := 0: cShokei := 0; // データ出力 begin // 小計を計算

ovArray: OleVariant: ...(1) SQLQuery1.SQL.Text := 'SELECT * FROM TRURI ORDER BY URIFL1': try // Excelオブジェクト生成 ovExcel := CreateOleObject('Excel.Application'): ovExcel.Visible := False: // 明細転送用の配列を準備 ...② ovArray := VarArrayCreate([0, 23, 0, 24], varVariant); // フォーマット読み込み SFileName := IncludeTrailingPathDelimiter(ExtractFileDir(Application.ExeName)) + 'Format.xlsx': ovWorkBook := ovExcel.Workbooks.Open(sFileName): ovWorkSheet := ovWorkBook.WorkSheets[1]: sFL1 := SQLQuery1.FieldByName('URIFL1').AsString: // 小計出力用営業所 while not SQLQuery1. Eof do egin // 一旦配列に格納 ...③ ovArray[iRow, 0] := SOLQuery1.FieldByName('URIFL2').AsString: // 名称 ovArray[iRow, 7] := SOLQuery1.FieldByName('URIFL3').AsString: // 住所 1 ovArray[iRow, 12] := SOLQuery1.FieldByName('URIFL4').AsString: // 住所 2 ovArray[iRow, 17] := SOLQuery1.FieldByName('URIFL5').AsString: // 電話番号 ovArray[iRow, 20] := SOLQuery1.FieldByName('URIFL6').AsString: // FAX番号 ovArray[iRow, 23] := SOLQuery1.FieldByName('URIFL6').AsString: // 金額

```
cShokei := cShokei + SQLQuery1.FieldByName('URIFL7').AsCurrency;
```





佐田 雄一

株式会社ミガロ.

システム事業部 システム1課

[Delphi/400] FireDAC 実践プログラ ミングテクニック

はじめに 1. FireDAC のデータ取得に関するテクニック 2. 2-1. データベースエンジン変更時のマッピングルール 2-2. ソート順の指定方法 2-3. フェッチによるパフォーマンス効果 FireDAC のデータ更新に関するテクニック 3-1. 双方向更新機能の活用 3-2. FireDAC のトランザクション制御 ファイルメンバーの制御 4. まとめ 5.



略歴 1985年12月6日生まれ 2009年3月甲南大学経営学部卒

** 2009 年 4 月 株式会社ミガロ. 入社 2009 年 4 月 システム事業部配属

現在の仕事内容:

Delphi/400を利用したシステム開 発や保守作業を担当。Delphi およ び Delphi/400 のスペシャリスト を目指して精進している。

1.はじめに

Delphi/400 10 Seattle のリリース以 降、新規開発や他の接続方式からの移行 において最新のデータベースエンジンで ある FireDAC が採用されることが多く なった。2016年版『ミガロ.テクニカル レポート』の SE 論文でも、「新データ ベースエンジン FireDAC を使ってみ よう!」と題して FireDAC の基本的な 使用方法を紹介している。

本稿では、FireDAC をさらに使いこ なすための実践的なテクニックを検証 し、紹介していく。データベースエンジ ンの操作には大きく分けてデータ取得と データ更新があり、各章に分けて技術ト ピックをまとめている。

なお、本稿ではDelphi/400の最新バー ジョンである Delphi/400 10.2 Tokyo の環境を使用している。

2.FireDACのデータ 取得に関する テクニック

2-1.データベースエンジン変更時の マッピングルール

Delphi/400とIBM i との間でデータ をやり取りする場合、値の受け渡しのた めに、データベースエンジンではコン ポーネントでデータ型を自動的に設定す る。

このデータ型のルールはデータベー スエンジンによって異なり、BDE、 dbExpress、FireDAC では一致しない データ型もある。たとえば整数や小数4 桁以内の実数は BCD 型、小数5桁以上 の実数は FMTBCD 型として扱われる。 図1】 (図2)

FireDAC で新規開発する場合はその ままで問題ないが、従来の BDE や dbExpress から FireDAC に移行する ときは、データ型が異なる部分を変更し ておく必要がある。たとえばデータの処 理時に FireDAC が受け取るデータ型 が、既存のプログラムで設定されている データ型と差異がある場合、そのまま処 理すると、エラーが発生してしまう【図3】。

また、FireDAC で新規開発する場合 でも、BCD 型のフィールドは内部的に 数値を Currency として保持するため、 「17.0 桁」のような小数4桁以内かつ Currency 型で扱えない巨大な桁数を セットしようとするとエラーになってし まう。【図4】

これらの現象に対してマッピング(変 換) のルールを設定することで、データ ベースエンジン間での違いを吸収し、差 異があるデータ型を従来と同様のデータ 型として扱うことが可能となる。以下に 設定手順を紹介する。

まず、TFDConnection コンポーネン トをダブルクリックすると、FireDAC 接続エディタが表示される。ここで設定 した内容は、 紐付く 各 TFDQuery や TFDTable にも一括で適用される。個 別の TFDQuery や TFDTable コンポー

図1 接続方式とDelphi/400のデータ型

データ タイプ	型	BDE	dbExpress	FireDAC
Α	半角のみの文字型	TStringField	TStringField	TStringField
0	全半角混合の文字型	TStringField	TStringField	TStringField
J	全角のみの文字型	TStringField	TStringField	TStringField
L	日付型	TDateField	TDateField	TDateField
Т	時刻型	TTimeField	TTimeField	TTimeField
Z	タイムスタンプ型	TDateTimeField	TSQLTimeStampField	TSQLTimeStampField ★
	1~9桁の整数型	TIntegerField	TIntegerField	TBCDField ★▼
Р	10~18桁の整数型	TFloatField	TFloatField	TBCDField ★▼
または S	有効桁数が18以内 かつ 小数4桁以内の実数型	TFloatField	TFloatField	TBCDField ★▼
	上記以外の数値型	TFloatField	TFloatField	TFMTBCDField ★▼
В	2進数型	TIntegerField	使用不可	使用不可
н	16進数型	TStringField	使用不可	使用不可

★=BDEから移行時にマッピングが必要

▼=dbExpressから移行時にマッピングが必要

図2 フィールド型の違い

	TUTADIEZSHUOUU TBUDHeid
SHAA18 SHOO18	の検索
SHDATE	
SHTIME	ConstraintEr 80桁の整数フィールドだが
SHJJ18	
SH0900	currency False マッピング木設定の場合はDOD空として扱う
SH1000	CustomCons (BDEやdbExpressではInteger型)
SH1100	DefaultExpre
SH1200	DisplayForma
SH1400	Display of the
SH0802	DisplayLadel SH0800
SH0902	DisplayWidth 9
SH1002 SH1102	EditFormat
SH1202	FieldKind fkData
SH1302	FieldName SH0800
SH1402	Fleidvane Shooo
SH1502 SH1602	Hasconstrain Halse
SH0804	ImportedCor
SH0904	Index 5
SH1004	KevFields E
SH1204	TiveBinding ∃LiveBinding ≓tf≠+
SH1304	Lade Cable El Cable
SH1404	
SH1504	LookupDataS
SH1704	LookupKeyFi
SH1804	LookupResult
SH0806	MaxValue 0
SH1006	Matter o
SH1106	Mirivalue
SH1206	Name FDTable2SH0800
SH1406	Origin SH0800
SH1506	Precision 8
	Provider Flags [nfinl Jodate.nfinWhere]
	Bandonky III Falco
	reduciny Paise
	クイック編集 クイックコピー名

ネント単位でも同様の接続エディタを 持っているため、特定の1つのコンポー ネントのみ例外的に特殊処理が必要な場 合は、個別に設定を行うことも可能であ る。

次にオプションタブを選択し、「継承 したルールを無視」チェックボックスを オンにすると、データマッピングルール の明細が入力可能になる。エラーメッ セージの内容が、たとえば先の【図3】 のように「FDTable1: フィールド'〇〇 ○'の型が一致しません。Integer が必 要ですが実際は BCD です.」と表示さ れた場合、BCD 型で受け取る数値を Integer 型として認識させればエラーと ならずに読み書きが行える。また、IBM iと通信を行う際に考慮が必要なフィー ルドの型は、【図1】のように数値以外 にも存在するため、マッピングルール例 としては【図5】のような設定も有効で ある。

なお、マッピングの設定を行っていな いと、数値をBCD型(TBCDField) としてIBM iと値の受け渡しを行うこ とになるが、Delphi/400側では TBCDFieldを内部的にCurrencyに一 度変換するため、【図6】【図7】のよう に従来とおりのロジックでフィールド値 の読み書きを行うことができる。

2-2.ソート順の指定方法

IBM i からデータを複数レコード取得 する際、その並び順は大きく分けて「① EBCDIC 順」「② ASCII 順」「③到着順」 の3種類のルールが存在する。EBCDIC 順は IBM i と同じ「A ~ Z \rightarrow 0 ~ 9」 の並び順、ASCII 順は Windows と同 様の「0 ~ 9 \rightarrow A ~ Z」の並び順、そし て到着順は対象ファイルのレコード登録 順 (物理レコード順) となっている。【図 8】

以下に、【図9】のようなファイルが 存在した場合に、FireDAC においてそ れぞれの並び順でデータを取得する方法 を紹介する。

① EBCDIC 順

EBCDIC 順でデータの取得を行う場 合、FireDAC では TFDQuery を使用 す る。TFDQuery の SQL 文 内 で ORDER BY 句を使ってフィールドの並 び順を指定すると、取得されるデータは ORDER BY で指定されたフィールドで EBCDIC 順に並べることができる。 【ソース1】【図 10】。

フィールドに降順を指定する際は、 IBM iの STRSQL コマンド実行時と同 様に ORDER BY 句の中で降順指定 フィールドの後ろに「DESC」と記載す る。ちなみに、BDE や dbExpress では TClientDataSet と 紐 付 け を 行 い、 TDataSetProvider の設定値によって EBCDIC 順になるよう指定する方法が ある。

② ASCII 順

ASCII 順でデータのソートを行う場 合、Delphi/400 のクライアント側で Index を設定する方法が最もシンプルで ある。TFDTable を使用して接続する 場合、TFDTable 自身または紐付けた TClientDataSet に、並べたい順にセミ コロン区切りで IndexFieldNames プロ パティを設定する。フィールドに降順を 指定したい場合は、TFDTableの IndexFieldNames プロパティに [(フィールドID):D]と指定することで、 そのフィールドだけ降順にすることがで きる。【ソース2】【図11】

なお、従来の BDE や dbExpress で 降順を指定する際に必要であった、 TClientDataSet 側の IndexDefs および IndexName プロパティで降順フィール ドを別途指定する方法は FireDAC でも 使用可能である。

また、TFDQuery を使用している場 合、TClientDataSet に紐付けた上、先 述の IndexDefs および IndexName プ ロパティを指定する。降順がない場合は IndexFieldNames プロパティでもよい。 この Index 指定は ORDER BY 句より も優先されるため、ASCII 順の並びで データが表示される。

③到着順

では、特に並び順を指定せずに TFDTableやTFDQueryをオープンす るとどうなるのか。答えは到着順になる 場合と、EBCDIC順になる場合の両方 がある。並び順を指定していない場合は、 並び順が保証されておらず、どちらの並 び順になるかは、IBM iのSTRSQLコ マンドでSQLを実行すると確認が可能 である(STRSQLの結果と同じ並び順 になる)。

意図的に到着順でレコードを表示さ せたい場合は、TFDQueryを使用し、 SQL文内でRRN(物理ファイルが内部 保持している相対レコード番号)に対し てORDER BY 句を掛けることで、対 象ファイルのレコードを到着順(=レ コードが追加された順)に並べることが 可能となる。【ソース3】【図 12】。

2-3.フェッチによるパフォーマンス 効果

TFDConnection コンポーネントの FetchOptionsプロパティによって、デー タをクライアント側へ転送する際の設定 を変更できる。たとえば数十万件といっ た大量データをオープンしようとする と、それだけで処理に時間がかかること が多い。しかし FireDAC の場合は、初 期設定で 50 件ずつレコードがフェッチ されるようになっており、データを 50 件ずつ取得・表示することによって、オー プン命令からすぐに 50 件のレコードが 表示できる。【図 13】

この際、明細表示後にカレントレコー ドが最終レコード(50 件目)まで到達 した状態で次のレコードを取得しようと すると、次の 50 件を取得する。この件 数は、FetchOptions プロパティ内の RowsetSize サブプロパティで変更がき る。この機能は、従来から使われる TClientDataSet の PacketRecords プ ロパティと同じような使い方ができる。

逆にデータの件数があまり多くなく、 かつデータを一括で全件表示させる必要 がある場合は、初期設定で50件ずつに なっているレコードのフェッチの設定を 無制限になるように設定できる。設定は FetchOptions プロパティ内の Mode サブ プロパティを初期設定の「fmOnDemand」 から「fmAll」に変更する。【図 14】

なお、TClientDataSet と紐付けてい る場合は RowsetSize の値よりも TClient DataSet の PacketRecords の値が優先 される。この構成では実際にデータを持 つコンポーネントが TClientDataSet と なるからである。



3.FireDACのデータ 更新に関する テクニック

3-1.双方向更新機能の活用

データベースエンジンには読み込ん だレコードを直接更新できる「双方向 データセット形式」と、読み込み専用で 更新はできないがパフォーマンスが高い 「単方向データセット形式」がある。

従来の BDE 接続では TTable が双方 向データセット形式で、dbExpress で は単方向データセット形式になってい る。

FireDAC では BDE と同様に双方向 データセット形式となっているため、適 切な設定を行えば読み込んだレコードへ 直接更新することができる。以下に手順 を記載する。

まず、先述のマッピングの設定と同 様、TFDConnection 側 で Update Options プロパティを【図 15】のよう に設定する。

この設定が行われていないと TFDTableやTFDQueryをオープンし た際に読み取り専用として開くため、 データの更新処理は正しく行えず、エ ラー等が発生する可能性がある。

次に、読み込んだデータを編集して更 新する場合の設定を行う。それぞれ目的 にあわせて UpdateOptions プロパティ の UpdateMode サブプロパティを【図 16】のように設定する。

TFDTable における各処理では内部 的に SQL 文を生成・実行する仕組みを 持っており、レコード参照・追加・変更・ 削除時に、実際にはそれぞれ SELECT・ INSERT・UPDATE・DELETE の SQL が TFDTable によって発行されて いる。

この中で UPDATE または DELETE を行う場合には、UpdateMode サブプ ロパティの設定値によって、どのレコー ドを更新対象とするかが決定する。【図 16】のとおり、UpdateMode で設定し た値によって、どのフィールドを更新条 件とするかが決定して WHERE 句を生 成している。【図 17】

続いて、更新条件とするフィールド側 にも設定を行う。オブジェクトインスペ クタ上にフィールドがある場合は【図 18】のように指定する。また、設計画面 上ではフィールドが存在せず、ソース内 で設定を行う場合は【ソース4】のよう に指定する。なお、UpdateMode サブ プロパティが「upWhereAll」の場合は 全フィールドを更新条件とするため指定 不要であるが、IBM iのデータに PC 側 で扱えないコード等が含まれていると条 件が一致しない可能性があるので注意が 必要である。

UpdateMode サブプロパティが 「upWhereAll」以外の場合、この指定 を行っておかないと、Postの際にどの フィールドを基準に更新(WHERE句 を作成)するかプログラムが把握できな いため、注意が必要である。 「upWhereKeyOnly」の場合は更新条件 となるフィールドがないため、プログラ ムが自動的にupWhereAllとして更新 を行う。「upWhereChanged」の場合は 値を変更したフィールドだけが更新条件 になるため、同じ値の他レコードもすべ て更新対象になってしまう。

3-2.FireDACのトランザクション 制御

FireDAC におけるトランザクション 制御は、従来の BDE と近い実装方法で 行うことができる。設定方法を説明する。

まず TFDConnection をダブルクリッ クして FireDAC 接続エディタを開き、 ODBCAdvanced パラメータを【図 19】 のように設定する。このパラメータには ライブラリリストなどの他の指定もでき るが、複数設定の指定を行う場合は、セ ミコロンで区切る必要がある。

各機能においてトランザクション処 理を行う方法は、従来の BDE と同様で ある。【ソース 5】

トランザクション制御において従来 のBDEと異なり注意が必要となるポイ ントを少し補足する。それはジョブ終了 時の制御である。

BDE で は、StartTransaction か ら Commit までの間に TDatabase との接 続を明示的に終了した場合、ジョブが終 了するためトランザクションはロール バックされる。しかし FireDAC では、 接続終了時のデフォルトの動作設定の初 期値が Commit になっているため、ト ランザクションがコミットされる。これ を BDE と同様に接続終了時のデフォル トの動作設定を Rollback にするために、 【図 20】のように TxOptions プロパティ の設定を行う。なお、Delphi/400 アプ リケーションを強制終了した場合など、 ジョブが異常終了する場合は BDE と同 様ロールバックされる。

4.ファイルのメンバー 制御

最後にデータの取得・更新に共通した テクニックとしてメンバーの制御方法を 紹介する。IBM i では1つのデータベー ス・ファイルに対して複数のメンバーを 持つことができる。たとえば、ワークファ イルを使用する際にユーザーごとにそれ ぞれメンバーを指定したい場合、従来の BDE では TTable の TableName プロ パティでメンバーを直接指定できるが、 FireDAC では SQL で動作するため直 接指定できない。【図 21】

このとき任意のメンバーに対して接 続する方法は大きく2種類存在する。

1つ目は、ネイティブ接続と併用可能 な場合にTAS400コンポーネントから OVRDBFコマンドを発行し、ファイル 名を一時変更する方法である。【ソース 6】のように、ロジック内でTFDTable をオープンする前にネイティブ接続側で OVRDBFコマンドを直接発行すれば、 DLTOVRコマンドで上書きするまで、 指定したメンバーを参照・更新できるよ うになる【図 22】。必要に応じて、 TFDTableをクローズした後に、 DLTOVRコマンドで一時変更を削除す れば解除できる。

2つ目は、FireDAC単体でメンバー を制御する方法である。これはWebア プリケーションやDataSnapアプリケー ションといったネイティブ接続とジョブ が別れてしまう構成で有効な実装方法と なる。

まず SQL で実行できる「CREATE ALIAS」「DROP ALIAS」でメンバー に対して別名を作成し、TableName プ ロパティでそのエイリアスを指定する。 【ソース7】【図 23】

この場合は OVRDBF とは異なり、 CREATE ALIAS は別名で実体を作成 するため、既に存在する名前でエイリア



スを作成しようとするとエラーになって しまう。名前の一部にジョブ番号を使用 するなど、エイリアス名が重複しないよ う留意する必要がある。 以上が FireDAC でのメンバーの制御 テクニックである。

5.まとめ

本稿では、Delphi/400の開発者目線 で、検証済みの技術ポイントを中心に、 実践的な FireDAC のテクニックを紹介 した。

最近は Windows10 対応などを中心に Delphi/400 のバージョンアップを行う 開発も多く、それに伴って BDE や dbExpress から FireDAC へ変更する 機会が増えている。その中で、参照・更 新や、メンバー処理にまつわるテクニッ クを把握しておけば、これまでのプログ ラムをスムーズに移行できる。

これまでのシステム開発経験を本稿 で共有することで、Delphi/400の新機 能である FireDAC を広く活用していた だければ幸いである。

Μ



上からこ (登録後	に一部レコードを	削除し、現在は16	レコード)					
行	L国1日化・・ + 1	· · · · · · · · · · · · · · · · · · ·	2	+	フィールドについて (SHAA10~12はいずれも)	Aタイプ)		
1.1	SHAA10	SHAA11	SHAA12		 SHAA10=キー、昇順 			
000001	MIGARO_001	SANKEIBLD_1	REPORT_00	1000	値は英字始まりのレコ	ードと		
000002	MIGARO_001	AAAAAAAAA_1	REPORT_00	0002	数子始まりのレコート:	か准仕		
000003	MIGARO_001	TECHNICAL_2	REPORT_00	0005	②SHAA11=非キー	た庙た祭祭		
000004	MIGARO_001	123456789_2	REPORT_O	0006		る喧こ五球		
000005	MIGARO 002	OSAKACITY 1	REPORT OF	0009	③SHAA12=非キー "REPORT "+レコード発	を録時の連番を登録	ž	
000007	MIGARO 002	XXXXXXXXXX 2	REPORT 00	0013	(※削除レコードの連番	手は欠番)		
000008	MIGARO_002	TECHNICAL_2	REPORT_00	0014				
000009	356356_001	TECHNICAL_1	REPORT_00	0017				
000010	356356_001	33333333333333_1	REPORT_O	018				
000011	356356 001	NANIWA_NU_2 テカニカルレホ - 9	REPORT OF	021				
000012	MIGARO 002	SYSTEM1KA 1	REPORT 00	0025				
000014	356356_002	MIGARON_1	REPORT_00	0026				
000015	356356_002	MINATOMAC_2	REPORT_00	0029				
000016	356356_002	XYZXYZXYZ_2	REPORT_00)030				
*****	******	取合者の終わり) *******	ĸ				
771 EE proced begin FDQu FDQu	BCDIC順参照(ure TForm1.B ery1.Close; ery1.SQL.Cle	のソース記述例 utton4EClick(ar;	Sender: TOb	ject);				
proced begin FDQu FDQu FDQu	CDIC順参照(ure TForm1.B ery1.Close; ery1.SQL.Cle ery1.SQL.Add ery1.SQL.Add	Dソース記述例 utton4EClick(ar; ('SELECT * F ('ORDER BY S	Sender: TOb ROM TR11F05 HAA11, SHAA	<pre>>ject); ;; ;); ;12 ');</pre>				
FDQu FDQu FDQu FDQu FDQu FDQu FDQu FDQu	BCDIC順参照(ure TForm1.B ery1.Close; ery1.SQL.Cle ery1.SQL.Add ery1.SQL.Add ery1.Open;	のソース記述例 utton4EClick(ar; ('SELECT * F ('ORDER BY S	Sender: TOb ROM TR11F05 HAA11, SHAA	ject); ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	yのSQLに、STRSQLで指定	する際と	7	
71 EE proced begin FDQu FDQu FDQu FDQu FDQu end;	BCDIC順参照(ure TForm1.B ery1.Close; ery1.SQL.Cle ery1.SQL.Add ery1.SQL.Add ery1.Open;	Dソース記述例 utton4EClick(ar; <u>('SELECT * F</u> (<u>'ORDER BY S</u>	Sender: TOb ROM TR11F05 HAA11, SHAA	oject); ; '); ;12 '); 「TFDQuer 同じよう	yのSQL1こ、STRSQLで指定 5 IこORDER BYを掛ける	する際と]	
-Z1 EE proced begin FDQu FDQu FDQu FDQu FDQu FDQu FDQu SHAA10 SHAA0	CDIC順参照の ure TForm1. B ery1. Close; ery1. SQL. Cle ery1. SQL. Cle ery1. SQL. Add ery1. SQL. Add ery1. Open; DIC順参照の SHAA11 001 テク=フルレム*~_2 001 AAAAAAAAA. 002 MIGARON1 002 MIGARON1 002 SYSTEM1KA_1 002 SYSTEM1KA_1 002 TECHNICAL_1 001 TECHNICAL_1 001 TECHNICAL_2 001 TECHNICAL_2	Dソース記述例 utton4EClick(ar; ('SELECT * F ('ORDER BY S	Sender: TOb ROM TR11F05 HAA11, SHAA SQLで指定 EBCDIC順(SHAA11の身	oject); (12'); TFDQuer 同じょう によう によう 子属順>SH	yのSQLに、STRSQLで指定 5 IこORDER BYを掛ける	する際と で		



図13 フェッチのレスポンス

SHAA18 A型	SH0018 O型	SHDATE 日付型	SHTIME 時刻型	SHJJ18 J型	SH0800	SH0900	SH1000	SH1100	SH1200	SH1300	SH1400
A000000001		2018/08/17	18:50:39		12345678	123456789	1234567890	2345678901	3456789012	4567890123	56789012
A000000002		2018/08/17	18:50:39		12345678	123456789	1234567890	2345678901	3456789012	4567890123	56789012
A000000003		2018/08/17	18:50:39		12345678	123456788	1234567890	2345678901	3456789012	4567890123	56788012
A000000004		2018/08/17	18:50:89		12345878	123456788	1234567890	2345678901	3456789012	4567890123	56789012
A000000005		2018/08/17	18:50:39		12345878	123456789	1234567890	2345678901	3456789012	4567890123	56789012
A000000006		2018/08/17	18:50:39		12345678	123456789	1234567890	2345678901	3456789012	4567890123	56789012
A000000		2018/08/17	18:50:39		12345678	123456789	1234567890	2345678901	3456789012	4567890123	56789012
▲000000 (中略)		2018/08/17	18-50-39		12345878	123456789	1234567890	2345678901	3456789012	4567890123	58789012
A000049		2018/08/17	18:50:39		12345678	123456789	1234567890	2845678901	8456789012	4567890123	56789012
A000049995		2018/08/17	18:50:39		12345678	123456789	1234567890	2345678901	3456789012	4567890123	56789012
A000049996		2018/08/17	18:50:39		12345678	123456789	1234567890	2345678901	3456789012	4567890123	56789012
A000049997		2018/08/17	18:50:39		12345678	123456789	1234567890	2345678901	3456789012	4567890123	56789012
A000049998		2018/08/17	18:50:39		12345678	123456789	1284567890	2345678901	8456789012	4567890123	56789012
A000049999	\sim							5678901	3456789012	4567890123	56789012
A000050000	50,000	レコードのデ	ータをオー	プンしてから	表示まで	の所要問	寺間:	5678901	3456789012	4567890123	56789012
	. 7 -	ッチ 実施時()	初期設定50	/コード単位) = 0.2	30秒				den en e	
		ノノ 天心时 (MALEX COUL		./ - 0.2	0019					
	・フェ	ツナ木実施時			=10.5	80秒					

図14 フェッチ設定の変更



> オブジェクト インスペク	9	FDC0400ドライバーの場合、UpdateOptions	
FDConnection1 TFDConne	ection	プロパティを次のとおり設定する	
₽ 検索			
プロパティレイベント			
StopOptions	[xoIfCmdsInactive,xoIfA	CheckReadOnly : False	
UpdateOptions	(TFDUpdateOptions)	(Trueだと編集できない)	
Assigned values			
CheckReadOnly	False	CheckRequired : False	
CheckRequired	T False	(Irueだと更新時に空のフィールドがある	
CheckUpdatable	False	とエラーになる	
CountUpdatedRecords	True	CheckUpdatable : False	
EnableDelete	📝 True	(更新時のTableの状態監視を止める)	
EnableInsert	True		
EnableUpdate	True		
Fastopuates	raise		
GeneratorName	Shereico		
LockMode	ImNone		
LockPoint	lpDeferred	RefreshMode rmManual	
LockWait	🔄 False	(更新後、レコードを自動リフレッシュ)	
ReadOnly	False	ない。リフレッシュ時のエラー防止)	
RefreshDelete	V True	1	
RefreshMode	rmManual 2	UpdateOptions : upWhereChanged	
UpdateChangedFields	V True	(計础は後述)	
UpdateMode	upWhereChanged	UpdateNonBaseFields True	
UpdateNonBaseFields	True	(更新時に正しくテーブルが指定されてい	
UpdateTransaction		ないという内部エラー防止)	
接続エディタ クイック編集 クイ	(ックコピー名		
すべての項目が表示されています			
すべての項目が表示されています			
すべての項目が表示されています			
すべての項目が表示されています UndateOntionsの	設定		
すべての項目が表示されています UpdateOptionsの	設定		
すべての項目が表示されています UpdateOptionsの	設定		
すべての項目が表示されています UpdateOptionsの LockWait ReadOnly	設定 日 Faise 日 Faise	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する	
すべての項目が表示されています UpdateOptionsの Lockwait ReadOnly RefreshDelete Percention	設定 Faise Faise 了True	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する	
すべての項目が表示されています UpdateOptionsの Lockwait ReadOnly RefreshDelete RefreshMode Bequest ive	設定 False False 了True rmManual	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する	
すべての項目が表示されています UpdateOptionsの LOCKWait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedField	設定 False 了 False 了 True rmManual 了 True	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する	
 すべての項目が表示されています UpdateOptionsの Lockwait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedField >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	設定 False False True rmManual True True UpWhereKeyOply	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する	
	設定 False False True rmManual True ds UpWhereKeyOnly upWhereAll	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する	
 すべての項目が表示されています UpdateOptionsの Lockwait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedFiele >> UpdateMode UpdateNonBaseFiele >> UpdateTransaction 	設定 False False True rmManual 了True ds 可True UpWhereKeyOnly upWhereAll upWhereChanged	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する	
 すべての項目が表示されています UpdateOptionsの LOCKWait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedField WpdateMode UpdateTransaction 接続工ディタ カイックを使用しています 	設定 False False True rmManual でTrue ds UPWhereKeyOnly upWhereAll upWhereChanged UPWhereKeyOnly	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する	
 すべての項目が表示されています UpdateOptionsの LOCKWait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedField UpdateNonBaseField UpdateTransaction 接続エディタ クイック編集 	設定 False False 「True rmManual 「True す True UpWhereKeyOnly upWhereAll upWhereKeyOnly upWhereKeyOnly DupWhereKeyOnly	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する	
 すべての項目が表示されています UpdateOptionsの LOCKWait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedField UpdateNonBaseField UpdateTransaction 接続エディタ クイック編集 	設定 False False True True True True True UpWhereKeyOnly upWhereKeyOnly upWhereKeyOnly topWhereKeyOnly Light~Postまた(はDel 更新を供たするフィー	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する	
 すべての項目が表示されています UpdateOptionsの Lockwait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedFiele UpdateNonBaseFiel UpdateTransaction 接続エディタ クイック編集 すべての項目が表示されてい 	設定 False False True rmManual True mManual True upWhereKeyOnly upWhereAll upWhereChanged upWhereKeyOnly Edit~PostまたはDel 更新条件とするフィー	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する	
 すべての項目が表示されています UpdateOptionsの LOCKWait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedFiele UpdateNonBaseFiel UpdateTransaction 接続エディタ クイック編集 すべての項目が表示されてい 	設定 False False True rmManual True mManual True upWhereKeyOnly upWhereAll upWhereKeyOnly to the false Edit ~ PostまたはDel 更新条件とするフィー upWhereAll	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する	
 すべての項目が表示されています UpdateOptionsの LOCKWait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedField UpdateNonBaseFiel UpdateTransaction 接続エディタ クイック編集 すべての項目が表示されてい 	設定 False False False True True True True UpWhereKeyOnly UpWhereAll UpWhereKeyOnly Edit~PostまたはDel 更新条件とするフィー UpWhereAll →全てのフィールドを	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する ete時に、 ールドを判定 E更新条件とする	
 すべての項目が表示されています UpdateOptionsの LOCKWait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedField UpdateNonBaseField UpdateTransaction 接続エディタ クイック編集 すべての項目が表示されてい 	設定 □ False □ False □ True rmManual □ True v True v True v True v True v True v VhereKeyOnly upWhereAll upWhereChanged upWhereKeyOnly Edit~PostまたはDelf 更新条件とするフィー upWhereAll →全てのフィールドを (全角フィールドか	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する ete時に、 ールドを判定 を更新条件とする が正しく判定されない場合がある)	
 すべての項目が表示されています UpdateOptionsの Lockwait ReadOnly RefreshDelete RefreshMode UpdateChangedField UpdateNonBaseField UpdateTransaction 接続エディタ クイック編集 すべての項目が表示されてい 	設定 □ False □ False □ True rmManual □ True ds □ True upWhereKeyOnly upWhereAll upWhereChanged <u>upWhereKeyOnly</u> Edit~PostまたはDelf 更新条件とするフィー upWhereAll →全てのフィールドを (全角フィールドか	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する ete時に、 -ルドを判定 を更新条件とする が正しく判定されない場合がある)	
 オペての項目が表示されています UpdateOptionsの LOCKWait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedFiele UpdateNonBaseFiel UpdateTransaction 接続エディタ クイック編集 すべての項目が表示されてい 	設定 False False False True rmManual True mManual True upWhereKeyOnly upWhereAll upWhereChanged <u>upWhereKeyOnly</u> Edit~PostまたはDelf 更新条件とするフィー upWhereAll →全てのフィールドを (全角フィールドか upWhereChanged →別途指定した更新名	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する ete時に、 ールドを判定 を更新条件とする が正しく判定されない場合がある) E性フィールド	
 オペての項目が表示されています UpdateOptionsの LOCKWait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedField UpdateMode UpdateTransaction 接続エディタ クイック編集 すべての項目が表示されてい 	設定 False False False True rmManual True mManual True upWhereKeyOnly upWhereAll upWhereAll upWhereKeyOnly Edit~PostまたはDel 更新条件とするフィー upWhereAll →全てのフィールドを (全角フィールドか upWhereChanged →別途指定した更新祭 および、今変更した	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する ete時に、 -ルドを判定 を更新条件とする が正しく判定されない場合がある) k件フィールド こフィールド(Edit時)を更新条件とする	
 オペての項目が表示されています UpdateOptionsの LOCKWait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedField UpdateNonBaseField UpdateTransaction 接続エディタ クイック編集 すべての項目が表示されてい 	設定 False False False True True True True UpWhereKeyOnly UpWhereAll UpWhereKeyOnly DiffereAll Differ	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する ete時に、 ールドを判定 を更新条件とする が正しく判定されない場合がある) &件フィールド EPUTY (Edit時)を更新条件とする	
 オペての項目が表示されています UpdateOptionsの LOCKWait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedField UpdateNonBaseField UpdateTransaction 接続エディタ クイック編集 すべての項目が表示されてい 	設定 □ False □ False □ False □ True True □ True □ True □ True □ True □ True □ True □ True □ True □ True □ WhereKeyOnly upWhereAll upWhereChanged □ DWhereKeyOnly Edit~PostまたはDel 更新条件とするフィー upWhereAll →全てのフィールドを (全角フィールドを □ 方 □ 別途指定した更新条 および、今変更した □ 新祭	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する サブプロパティを次のとおり設定する ete時に、 -ルドを判定 を更新条件とする が正しく判定されない場合がある) &件フィールド =フィールド(Edit時)を更新条件とする B値)	
 すべての項目が表示されています UpdateOptionsの Lockwait ReadOnly RefreshDelete RefreshMode UpdateChangedField UpdateNonBaseFiel UpdateTransaction 接続エディタ クイック編集 すべての項目が表示されてい 	設定 □ False □ False □ False □ True rmManual □ True vg True upWhereKeyOnly upWhereAll upWhereChanged <u>vgWhereKeyOnly</u> Edit~PostまたはDelf 更新条件とするフィー upWhereAll →全てのフィールドを (全角フィールドを なした更新条 および、今変更した upWhereKeyOnly(初其 →別途指定した更新条	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する ete時に、 ールドを判定 を更新条件とする が正しく判定されない場合がある) 条件フィールド(Edit時)を更新条件とする 8倍フィールド(Edit時)を更新条件とする	
 オペての項目が表示されています UpdateOptionsの Lockwait ReadOnly RefreshDelete RefreshMode RequestLive UpdateChangedFiele UpdateNonBaseFiel UpdateTransaction 接続エディタ クイック編集 すべての項目が表示されてい 	設定 □ False □ False □ True True □ True □ True □ True □ True □ True □ True □ True □ True □ True □ WhereKeyOnly ds □ WhereChanged □ WhereAll □ WhereAll □ WhereAll □ A C O D T	目的にあわせて、UpdateMode サブプロパティを次のとおり設定する ete時に、 ールドを判定 を更新条件とする が正しく判定されない場合がある) E件フィールド(Edit時)を更新条件とする Alignment (Edit時)を更新条件とする Alignment (Edit時)を更新条件とする Alignment (Edit時)を更新条件とする Alignment (Edit時)を更新条件とする Alignment (Edit時)を更新条件とする Alignment (Edit (Edi	



begin		
TableNamo '- 'T	R11E02'	
Open:	NTIT 02 ;	
// 一意になる	ように更新キーフィールド	を指定(複数指定可能)
FieldByName('	SHAA18').ProviderFlags 🔅	= [pfInKey]; // キーフィールド
First;		
for i := 1 to	ataliat DowCount 1 da	フィールドのProviderFlagsプロパティを
hegin	Siglisi. Kowoouni - 1 do	直接ソースで指定する
Edit:		
//(各フィー	-ルド値セット)	
Post;		
Next:		
end;		
end,		
UTU ,		
トランザクションに必要	国な設定	
💿 FireDAC 接続エディタ - [FD	Connection1]	
	特殊学業の名前を運搬してから パラマンカキャ	25. 25. 73. 11 + 7
I 21/10/00/2 /1 21/30	NROLENSO - INI EXANO CASCASS - SEC	31737049
定義 オブション 情報 SQL ス	、クリプト	
定義 オブション ドライバ ID(D):	クリプト	•
定義 オプション 情報 SQL 7 ドライバ ID(D): 接続定義名(M): CO40	りファト DODEF	•
定義 オブション 情報 SQL 7 ドライバID(D): 接続定義名(M): CO40	ンクリプト 2000EF S(W) デフォルト(に戻す(R)) ヘルプの	•
定義 オブション 情報 SQL 7 ドライバID(D): 接続定義名(M): CO4(テスト(1) ウィザート	2017ト 2000EF ※(W) 「デフォルトに戻す(B)」 ヘルプ企	
定義 オブション 情報 SQL ス ドライバ ID(D): 接続定義名(M): CO4(テスト(D) ウィザート パラメータ	2000EF ⁴ (W) デフォルト(2戻す(R)) ヘルプ(L) (値	 ▼ ♪ 〕 □ □ □ □ □ □
定義 オブション 情報 SQL ス ドライバ ID(D): 接続定義名(M): CO40 テスト(D) ウィザート パラメータ DriverID Pooled	2000EF ² (W) デフォルト(こ戻す(B)) ヘルプ企 値 この400 False	・ ・ ・ ・ 「Commitment- ・ ト記載
定義 オブション 情報 SQL 7 ドライバ ID(D): 接続定義名(M): CO40 テスト(T) ウィザート パラメータ DriverID Pooled Database	2017ト 200EF ^S (<u>W</u>) 「デフォルト(ご戻す(<u>R</u>)」 ヘルプ(L 値 <i>C0400</i> False	・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
定義 オブション 情報 SQL ス ドライバ ID(D): 接続定義名(M): CO4C テスト(T) ウィザート パラメータ DriverID Pooled Database User_Name	2000EF ⁴ (<u>W</u>) デフォルト(2戻す(<u>R</u>) ヘルプ(L (値 <i>CO 400</i> False	・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
定義 オブション 情報 SQL ス ドライバ ID(D): 技統定義名(M): CO40 テスト(I) ウィザート パラメータ DriverID Pooled Database User_Name Password	2000EF <u>*(W) デフォルト(こ戻す(R)</u> ヘルプ企 値 <i>CO 400</i> False	▼ ▼ 「Commitment=●●」と記載 (●●= *CHG *CS *ALL のいずれか) ライブラリなど他の設定がある場合、セミコロンで区切る
定義 オウション 情報 SQL ス ドライバ ID(D): 接続定義名(M): CO4(テスト(T) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy	(5)リフト 500EF 5(W) (デフォルト(C戻す(R)) ヘルプ化 値 CO-400 False	▼ ▼ 「Commitment=●●」と記載 (●●= *CHG *CS *ALL のいずれか) ライブラリなど他の設定がある場合、セミコロンで区切る
定義 オブション 情報 SQL ス ドライバ ID(D): 接続定義名(M): CO40 テスト(T) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy ODBCAdvanced	2000EF ⁵ (<u>W</u>) デフォルトに戻す(<u>R</u>) ヘルプ企 値 <i>CO-400</i> False LibraryOption=;Commitme	▼ ▼ 「Commitment=●」と記載 (●●= *CHG *CS *ALL のいずれか) ライブラリなど他の設定がある場合、セミコロンで区切る ent=*CHG
定義 オウション 情報 SQL ス ドライバ ID(D): 技統定義名(M): CO4(C テスト(T) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy ODBCAdvanced LoginTimeout	2000EF 	・ ・ 「Commitment=●」と記載 (●●= *CHG *CS *ALL のいずれか) ライブラリなど他の設定がある場合、セミコロンで区切る ent=*CHG
定義 オウション 情報 SQL ス ドライバ ID(D): 技統定義名(M): CO40 テスト(T) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy ODBCAdvanced LoginTimeout Alias	2000EF [※] (W) 「デフォルト(ご戻す(R)) へルプ企 値 <i>CO 400</i> False LibraryOption=;Commitme	▼ ▼ 「Commitment=●●」と記載 (●●= *CHG *CS *ALL のいずれか) ライブラリなど他の設定がある場合、セミコロンで区切る =nt=*CHG
定義 オブション 情報 SQL ス ドライバ ID(D): 技統定義名(M): CO40 テスト(I) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy ODBCAdvanced LoginTimeout Alias Server	2000EF ² (W) 「デフォルト(ご戻す(R)」 ヘルプ化 値 CO-400 False LibraryOption=;Commitme	デフォルト 「Commitment=●●」と記載 (●●= *CHG *CS *ALL のいずれか) ライブラリなど他の設定がある場合、セミコロンで区切る ent=*CHG
定義 オウション 情報 SQL ス ドライバ ID(D): 接続定義名(M): CO40 テスト(T) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy ODBCAdvanced LoginTimeout Alias Server Port ExtendedMatadata	2000EF S(W) 「デフォルト(ご戻す(B)) ヘルプ化 値 CO-400 False LibraryOption=;Commitme	マンフォルト 「Commitment=●」と記載 (●●= *CHG *CS *ALL のいずれか) ライブラリなど他の設定がある場合、セミコロンで区切る ent=*CHG
定義 オブション 情報 SQL ス ドライバ ID(D): 接続定義名(M): CO40 テスト(T) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy ODBCAdvanced LoginTimeout Alias Server Port ExtendedMetadata MetaDefSchema	200EF 	・ ・ 「Commitment=●」と記載 (●●= *CHG *CS *ALL のいずれか) ライブラリなど他の設定がある場合、セミコロンで区切る ent=*CHG
定義 オウション 情報 SQL ス ドライバ ID(D): 技統定義名(M): CO40 テスト(T) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy ODBCAdvanced LoginTimeout Alias Server Port ExtendedMetadata MetaDefSchema	200EF 	▼
定義 オフション 情報 SQL ス ドライバ ID(D): 接続定義名(M): CO40 テスト(D) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy ODBCAdvanced LoginTimeout Alias Server Port ExtendedMetadata MetaDefSchema MetaCurSchema	2000EF ^S (W) 「デフォルト(ご戻す(R)) へルプ化 値 CO 400 False LibraryOption=;Commitme True	▼ 「Commitment=●●」と記載 (●●= *CHG *CS *ALL のいずれか) ライブラリなど他の設定がある場合、セミコロンで区切る Fabe Fabe
定義 オウション 情報 SQL ス ドライバ ID(D): 技統定義名(M): CO40 テスト(I) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy ODBCAdvanced LoginTimeout Alias Server Port ExtendedMetadata MetaDefSchema MetaCurSchema	2000EF ⁵ (W) 「デフォルト(ご戻す(R)) へルプ化 値 CO-400 False LibraryOption=;Commitme True	▼
定義 オクション 情報 SQL ス ドライバ ID(D): 接続定義名(M): CO40 テスト(D) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy ODBCAdvanced LoginTimeout Alias Server Port ExtendedMetadata MetaDefSchema MetaCurSchema	200EF S(W) 「デフォルト(ご戻す(R)) へルプ化 値 CO-400 False LibraryOption=;Commitme True	「Commitment=●」と記載 「Commitment=●」と記載 (●●= *CHG *CS *ALL のいずれか) ライブラリなど他の設定がある場合、セミコロンで区切る ent=*CHG
定義 オウション 情報 SQL ス ドライバ ID(D): 技統定義名(M): CO40 テスト(T) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy ODBCAdvanced LoginTimeout Alias Server Port ExtendedMetadata MetaDefSchema MetaCurSchema	200EF [※] (W) デフォルト(ご戻す(R)) ヘルレク(L (値 CO-400 False LibraryOption=;Commitme True	「Commitment=●」と記載 「Commitment=●」と記載 (●●= *CHG *CS *ALL のいずれか) ライブラリなど他の設定がある場合、セミコロンで区切る ent=*CHG
定義 オフション 情報 SQL ス ドライバ ID(D): 接続定義名(M): CO40 テスト(D) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy ODBCAdvanced LoginTimeout Alias Server Port ExtendedMetadata MetaDefSchema MetaCurSchema	2000EF ^S (W) 「デフォルト(ご戻す(R)) へルプ化 (値 CO 400 False LibraryOption=;Commitme True	▼ 「Commitment=●●」と記載 (●●= *CHG *CS *ALL のいずれか) ライブラリなど他の設定がある場合、セミコロンで区切る ent=*CHG Fase QK 年ャンセル
定義 オウション 情報 SQL ス ドライバ ID(D): 技統定義名(M): CO40 テスト(T) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy ODBCAdvanced LoginTimeout Alias Server Port ExtendedMetadata MetaDefSchema MetaDefSchema	(シリプト) DODEF ^S (W) (デフォルト(ご戻す(R)) ヘルプ化 (値 CO-400 False LibraryOption=;Commitme True	デフォルト 「Commitment=●●」と記載 (●●= *CHG *CS *ALL のいずれか) ライブラリなど他の設定がある場合、セミコロンで区切る ent=*CHG Fase QK 年ャンセル
定義 オウション 情報 SQL ス ドライバ ID(D): 技統定義名(M): CO40 テスト(I) ウィザート パラメータ DriverID Pooled Database User_Name Password MonitorBy ODBCAdvanced LoginTimeout Alias Server Port ExtendedMetadata MetaDefSchema MetaCurSchema	(シリプト) DODEF [※] (W) (デフォルト(ご戻す(R)) ヘルプ化 「値 CO-400 False LibraryOption=;Commitme True	『 「Commitment=●」と記載 「Commitment=●」と記載 (● = *CHG *CS *ALL のいずれか) ライブラリなど他の設定がある場合、セミコロンで区切る ent=*CHG Fake CHG Fake CHG Fake CK F

harin	pecubal control rok (our	ider. IUbject),			
begin					
if (not FDConnec	tion1.InTransaction)	then [1
begin	A		トランザクションの開始		
FDConnection1.	StartIransaction;				
end,					
enu,					
procedure TForm2.	peedButton5Click(Ser	der: TObject);			
begin					
if (FDConnection	1. InTransaction) the	n 🗌		2	1
begin			トランザクションのコミ	ット]
FDConnection1.	Commit;				
end;					
ena,					
procedure TForm2 S	needButton6Click(Ser	der: TObject):			
begin					
if (FDConnection	1. InTransaction) the	n r			1
begin			トランザクションのロール	レバック	
FDConnection1.	Rollback:				
end;					
end,					
0 明示切断時設定の					
0 明示切断時設定の	変更点				
0 明示切断時設定の ○ オブジェクトインスへ EDConnection1 JEDCo	変更点 20タ				
0 明示切断時設定の ○ オブジェクトインスへ FDConnection1 TFDCo	変更点 20タ Innection				
0 明示切断時設定の の オブジェクトインスペ FDConnection1 TFDCo P検索	変更点 のrection	Connect i on σ TxOpt	ionsプロパティ内の		
0 明示切断時設定の ○ オブジェクトインスへ FDConnection1 TFDCo ○ 検索 〕 プロパティ 【イペント】 □ PaceureaOntions	変更点 ウタ に 「E nnection レーー「FD Di	D X ConnectionのTxOpt sconnectActionサン Rollbackに指定	ionsプロパティ内の ブプロパティを		
0 明示切断時設定の → オブジェクトインスへ FDConnection1 TFDCo ●検索 「プロパティ【イヘント】 ● ResourceOptions Tan	変更点 nnection (TFDTopResourceOp XC	ConnectionのTxOpt sconnectActionサフ Rollbackに指定	ionsプロパティ内の ブプロパティを		
0 明示切断時設定の → オブジェクトインスペ FDConnection1 TFDCo の検索 「プロパティ イベント」 ■ ResourceOptions Tag Transaction	変更点 nnection (TFDTopResourceOp XC 0	DeconnectionのTxOpt sconnectActionサス IRollbackに指定	ionsプロパティ内の ブプロパティを		
0 明示切断時設定の → オブジェクトインスへ FDConnection1 TFDCa → 検索 「プロパティ イハント」 ■ ResourceOptions Tag Transaction ■ TxOptions	変更点 nnection (TFDTopResourceOt XC 0 (TFDTxOptions)	DConnectionのTxOpt sconnectActionサラ IRollbackに指定	ionsプロパティ内の ブプロパティを		
0 明示切断時設定の ■ オブジェクトインスペ FDConnection1 TFDCa ● 検索 〕プロパティ 【イペント】 ■ ResourceOptions Tag Transaction ■ TxOptions AutoCommit	変更点 かタ nnection (TFDTopResourceOu xc 0 (TFDTxOptions) 『 True	DConnectionのTxOpt sconnectActionサン IRollbackに指定	ionsプロパティ内の ブプロパティを		
0 明示切断時設定の → オブジェクトインスペ FDConnection1 TFDCa → 検索 「プロパティ イベント」 ■ ResourceOptions Tag Transaction ■ TxOptions AutoCommit AutoStart	変更点 nnection (TFDTopResourceOp XC 0 (TFDTxOptions) 『True 『True 『True	DConnectionのTxOpt sconnectActionサコ IRollbackに指定	ionsプロパティ内の ブプロパティを		
0 明示切断時設定の ■ オブジェクトインスへ FDConnection1 TFDCa ● 検索 「プロパティ イベント ■ ResourceOptions Tag Transaction ■ TxOptions AutoCommit AutoStart AutoStop	変更点 nnection (TFDTopResourceOt XC 0 (TFDTxOptions) ② True ② True ② True ② True	DConnectionのTxOpt sconnectActionサラ IRollbackに指定	ionsプロパティ内の ブプロパティを		
0 明示切断時設定の ■ オブジェクトインスペ FDConnection1 TFDCa ● 検索 〕プロパティ 【イペント】 ■ ResourceOptions Tag Transaction ■ TxOptions AutoCommit AutoStart AutoStart AutoStop DisconnectAction	変更点 クタ nnection (TFDTopResourceOp 0 (TFDTxOptions) 「True 「True 「True 「True 「True 「True 「True 「True	DConnectionのTxOpt sconnectActionサフ IRollbackに指定	ionsプロパティ内の ブプロパティを		
0 明示切断時設定の → オブジェクトインスペ FDConnection1 TFDCa → 検索 「ゴロ/(ティ 【ハント ■ ResourceOptions Tag Transaction ■ TxOptions AutoCommit AutoStart AutoStop DisconnectAction EnableNested	変更点 nnection (TFDTopResourceO) の (TFDTxOptions) 『 True 『 True 『 True 『 True 『 True 『 True 』 True	DConnectionのTxOpt sconnectActionサン Rollbackに指定	ionsプロパティ内の ^プ プロパティを		
0 明示切断時設定の ■ オブジェクトインスペ FDConnection1 TFDCa ● 検索 「プロパティ イベント ■ ResourceOptions Tag Transaction ■ TxOptions AutoCommit AutoStart AutoStart AutoStart AutoStart AutoStart AutoStart AutoStart AutoStart AutoStart	変更点 nnection (TFDTopResourceOt XC 0 (TFDTxOptions) 『True 『True 『True 『True 』True 』True 』True 』True 』True 』True	DConnectionのTxOpt sconnectActionサラ IRollbackに指定	ionsプロパティ内の ブプロパティを		
0 明示切断時設定の → オブジェクトインスペ FDConnection1 TFDCo ● 検索 → プロパティ イペント ■ ResourceOptions Tag Transaction ■ TxOptions AutoCommit AutoStart	変更点 クタ nnection (TFDTopResourceOp 0 (TFDTxOptions) 「True 「True 「True 「True 、 XdRollback 「 XdRollback	DConnectionのTxOpt sconnectActionサフ IRollbackに指定	ionsプロパティ内の ブプロパティを		
0 明示切断時設定の → オブジェクトインスへ FDConnection1 TFDCo → 検索 「プロ/(ティ 【ハント ■ ResourceOptions Tag Transaction ■ TxOptions AutoCommit AutoStart AutoStart AutoStart AutoStart AutoStop DisconnectAction EnableNested Isolation Params ReadOnly	変更点 nnection (TFDTopResourceO) 0 (TFDTxOptions) 『True 『True 『True 『True 『True 』True 『True 』True 』True 』True 』True 』True 』True 』True 』True 』True	DConnectionのTxOpt sconnectActionサン IRollbackに指定	ionsプロパティ内の ブプロパティを		
0 明示切断時設定の → オブジェクトインスへ FDConnection1 TFDCa → 検索 「プロパティ イベント」 ■ ResourceOptions Tag Transaction □ TxOptions AutoCommit AutoStart AutoStart AutoStart AutoStop DisconnectAction EnableNested Isolation Params ReadOnly ● StopOptions	変更点 nnection (TFDTopResourceOT XC 0 (TFDTxOptions) 『True	DConnectionのTxOpt sconnectActionサフ IRollbackに指定	ionsプロパティ内の ブプロパティを		
0 明示切断時設定の → オブジェクトインスペ FDConnection1 TFDCo ● 検索 → プロパティ イペント ■ ResourceOptions Tag Transaction ■ TxOptions AutoCommit AutoStart AutoStart AutoStart AutoStart AutoStart AutoStart AutoStart AutoStart AutoStart StopOptions ■ UpdateOptions UpdateOptions	変更点 nnection (TFDTopResourceOf 0 (TFDTxOptions) ② True ② True ② True ② True 文材Rollback ② True xiReadCommitted (TStrings) ③ False [xoIfCmdsInactive,xoIfA (TFDUpdateOptions)	DConnectionのTxOpt sconnectActionサン IRollbackIに指定	ionsプロパティ内の ブプロパティを		
0 明示切断時設定の → オブジェクトインスペ FDConnection1 TFDCa → 検索 「プロパティ イベント」 ■ ResourceOptions Tag Transaction ■ TxOptions AutoCommit AutoStart AutoStart AutoStart AutoStart AutoStop DisconnectAction EnableNested Isolation Params ReadOnly ● StopOptions UpdateOptions UpdateTransaction	変更点 nnection (TFDTopResourceO) 0 (TFDTxOptions) 『True 』True	DConnectionのTx0pt sconnectActionサン IRollbackIに指定	ionsプロパティ内の ブプロパティを		
0 明示切断時設定の → オブジェクトインスへ FDConnection1 TFDCa → 検索 「プロパティ イベント」 ResourceOptions Tag Transaction TxOptions AutoCommit AutoStart AutoStart AutoStart AutoStart AutoStop DisconnectAction EnableNested Isolation Params ReadOnly ◆ StopOptions UpdateOptions UpdateTransaction 接続エディタ クイック編集	変更点 かタ nnection (TFDTopResourceO) (TFDTxOptions) 「True 」 「True 「True 「True 」 「 」 「 」 「 」 「 」 」 「 」 」 「 」 」 」 「 」	DConnectionのTxOpt sconnectActionサラ IRollbackに指定	ionsプロパティ内の ブプロパティを		

1 TableNam	
	eのメンバー指定
OPT メンハ	
$ \frac{1R11}{X ME}$	F05 18/08/20 <u>ナストのソート順検証用</u> 81 18/08/20 別メンバー参昭の検証田
AMIL	※同一ファイルのメンバー一覧画面
	TR11F05:【図9】のレコードが入っている
	X_MBRI : 今回参照用の別レコートか入っている
💿 オブジェクト	
FDTable1 TFD	Table
プロパティーイベン	
⊕ ResourceOpt	ions (TFDBottomResourceOptions)
SchemaAdap	
» TableName	e BDEのように、TableName BDEのように、TableName アクトン BDEのように、TableName アクトン BDEのように、TableName
Tag	0 OPEN メンバーを指定する事はできない
UpdateObjec	
UpdateOption	
しpdate Trans	[FireDAC][Phys][ODBC][SystemObjects][ClientObjects/400][S0
バインドソースの追	LPrepareCur]カラム修飾子またはテーブルAが未定義である。
すべての項目が表述	촄쳌쳌쳌쳌쳌쳌쳌쳌쳌쳌쳌쳌쳌쳌쳌쳌쳌.
	ок
	こによるシンバー参照
Z6 OVRDB	Fによるメンバー参照
Z6 OVRDB	Fによるメンバー参照
Z6 OVRDB procedure T begin EDTable1	Fによるメンバー参照 Form1.Button4Click(Sender: TObject); Close:
76 OVRDB procedure T begin FDTable1. FDTable1.	Fによるメンバー参照 Form1.Button4Click(Sender: TObject); Close: TableName := 'TR11F05';
76 OVRDB procedure Ti begin FDTable1. FDTable1. AS4001.Rep FDTable1	Fによるメンバー参照 Form1.Button4Click(Sender: TObject); Close; TableName := 'TR11F05'; noteCmd('OVRDBF FILE(TR11F05) TOFILE(TR11F05) MBR(X_MBR1) OVRSCOPE(*JOB)'); Dpen;
76 OVRDB procedure T begin FDTable1. FDTable1. AS4001.Re FDTable1. end;	Fによるメンバー参照 Form1. Button4Click (Sender: TObject); RemoteCmdでネイティブ側からOVRDBFコマンドを 発行し、メンバー指定を行う Close: TableName := 'TR11F05'; noteCmd ('OVRDBF FILE (TR11F05) TOFILE (TR11F05) MBR (X_MBR1) OVRSCOPE (*JOB)'); Open; フーブルのClose後、必要に広じてDLTOVRコマンドを発行し、メンバー指定も紹吟する
76 OVRDB procedure T begin FDTable1. FDTable1. AS4001. Rei FDTable1. end;	Ficよるメンバー参照 Form1. Button4Click (Sender: TObject); TableName := 'TR11F05'; noteCmd ('OVRDBF FILE (TR11F05) TOFILE (TR11F05) MBR (X_MBR1) OVRSCOPE (*JOB)'); Dpen; デーブルのClose後、必要に応じてDLTOVRコマンドを発行し、メンバー指定を解除する AS4001. RemoteCmd ('DLTOVR FILE (TR11F05) OVRSCOPE (*JOB)'):
76 OVRDB procedure T begin FDTable1. FDTable1. AS4001. Re FDTable1. end;	Ficulta Section 4 Click (Sender: TObject); Form1. Button 4 Click (Sender: TObject); Close: TableName := 'TR11F05'; noteCmd ('OVRDBF FILE (TR11F05) TOFILE (TR11F05) MBR (X_MBR1) OVRSCOPE (*JOB)'); Dpen; テーブルのClose後、必要に応じてDLTOVRコマンドを発行し、メンバー指定を解除する AS4001. RemoteCmd ('DLTOVR FILE (TR11F05) OVRSCOPE (*JOB)');
76 OVRDB procedure T begin FDTable1. FDTable1. AS4001. Ren FDTable1. end;	FCLよるメンバー参照 Form1. Button4Click (Sender: TObject); Close; TableName := 'TR11F05'; noteCmd('OVRDBF FILE(TR11F05) TOFILE(TR11F05) MBR(X_MBR1) OVRSCOPE(*JOB)'); Dpen; デーブルのClose後、必要に応じてDLTOVRコマンドを発行し、メンバー指定を解除する AS4001. RemoteCmd('DLTOVR FILE(TR11F05) OVRSCOPE(*JOB)');
76 OVRDB procedure T begin FDTable1. FDTable1. AS4001. Red FDTable1. end;	FicLaSメンバー参照 Form1. Button4Click (Sender: TObject); Close: TableName := 'TR11F05'; moteCmd('OVRDBF FILE(TR11F05) TOFILE(TR11F05) MBR(X_MBR1) OVRSCOPE(*JOB)'); Dpen; テーブルのClose後、必要に応じてDLTOVRコマンドを発行し、メンバー指定を解除する AS4001. RemoteCmd('DLTOVR FILE(TR11F05) OVRSCOPE(*JOB)'):
76 OVRDB procedure T begin FDTable1. FDTable1. AS4001. Re FDTable1. end;	Fによるメンバー参照 Form1. Button4Click (Sender: TObject); Close: TableName := 'TR11F05'; moteCmd('OVRDBF_FILE(TR11F05)_TOFILE(TR11F05)_MBR(X_MBR1)_OVRSCOPE(*JOB)'); Open; テーブルのClose後、必要に応じてDLTOVRコマンドを発行し、メンバー指定を解除する AS4001. RemoteCmd('DLTOVR_FILE(TR11F05)_OVRSCOPE(*JOB)'):
76 OVRDB procedure T begin FDTable1. FDTable1. AS4001. Re FDTable1. end;	Fによるメンバー参照 Form1. Button4Click (Sender: TObject); Close: TableName := 'TR11F05'; moteCmd (' OVRDBF FILE (TR11F05) TOFILE (TR11F05) MBR (X_MBR1) OVRSCOPE (*JOB)'); Open; テーブルのClose後、必要に応じてDLTOVRコマンドを発行し、メンバー指定を解除する AS4001. RemoteCmd (' DLTOVR FILE (TR11F05) OVRSCOPE (*JOB)'):
ス6OVRDBprocedure T begin FDTable1. FDTable1. AS4001. Re FDTable1. end;	Fによるメンバー参照 Form1. Button4Click (Sender: TObject); Close: TableName := 'TR11F05'; moteCmd('OVRDBF FILE(TR11F05) TOFILE(TR11F05) MBR(X_MBR1) OVRSCOPE(*JOB)'); Open: テーブルのClose後、必要に応じてDLTOVRコマンドを発行し、メンバー指定を解除する AS4001. RemoteGmd('DLTOVR FILE(TR11F05) OVRSCOPE(*JOB)');

SHAA10	SHAA11	SHAA12	4		
MEMBER	X NDD1	TESTUT			
MEMBER	X_MBRT	TEST02			
		TEOTOA		と同じファイルだが	
MEMBER	X_WRK1	1E5104	1指定	と回しファイルにか、 たメンバーのレコードが表示される	
		- 177			
X7 ALIASIS	よるメンバー参	影照			
11 17-7	ブルオープンを	に行う処理			
procedure T	Form1. Button	15Click (Sender: T	Object);		
begin					
FDTable1.	Close;				
FDTable1.	TableName :=	= sAliasName;		」 TableNameにエイリアスの名前を指定	
FDTable1.	Open: // こ	こで②が走る		「W他のジョブと重複しない名前を	
end;				予め変数にセットしておく	
11 @=	<i>テぃ</i> +_ ー*・,ಡ	t I N MA		(「00+ショノ番号」形式を推奨)	
nrocedure I	アレオーノン時 Form1 FDTabl	F1 ~~~ le1ReforeOnen (Dat	2eteUT · to2e	et).	
begin		cructor copen (bac	JOCL: IDalao		
FDConnect	ion1. ExecSQL	('CREATE ALIAS '	+ sAliasNam	e + ' FOR TR11F05(X_MBR1)');	
end;					1
		· · · ·		メンバーを指定してエイリアスを生成	
// ③画面ク	ロース時イベ Form1 FormCl	シト Loop (Sondar: TOhi	aat: var hat	ion: T(locolotion):	
hegin	FORINT. FORINGI	ose (sender Tob)	BCL, VAR ACL	TOTIC TOTOSEAGLION),	
FDTable1.	Close: // こ	こで④が走る			
end;				※FormDoctroyでは①がまらたい場合がある	1
				「「「「「」」」、「「」」、「「」」、「「」」、「「」」、「」、「」、「」、「」	
// ④テーフ	ルクローズ時	デイベント	-0-+- TD - 0		
procedure I	FormI. FUIabl	elatterclose (Data	aset. IDataS	et),	
harin	ion1 ExecS0	('DROP ALLAS ' +	sAliasName)		1
begin FDConnect	TOTT: EXCOURE	C DIGI NETNO	onnanoy	エイリアスの削除	
begin FDConnect end:				1	
begin FDConnect end:					

SHAA10	SHAA11	SHAA12	
MEMBER	X_MBR1	TEST01	
MEMBER	X_MBR1	TEST02	
MEMBER	X_MBR1	TEST03	
MEMBER	X MBR1	TEST04	【図22】と同様、指定したメンバーの

株式会社ミガロ.

RAD事業部 営業·営業推進課

[Delphi/400] RESTによるWebサービスを活用 した機能拡張テクニック

はじめに
 REST による Web サービスとは?
 REST 機能を利用する方法
 IBM Watson API 活用方法
 REST 機能をもつコンポーネントの作成
 さいごに



Delphi/400 中級

略歴 1973年8月16日生まれ 1996年3月三重大学工学部卒業 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 2013年4月RAD事業部配属

現在の仕事内容: ミガロ. 製品の素晴らしさをアピー ルするためのセミナーやイベントの 企画・運営等を主に担当している。

1.はじめに

近年アプリケーションの開発におい て、Web サービスの活用が盛んになっ ている。Web サービスとは、インター ネット技術を応用し、他のWeb サイト 上のソフトウェアを呼び出して利用する 仕組みのことである。【図 1】

現在では、大量のデータを蓄積してい る業者等が、そのデータを Web サービ スの形で一般のユーザーやプログラマー に提供する事例が多くなっている。たと えば、ネットショッピングサイト大手の アマゾンには「Product Advertising API」、楽天には「楽天市場商品検索 API」といった Web サービスがあり、 これら Web サービスを利用すると、サ イト上の商品検索等を自分のプログラム に組み込むことができる。さらに近年で は、従来の大量データをもつ業者だけで なく、IBM Watson のような自然言語 を理解し、機械学習により人間の意思決 定を支援するシステムまでもが、Web サービスとして利用可能になっている。

Web サービスは、インターネット技 術を使用するのが特徴だが、その手法に はいくつかあり、代表的なのが SOAP (Simple Object Access Protocol) お よび REST (REpresentational State Transfer) である。

SOAP は、SOAP メッセージという XML によってメッセージ交換を行う方 法で事前にやり取りの定義が必要なた め、難易度が高い。最近は、よりシンプ ルな REST が主流である。本稿では、 Delphi/400 を使用した REST による Web サービスの使用方法や機能拡張方 法について説明する。なお、本稿のプロ グラムは Delphi/400 10 Seattle 以降の 環境を前提としている。

2.RESTによる Webサービスとは?

REST とは、Web サービスの設計モ デルのことで、ネットワーク上のデータ (リソース)を一意な URL で表すもの である。サービスの URL に HTTP メ ソッドでアクセスすることでデータの送 受信が行える。パラメータを指定して URL にアクセスすると特定の形式で データが返ってくるものだ。データ形式 には、XML あるいは JSON が利用可能 であるが、近年は、よりシンプルな JSON が使われることが多い。

JSON と は、JavaScript Object Notation の略で、軽量のデータ交換 フォーマットのことである。key (名前) と value (値) を「:」で対にして記述し、 まとまりごとに{}で囲うといった表記 法で、[]で配列を表現することもできる。 たとえば、【図2】のような JSON は、 "result" というキーの配列の第一要素の 中にある "score" というキーの値が 80 であると解釈できる。

では、ここで REST + JSON による Web サービスを試してみる。livedoor が提供する「お天気 Web サービス」を 使ってみる。

http://weather.livedoor.com/weather_ hacks/webservice



(Google で "お天気 Web サービス"
 を検索すると上位に表示される「お天気
 Web サービス仕様」。)

このサービスは、現在全国142カ所 の今日・明日・明後日の天気予報・予想 気温と都道府県の天気概況情報を提供す るものである。

Chrome ブラウザを立ち上げてアドレ ス欄に

[http://weather.livedoor.com/forecast/ webservice/json/v1?city=130010]

と入力してアクセスを行う。すると、【図 3】のような JSON が表示される。

これは、地域 ID=130010(東京都) の天気予報情報にアクセスした結果の JSON である。天気や最高気温などの情 報が JSON の中に含まれていることが わかる。このように REST + JSON に よる Web サービスは、URL にパラメー タを付けて呼び出すとレスポンスとして JSON データが返ってくることが確認で きる。

3.REST機能を 利用する方法

では、この「お天気 Web サービス」 を Delphi/400 から使用する方法を検討 する。Delphi/400 には、REST による Web サービスを使用するためのコン ポーネントが用意されている。それが、 「TRESTClient」、「TRESTRequest」そ して「TRESTResponse」である。【図 4】 「TRESTClient」は、Web サービス へのリクエストを実行するコンポーネン トで、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 け取るものである。「TRESTRequest」 は、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する。 「TRESTResponse」は、Web サービス からのすべての戻りデータを保持する。 実際の設定は次のようになる。【図5】

ポイントは、RESTClient1 コンポー ネントの BaseURL プロパティに WEB サービスの基底 URL を指定すること、 RESTRequest1 コ ン ポ ー ネ ン ト の Method プロパティに HTTP メソッド の種類を、Resource プロパティに実行 パラメータを指定することである。

このプログラムの[検索]ボタンク リック時の処理は、【ソース1】となる。

1行目は、画面上で指定した地域 ID を Resource プロパティに記した "{CITY}"にセットする処理である。 Params プロパティの AddItem メソッ ドがリクエストのパラメータを定義する メソッドである。2行目は、Web サー ビスへのリクエスト実行になり、レスポ ンスの JSON 文字列を取得して Memol にセットするのが、3行目である。実際 に実行した結果が【図6】となる。

REST による Web サービスによって JSON データが取得できることを確認し たが、実際にはこの JSON データをパー ス (解釈) し必要な情報を抜き出す必要 がある。Delphi/400 にはこの JSON を 取り扱うためのユニットが用意されてい る。それが、"System.JSON"ユニットで ある。この中に、JSON オブジェクトを 実装したクラス TJSONObject や、文 字列、数値、オブジェクト、配列、 true/false の型を持つすべての JSON クラスの上位クラス TJSONValue が用 意されているので、これらを使用するこ とでパースすることが可能である。

たとえば、【図 7】が JSON をパース して特定のキーの値を取得するロジック 例である。

お天気情報の中から、今日と明日と明 後日の天気を取得して表示する処理を実 装してみる。

「お天気 Web サービス」の説明 Web ページを確認すると、レスポンスの仕様 が記載されている。[forecasts] プロパ ティの中にある [date] プロパティが予 報日、そして [telop] プロパティが天気 である。なお、3日分のデータは配列と して定義されている。この情報をもとに JSON をパースして、天気予報を画面に 表示するように改良したのが、【ソース 2】である。完成したプログラムを実行 すると、【図 8】のように指定した地域 の3日分の天気予測を取得することがで きる。

もう1つWebサービスの例を紹介す る。「HeartRails Geo API」(http:// geoapi.heartrails.com/)である。これは、 郵便番号/住所/緯度経度データ等の地 理情報を提供するWebサービスである。 この中に、「最寄駅情報取得 API」とい う機能があり、これは郵便番号を指定す ると、その地区の一番近い最寄駅がわか る機能である。これを活用すると、たと えば社内の取引先マスタにある郵便番号 を使用して、同じ最寄駅から歩いて訪問 できる取引先をピックアップする使い方 ができる。

サンプルプログラムを紹介する。 REST コンポーネントの設定は、【図 9】、 プログラムは【ソース 3】となる。この API は、パラメータ postal に郵便番号 をセットし呼び出すと、レスポンスとし て、response プロパティの station 配 列の中にある prefecture プロパティに は都道府県が、line プロパティに路線が、 そして name プロパティには最寄駅が セットされる。作成したプログラムを実 行して、郵便番号を入力し、[検索]ボ タンをクリックすると、最寄駅が表示さ れることがわかる。【図 10】

この最寄駅情報取得 API も、先ほど の天気情報の Web サービスと全く同じ やり方で処理ができることがわかる。こ のように、REST + JSON の Web サー ビスは、とても簡単に使用できるので、 いろいろなサービスを試してみていただ きたい。本稿執筆にあたり、Web サー ビスを調査したが、「API List 100+」 (http://smsurf.app-rox.com/api/) と いうサイトが役立った。ここにはいろい ろな Web サービスが一覧掲載されてい るので、便利な機能を見つけてみてほし い。【図 11】

4.IBM Watson API 活用方法

「コグニティブ」という言葉を聞いた ことがないだろうか。日本語では「認知」 のことで、ある事象についてコンピュー タが自ら考え、学習し、自らの答えを導 き出すシステムのことをいう。身近なと ころでは、iPhone の Siri やスマートス ピーカー等が有名である。従来システム との本質的な違いは、音声・画像・文章 等の非定型データも処理できることであ る。従来システムがもつ定型データと組 み合わせることで、人の作業を補助し、 より便利なシステム構築が可能になる。 この「コグニティブ」分野で IBM が提 供するのが、Watson である。この



Watson も Web サービスとして利用す ることができるので本章で説明する。

Watson は、IBM のクラウドサービ ス(IBM Cloud)上で API として提供 されており、クラウドサービスに登録す れば誰でも使用できる。本格的な利用に は有償プランが必要だが、IBM Cloud の各種サービスを無料で使用できる「ラ イト・アカウント」があるので、こちら を使用するとよい。「ライト・アカウント」 には、サービス使用量や機能の制約はあ るが、クレジットカード登録不要で簡単 に登録できる。

https://www.ibm.com/cloudcomputing/jp/ja/bluemix/lite-account/ (Google で"IBM Cloud ライト・ア

カウント"を検索すると上位に表示される「IBM Cloud ライト・アカウント -Japan」。)

2018 年 8 月時点で確認した Watson の API が【図 12】である。調べたすべ てのサービスが「ライト・アカウント」 で使用可能だ。

今回は、Language Translator サー ビスを例に説明する。これは言語変換(翻 訳)サービスで、特長としては、一般的 なWEB翻訳サービスと違い、専門用語 等を個別に登録することや、機械学習に よるカスタム翻訳モデルを作成すること が可能で、高精度の翻訳が行えることで ある。このサービスを活用すると、海外 担当者とのやり取り時の自動翻訳や、シ ステムの多言語対応で、DB上に日本語 でしか保持していない情報を翻訳して、 画面に出力することができる。

Watson は、サービスごとにインスタ ンスの作成が必要である。まず IBM Cloud (https://console.bluemix.net/) からサインインを行い、表示されたダッ シュボード画面より、[リソースの作成] ボタンをクリックする。Watsonをはじ めとするサービス選択画面が表示される ので、[Language Translator]を選択す る。サービス概要画面が表示され、プラ ンの選択ができるようになるため、「ラ イト」が選択されていることを確認し、 [作成]をクリックすれば、完了である。 インスタンスの作成が完了すると、作成 したサービスの管理画面が表示される。 【図 13】 【図 13】の管理画面の中にある「資格 情報」が重要である。2018年8月現在サー ビスによって「資格情報」には2種類あ り、ユーザーとパスワードが表示される サービスと、API Key が表示されるサー ビスとがある。Language Translator サービスの場合、API Key が表示され るため、この API Key と URL を控え ておけばよい。

また、管理画面には「APIリファレ ンス」画面へのリンクがあり、そこにア クセスすれば APIの仕様が記載されて い る。2018 年 8 月 現 在 Language Translator サービスは、V3 というバー ジョンになっており、APIの仕様は、【図 14】のとおりである。

Delphi/400から利用するポイントを 説明する。今回は、画面上に日本語で入 力したテキストを英語に翻訳するアプリ を作成する。(英語から日本語への翻訳 も可能にする。)

WatsonAPI の場合、資格情報が必要 なため、認証が必要である。 WatsonAPI では、基本認証を使用する ことができ、これは「THTTPBasic Authenticator」コンポーネントが使用 できる。

REST コンポーネントの設定は【図 15】のとおりである。

サービスの資格情報がユーザーとパ スワードの場合は、そのまま 「THTTPBasicAuthenticator」コンポー ネントのUsername プロパティと Password プロパティにセットすればよ く、APIKeyの場合は、Username プ ロパティに "apikey"、Password プロパ ティに資格情報のAPIKey を入力すれ ばよい。

今回のアプリの画面レイアウトは、 【図 16】、プログラムは【ソース 4】の ようになる。この Web サービスは、 POST メソッドとなり、リクエスト本 体に JSON 形式でパラメータを渡すと ころがポイントである。

実際に実行したアプリケーションが、 【図 17】である。Watson API を利用す るアプリケーションも REST + JSON で簡単に構築できることがわかる。

5.REST機能をもつコ ンポーネントの作成

今回、REST + JSON を使用した Web サービスの活用方法について具体例を挙 げながら説明したが、Web サービスの 課題点は、サービス提供者の都合により、 サービスの仕様が変更されたり、サービ ス自体が終了してしまう可能性があるこ とだ。

ある Web サービスを活用したアプリ ケーションを使用していた場合に、この ような事態が発生すると、新しい仕様に あわせてプログラムを変更したり、ある いは代替サービスに置換したりといった 作業が必要になる。こういったことを想 定した場合、Web サービスの機能を個々 のプログラムに都度記述する方法だと、 修正ボリュームが多大になることが想像 できる。

また、プログラムの中から Web サー ビスの部分を抜き出して修正しなければ いけないため、煩雑な作業になることが 予想される。

この問題を解決するには、どうすれば よいか? 1つの方法が各 Web サービ スごとにコンポーネント化してしまうこ とである。そうすれば、Web サービス の仕様変更時にも、コンポーネントソー スのみ修正し、各プログラムは、リコン パイルだけすれば済むはずである。

今回は、コンポーネント化の例とし て、 前 章 で 使 用 し た Language Translator サービスのコンポーネント 化を検討する。(TComponent を継承し た TTranslator コンポーネント (非ビ ジュアルコンポーネント)を作成する。)

本稿では、コンポーネントそのものの 基本的な作成手順については割愛する が、作成手順が分からない場合は、2012 年度版『ミガロ.テクニカルレポート』 のSE 論文「カスタマイズコンポーネン ト入門」を参照していただきたい。

宣言部は、【ソース5】となる。変換 元の言語(SourceLanguage プロパティ) と変換後の言語(TargetLanguage プ ロパティ)、そして変換対象の文字列 (Source プロパティ)を設定した後、 Translate メソッドを実行すると翻訳が 行われ、その結果は、Destination プロ パティにセットされるという仕様を想定 している。

ノース1 柞	検索ボタンのOnClick	イベント		
uses	s REST.Types;	, , ,		
pro	cedure TForm1	.Button1Click(Sende	r: TObject):	
beg		4 0 世白		
RE	ESTRequest1.P	タの疳 <i>正</i> 'arams.AddItem('CITY	'. Edit1.Text. pkURLSE	GMENT):
	11	2.4=		
RE	/リクエスト美 ESTRequest1.E	17 xecute:		
	11.7 2 2.7.15	SON 本 志 テ		
Me	emol.Text :=	RESTResponse1.JSONT	ext;	
end				
図6 お天	気情報取得実行例			
🛞 For	m1		- 🗆 ×	
t	也域ID 130010	検索		
Г	"link":"http://we	ather livedoor.com/area/for	recast/130010".	
	"forecasts":			
	[
	ر "dateLabel":"	'今日",		
	"telop":"晴の*	5曇",		
	"temperature	-08-25", a":		
	{	10.1.		
	"min":null, "max":			
	{			
	4		~	
L				
⊠7 JSO	N パース例			
変数sR	et の値	procedure TForm1 Rutton10	Click(Sender: TObiect):	
"resu	lt":[Var	anakionidan, lobject),	
Ċ	"subject":"english",	begin $//JSON = - 2 m x^2 - 2$		
	"state":"good"	JSONValue := TJSONObjec //結果の取得	ct.ParseJSONValue(sRet);	
3,		ShowMessage(JSONValue.(GetValue< string >(' <u>result[0].su</u>	bject'));
	"subject":"math",	皇行 法里	Project1	
	"score": 70,			×
3	"score": 70, "state":"normal"		english	×
}	"score": 70, "state":"normal"		english	× OK

ス7] となる。コンポーネントの生成時 (Create メソッド) において、内部的に REST コンポーネントを生成し、 Language Translator サービスの仕様 に基づいたパラメータの設定を行ってい る。あとは、Translate メソッドにて、 【ソース4】と同様の変換処理を行って いる。 このコンポーネントを使用したサン プルプログラムは、とてもシンプルであ る。画面レイアウトは、【図18】、プロ

実装部は、【ソース6】および【ソー

る。画面レイアウトは、【図 18】、プロ グラムは【ソース 8】である。Web サー ビス自体をコンポーネント化しているた め、API の仕様部分はこのプログラム には含まれていないことがわかる。これ によって、将来 Web サービスが終了し ても、コンポーネントの内容を別の Web サービスに変更すれば、個々のプ ログラムを変更する必要がなくなり、耐 性の強いプログラムであることがわかる だろう。

6.さいごに

本稿では、REST による Web サービ スを活用した Delphi/400 の機能拡張と して、いくつかの Web サービスを使用 した具体例を紹介してきた。REST + JSON 方式が簡単に Delphi/400 から活 用できることがわかる。単純に REST コンポーネントを組み込むだけでも十分 活用できるが、コンポーネント化まで検 討することにより、より耐性の強い仕組 みが作れるのである。ぜひ本稿を参考に いろいろな Web サービスの活用をご検 討いただきたい。

Μ



```
図8 お天気情報取得実行例2
```

Form1			-	×
地域ID 130010	検	索		
今日	明日	明後日		
2018-08-25	2018-08-26	2018-08-27		
晴のち曇	晴時々曇	晴のち曇		



都道府県	铬線	最寄り駅	
東京都	東京火トロ日比谷線	霞ケ関	
← → C ④ 保護されていない通信 sm	surf.app-rox.com/api/	☆ × 0 0 Ⅲ I	1 0 0 1
← → C ④ 保護されていない通信 sm	surf.app-rox.com/api/	☆ × ⊖ 0 回 I	1 (0) (0) 1
海外·国内 API一覧	API LIST 100+	G- 3141 14 UNUN2 1 122	/√−► B1734
<u>海外・国内 API一覧</u> 公開され おも	API LIST 100+ ている気になるAPI/Webサービスをリスト しろそうなAPIを見つけたらつど追加してい	G- 用作 (4-14-121 92) (1-14 としてみました)きますね	и с ој : И с ој :
海外 - 国内 API一覧 公開され あも の Developer Center	API LIST 100+ ている気になるAPI/Webサービスをリスト しろそうなAPIを見つけたらつど追加してい	G- RH (* 10021 22) メジ としてみました 注ますね の API キーワード検索	2 C II :
海外 - 国内 API一覧 公開され あも の Developer Center <u>Google Developers</u> 提供 : Google	API LIST 100+ ている気になるAPI/Webサービスをリスト しろそうなAPIを見つけたらつど追加してい	G- 用用 (# 100021 92) 99 としてみました きますね ク API キーワード検索 キーワードを1回で 検索	P4-1-1 B1734
海外 - 国内 API一覧 公開され おも	APILIST 100+ ている気になるAPI/Webサービスをリスト しろそうなAPIを見つけたらつど追加してい を提供。Google Play、Google+、 id などのAPIも多数公開	G- 用料 (# 1992) 92 98 としてみました いきますね ク API キーワード検索 キーワードを1回で 検索	
海外 - 国内 API一覧 公開され おも	APILIST 100+ ている気になるAPI/Webサービスをリスト しろそうなAPIを見つけたらつど追加してい を提供。Google Play、Google+、 id などのAPIも多数公開	G- 用用 (# 1992) 92 98 としてみました きますね の API キーワード検索 キーワードを1回で 被集 広告は Google により した	A BIZ23 PA-ト BIZ23 D終了しま
 海外・国内 API一覧 公開され、 あも Developer Center Google Developera 提供: Google Android, iOS, Web環境での開発キット Maps, YouTube, Books, Gmail, Clou Google Cloud 提供: Google コンピューティング、ストレージ・デー/ ラから機械学習による分析まで様々な職 	APILIST 100+ ている気になるAPI/Webサービスをリスト しろそうなAPIを見つけたらつど追加してい や提供。Google Play、Google+、 dd などのAPIも多数公開 タペース、ネットワークといったインフ Eを提供	G- 用用 (* 1992) 92 99 としてみました きますね ク API キーワード検索 キーワードを1回で 被集 広告は Google により した この広告の表示を付	A BIZ31 PY-ト BIZ31)終了しま 弊止
<u>海外・国内 API一覧</u> 公開され あも Developer Center <u>Google Developers</u> 提供: Google Android, iOS, Web環境での開発キット Maps, YouTube, Books, Gmail, Clou <u>Google Cloud</u> 提供: Google コンピューティング、ストレージ・デー ラから機械学習による分析まで様々な職 <u>Microsoft Azure</u> 提供: Microsoft	APILIST 100+ ている気になるAPI/Webサービスをリスト しろそうなAPIを見つけたらつど追加してい を提供。Google Play、Google+、 od などのAPIも多数公開 タベース、ネットワークといったインフ 底を提供	G- ## ゆ いいむ 22 9 20 としてみました きますね ク API キーワード検索 キーワードを1回で 被集 広告は Google により した この広告の表示を2	A/(-ト BI77) P/(-ト BI77) の終了しま 学止
 ⁽⁾ (AP) - 国内 AP) - 国 ⁽⁾ (AP) - 国 ⁽⁾ Developer Center ⁽⁾ (Developer Letter) ⁽⁾ (Developer 提供: Google Android, iOS, Web環境での開発キット Maps, YouTube, Books, Gmail, Clou Google Cloud 提供: Google Dンピューティング、ストレージ・デー ラから機械学習による分析まで様々な機 Microsoft Azure 提供: Microsoft コンピューティング、ストレージ・デー ラから機械学習による分析まで様々な機	APILIST 100+ ている気になるAPI/Webサービスをリスト しろそうなAPIを見つけたらつど追加してい やを提供。Google Play、Google+、 od などのAPIも多数公開 タペース、ネットワークといったインフ 能を提供	G- 用用 (* 1992) 92 99 としてみました きますね ク API キーワード検索 キーワードを1回で 残気 広告は Google により した この広告の表示を作 広告表示設定 0	A G of : P(-ト B173) ↑ D終了しま P止 D
 高外・国内 API一覧 公開され、 あも Developer Center Google Developers 提供: Google Android, iOS, Web環境での開発キット Maps, YouTube, Books, Gmail, Clou Google Cloud 提供: Google コンピューティング、ストレージ・デー/ ラから陽城学習による分析まで様々な職 Microsoft Azure 提供: Microsoft コンピューティング、ストレージ・デー/ ラから陽城学習による分析まで様々な職 Bing for partners 提供: Microsoft 	APILIST 100+ ている気になるAPI/Webサービスをリスト しろそうなAPIを見つけたらつど追加してい やを提供。Google Play、Google+、 id などのAPIも多数公開 タペース、ネットワークといったインフ 影を提供 タペース、ネットワークといったインフ 影を提供	G- ## ゆ いいむ 22 99 としてみました きますね ク API キーワード検索 キーワードを1道で 検索 広告は Google により した この広告の表示を1 広告表示設定 0	PAGe Top
 海外・国内 API一覧 公開され、 あも Developer Center Google Developers 提供: Google Android, iOS, Web環境での開発キット Maps, YouTube, Books, Gmail, Clou Google Cloud 提供: Google コンピューティング、ストレージ・デー・ ラから機械学習による分析まで様々な職 Microsoft Azure 提供: Microsoft コンピューティング、ストレージ・デー・ ラから機械学習による分析まで様々な職 Microsoft Azure 提供: Microsoft コンピューティング、ストレージ・デー・ ラから機械学習による分析まで様々な職 Bing for partners 提供: Microsoft 地回、音声、翻訳、検索、Web管理、広い 	APILIST 100+ ている気になるAPI/Webサービスをリスト しろそうなAPIを見つけたらつど追加してい ・を提供、Google Play、Google+、 id などのAPIも多数公開 タペース、ネットワークといったインフ 能を提供 BAズーのAPIを公開	G- 用用 ゆ いいな1 92 99 としてみました いきますね ク API キーワード検索 キーワードを1度で 検索 広告は Google により した 広告表示設定 0	PYT-ト B1229 PXT-ト B1229 の終了しま 停止 Page Top

73 M	API種類	機能	1711
照会応答系	Assistant (照会応答)	自然言語で対話可能なアプリケーションを、シンプルな開 発ツールで迅速に構築	対応
言語系	Language Translator (言語変換)	コンテンツのテキストを、ある言語から別の言語にリアル タイムで翻訳	対応
	Natural Language Understanding(自然言語理解)	テキスト分析を行い、コンテンツから概念、エンティ ティー、キーワード、カテゴリー、感情、関係、意味役割 などのメタデータを抽出	対応
心理系	Personality Insights (性格分析)	テキストから筆者のパーソナリティ(ビッグ・ファイブ、 価値、ニーズ)の3つの特徴を推測	対応
音声系	Speech to Text(音声認識)	ディープ・ラーニングを活用して、音声からテキストを書き起こす	対応
	Text to Speech(音声合成)	テキストから自然な音声を合成	対応
知識探索系	Discovery(検索)	大量のデータを検索するとともに、適切な意思決定を支援	対応
	Knowledge Studio	業界や分野ごとの知識だけでなく、各分野の言葉の使われ 方の微妙な違いをWatsonに教えることができるツール	対応
画像系	Visual Recognition(画像認識)	ディープ・ラーニングを使用して、画像に写った物体・情景・顔など様々なものを分析・認識	対応
3 Language	Translatorサービス管理画面		
3 Language	Translatorサービス管理画面	θ	- 0
3 Language	Translatorサービス管理画面 ス・ヘー: × ここ ここで、またまたまたまで、 https://console.bluemix.net/sen	ces/language-translator/cm%3Av1%3Abluemix%3A 🕶 🖈 🖂 🥥 🔯 📘	
3 Language	Translatorサービス管理画面 ス・ヘー: × 護された通信 https://console.bluemix.net/sen カタログ 資料	e vices/language-translator/cm%3Av1%3Abluembc%3A ♥ ☆ ズ @ (2) 図 サポート 管理 Q 尾崎浩司's Acc	- D Sount O
3 Language ○ IBM Watson サービ ← → C ● 保 ■ IBM Cloud 管理	Translatorサービス管理画面 ス・ペー:× 護された遺信 https://console.bluemix.net/serv カタログ 資料 Language Translator /	e vices/language-translator/cm%3Av1%3Abluemix%3A ♀☆ ☆ < @ @ 図] サポート 管理 Q 尾崎浩可すAcc Watson / >Translator-wl	- D oount O
3 Language ○ IBM Watson サービ ← → C ● 保 ■ IBM Cloud 管理 サービス資格領報	Translatorサービス管理画面 ス・ペー: × 置された通信 https://console.bluemix.net/serv カタログ 資料 Language Translator / (デ) Language リソース・グループ: D	・ vices/language-translator/crn%3Av1%3Abluemix%3A ♀☆ ☆ ズ ④ ジ 図 サポート 管理 Q 尾崎浩可すAcc Watson / e Translator-wl Default 場所:米国南部	- D Sount O
3 Language ○ IBM Watson サービ ← → C ● 保 ■ IBM Cloud 管理 サービス資格情報 ブラン 論述	Translatorサービス管理画面 ス・ペー: × 置きれた通信 https://console.bluemix.net/serv カタログ 資料 Language Translator / ② Language リソース・グループ: 0	e vices/language-translator/crm%3Av1%3Abluemix%3A ♀☆ ☆ ズ @ ジ 図 サポート 管理 Q Re終治可すAcc Watson / e Translator-wl Default 場所:米国南部	- D Sount O
3 Language ○ IBM Watson サービ ← → C ■ 保 ■ IBM Cloud 管理 サービス資格情報 プラン 接続	Translatorサービス管理画面 スペー: × 歴された通信 https://console.bluemix.net/serv カタログ 資料 Language Translator / ② Language リソース・グループ: o	e service. e service. e Service.	- C ount O
3 Language ○ IBM Watson サービ ← → C ■ 保 ■ IBM Cloud 管理 サービス資格情報 プラン 接続	Translatorサービス管理画面 スペー: × 置された通信 https://console.bluemix.net/serv カタログ 資料 Language Translator / ② Language リソース・グループ: D Get started with the チュートリアル	e service. ● */ces/language-translator/crn%3Av1%3Abluemix%3A ◆☆☆ ② ② ② ② サポート 管理 Q 尾崎浩司's Acc ② * * * * * * * * * * * * *	- C Sount O
3 Language ⊂ IBM Watson サービ ← → C ■ 保 ■ IBM Cloud 管理 サービス資格情報 プラン 接続	Translatorサービス管理画面 ス・ペー: × 置された遺信 https://console.bluemix.net/serv カタログ 資料 Language Translator / ② Language リソース・グループ: D Get started with the チュートリアル	・ vices/language-translator/crm%3Av1%3Abluembc%3A ♀☆☆ く ② ⑦ 図 サポート 管理 Q 死ぬ治司's Acc Watson / e Translator-wl Default 場所:米国南部 e service. プラン:Lite アップグレー	- C Sount Q
3 Language ○ IBM Watson サービ ← → C ● 保 ■ IBM Cloud 管理 サービス資格情報 プラン 接続	Translatorサービス管理画面 スペー:× 置された遺信 https://console.bluemix.net/sen カタログ 資料 Language Translator / ① Language リソース・グループ: ロ Get started with the チュートリアル 資格情報	e service. APIリファレンス ● #EEEE @ 100000000000000000000000000000000	- C Sount O
3 Language ○ IBM Watson サービ ← → C ■ 保 ■ IBM Cloud 管理 サービス資格領報 プラン 接続	Translatorサービス管理画面 ス・ペー: × 置された通信 https://console.bluemix.net/serv カタログ 資料 Language Translator / () Language リソース・グループ: 0 Get started with the チュートリアル 資格情報	● rices/language-translator/cm%3Av1%3Abluemix%3A	
3 Language ○ IBM Watson サービ ← → C ■ 保 ■ IBM Cloud 管理 サービス資格情報 ブラン 接続	Translatorサービス管理画面 スペー: × 置きれた通信 https://console.bluemix.net/serv カタログ 資料 Language Translator / ② Language リソース・グルーブ: 0 Get started with the チュートリアル 資格情報 API Key: sa(e service. APIリフェレンス CYB7u(
3 Language ○ IBM Watson サービ ← → C ● 保 ■ IBM Cloud 管理 サービス資格情報 プラン 接続	Translatorサービス管理画面 スペー:× 置された遺信 https://console.bluemix.net/serv カタログ 資料 Language Translator / ② Language リソース・グループ: 0 Get started with the チュートリアル 資格情報 API Key: sat Urt: htt	e service. ブラン: Lite アップグレー APIリフェレンス 原表云 Ø CYB7u(PQ769MxzH	

POST /v3/tr	anslate		※AP11工体の	Diff 袖は、API」	リノアレノ人を参照	
パラメータ	/	タイプ	1	内容		
version		URLパラメータ	7	V3公開日((固定值:2018-	05-01)
request		リクエストの本	《体	翻訳に必要	な下記パラメーク	タをJSON形式で指定
text		string []		翻訳したい	元のテキストを打	指定
source		string		元テキスト	の言語を指定 (j	a,en)
target		string		翻訳後の言	語を指定 (ja,er	ı)
レスポン	ス例		パラメー	-9	内容	
{ "translat	ions" :[{		translat [transla	ions: tion]	翻訳結果のう	テキスト
"transla	ition" : "Hi,"		word_c	ount	元テキストの	D単語数
}], "word c	ount" :1.		charact	er_count	元テキストの	D文字数
Charact	er_count":5 Translatorサー THTTPBasicAuthenticator	-ビス REST設な	定 ■		パースプジェクトインスペク RESTRequest TRESTR	9 – 🗆 X equest
"charact } 5 Language パオブジェクトインスペクタ HTTPBascAuthenticator1 ク検索 ブロパティ イベント1 el bindSource BlindSource BlindSource Name	er_count":5 Translatorサー THTTPBasicAuthenticator IPBasicAuthenticator1.Bind iBinding デザイナ 110505404thenticator1	-ビス REST設た Source	定 IPBasicAuthenticat	or 1	メナリシント イクスペク RESTRequest1 TRESTR ク 検索 プロパティ イベント Accept Accept Accept Accept Accept Accept Accept	7 Capest Capplication/json, text/plain; q=0.9, text/1 UTF-8, *;q=0.8
"charact } 5 Language メデオブジェクトインスペクタ HTTPBaskAuthenticator の検索 ブロパティ イベント1 6) BindSource 日 LiveBinding デザイナ Liv Password Tag 0 Username api	er_count":5 Translatorサー THTTPBasicAuthenticator BlasicAuthenticator1.Blad eBinding デザイナ 1985956Authenticator1 YB7uQca	-ビス REST設た Source	定 TPBasicAuthenticat	RESTRequest1	 ボガジェクトインスペク RESTRequest1 TRESTR D 検索 プロパティ イベント Accept AcceptCharset AcceptCharset AcceptEncoding AutoCreateParams BindSource Client HandleRedirects LiveBinding デザイナ 	ターロン tquest application/json, text/plain; q=0.9, text/1 ヘ UTF-8, *;q=0.8 日 False RESTReques RESTClient1 図 True LiveBinding デザイナ
"charact } 5 Language #TPBaicAuthenticator 2 #TTPBaic	er_count":5	-ビス REST設な	定 TPBasicAuthenticat	RESTRequest 1	※ オブジェクト イクスペク RESTRequest1 TRESTRA ク 検索 ブロパティ イベント Accept AcceptChaset Accept	ターローン equest application/json, test/plain; q=0.9, test/1 ~ UTF-8, ";q=0.8 日 False RESTRequest RESTClient1 図 True LiveBinding デザイナ FRSTRequest1 POST
"charact } 5 Language ATDEDE 477899 HTTPBasicAuthenticator ク検索 プロパティ イベントI BindSource BindSource BindSource UveEnding デザイナ UveEnding デザイナ I Dessword Tag 0 UveEname api	er_count":5	-ビス REST設な Source	定 TPBasicAuthenticat	RESTRequest 1	 ボガジェクト・インスペク RESTRequest1 TRESTRA 検索 プロパティ イベント Accept AcceptCharset AcceptCharset AcceptCharset BindSource Client HandleRedirects LiveBinding デザイナ Method > Name Params Resource Resource Resource 	9 - ・ × tquest ・ ・ application/json, test/plain; q=0.9, test/1 ・ UTF-8, *;q=0.8 日 False RESTRequest RESTClient1 日 True LiveBinding デザイナ rmPOST RESTRequest1 (TRESTRequestParameterList) v3/translate?version=2018-05-01 RESTResponse1 RESTResponse1
"charact } Charact } Charact) Charach	er_count" : 5 Translatorサー THTTPBasicAuthenticator IPBasicAuthenticator1.Bind E8inding デザイナ IPBosicAuthenticator1 YB7uQca	-ビス REST設た Source	定 IPBasicAuthenticat	RESTRequest1	 パースプジェクトインスペク RESTRequest1 TRESTRA ク 検索 プロバテイ イベント1 Accept AcceptCharset AcceptCharset AcceptCharset BindSource BindSource Client HandleRedirects LiveBinding デザイナ Method Name Params ResourceSuffix Response SynchronizedEvents メーオプジェクト・インスペクク 	9
"charact } Charact } Charact) Char	er_count" : 5	-ビス REST設た Source	定 TPBasicAuthenticat	or1 RESTRequest1 んし RESTResponse1	 ボガジシクトインスペク RESTRequest1 TRESTR ● 総添 ブロバティ イベント1 Accept AcceptCharset AcceptCharset AcceptCharset BindSource Client HandleRedirects Client HandleRedirects Client Params Resource Resource Resource Resource Response SynchronizedEvents ボガジェクトインスペク/ RESTResponse1 TRESTR ● 総添 	P equest
"charact } Language メオジェクトインスペクタ HTTPBaicAuthenticator ク検索 プロパティ(イベント) BindSource HTTPBaicAuthenticator プロパティ(イベント) BindSource Username コロ Karpicological JDI/ディ(イベント) Name Tag Username コロ メオブジェクトインスペクタ RESTClient1 アレディイベント) ※ Accept Charset AcceptCharset AcceptCharset AcceptCharset AcceptCharset AutoCreateParams BaseUR	er_count" : 5	-ビス REST設な Source	定 IPBasicAuthenticat	or1 RESTRequest1 人心 RESTResponse1 実行パラメー	 メオブジェクトインスペク RESTRequest1 TRESTR ● 検索 ⑦プロパティ イベント Accept Accept AcceptCharset AcceptCharset AcceptCharset BindSource BindSource Client HandleRedirects EliveBinding デザイナ Method Name Params ResourceSuffix Response SynchronizedEvents メオジェクト・インスペク RESTResponse1 TRESTR ⑦ 樹家電 ⑦プレディイベント ③ BindSource Content ContentEncoding 	Post Po
Charact } Charact } Charact } Charact } Charact Ch	er_count":5	-ビス REST設た Source	TPBasicAuthenticat	or1 RESTRequest1 RESTResponse1 実行パラメー 基底URL	 メオウェクト インスペク RESTRequest1 TRESTRA 検索 ブロバテイ イベント Accept AcceptCharset AcceptCharset AcceptCharset AcceptCharset BindSource Client HandleRedirects LiveBinding デザイナ Method Name Params Resource SynchronizedEvents メオフジスクスクク RESTResponse1 TRESTR DiadSource Content Content ContentEncoding ContentType LiveBinding デザイナ Name 	Post Post


ja ~	
今日の天	に気は曇り時々晴れです。明日は快晴になるでしょう。
00 v	· · · · · · · · · · · · · · · · · · ·
Today's	weather is cloudy and sometimes sunny. It will be fine
tomorro	w.
	~
5 TTr	anslatorコンポーネントの宣言部
5 TTra	anslatorコンポーネントの宣言部
5 TTra nit Tran	anslatorコンポーネントの宣言部 nslator; e REST関連のユニット、JSONユニット
5 TTra nit Tran nterface System REST.Au	anslatorコンポーネントの宣言部 nslator; e REST関連のユニット、JSONユニット .SysUtils, System.Classes, <u>System.JSON, RESI.Client.</u>
5 TTra nit Tran nterface System <u>REST.A</u> ype	anslatorコンポーネントの宣言部 nslator; e REST関連のユニット、JSONユニット .SysUtils, System.Classes, <u>System.JSON, REST.Client,</u> uthenticator.Basic, Rest.Types, <u>IPPeerClient;</u> TComponentを継承
5 TTra nit Tran nterface System REST.Au TLangua	anslatorコンポーネントの宣言部 nslator; e .SysUtils, System.Classes, <u>System.JSON, REST.Client,</u> uthenticator.Basic, Rest.Types, IPPeerClient; age = (Japanese, English); lator = class(IComponent) Lator = class(IComponent)
5 TTra nit Tran nterface System REST.Au PPe TLangua Privat (Pr	anslatorコンポーネントの宣言部 nslator; e REST関連のユニット、JSONユニット .SysUtils, System.Classes, <u>System.JSON, REST.Client.</u> uthenticator.Basic, Rest.Types, TPPeerClient; age = (Japanese, English); lator = class(IComponent) ^e ivate 宣言 /
5 TTra nit Tran nterface System REST.Au TLangua TTrans privato (Pr FRes FRes	anslatorコンポーネントの宣言部 hslator; e REST関連のユニット、JSONユニット .SysUtils, System.Classes, <u>System.JSON, REST.Client.</u> uthenticator.Basic, Rest.Types, IPPeerClient: age = (Japanese, English); lator = class(IComponent) <i>ivate 宣言 /</i> tClienct: IRESICLient; tRequest: IRESICLient; tRequest; IRESIRequest;
5 TTra nit Tran nterface System REST.A TLangua TLangua TIrans privat FRes FRes FRes FRes	anslatorコンポーネントの宣言部 nslator; e REST関連のユニット、JSONユニット .SysUtils, System.Classes, <u>System.JSON, REST.Client,</u> uthenticator.Basic, Rest.Types, IPPeerClient; age = (Japanese, English); lator = class(IComponent) ivate 宣言 / tClienct: IRESTClient; tResponse: IRESTResponse; icAuthenticator; INTIPE
5 TTra nit Tran nterface ses System REST.Au TLangu TLangu TLangu TIrans Privato FResi FResi FResi FResi FResi FSou	anslatorコンポーネントの宣言部 nslator; e REST関連のユニット、JSONユニット .SysUtils, System.Classes, System.JSON, RESI.Client, uthenticator.Basic, Rest.Types, TPPeerClient; age = (Japanese, English); Lator = class(IComponent) ivate 宣言 / tClienct : IRESIResponse; icauthenticator: IHITPBasicAuthenticator; rceLanguage: ILanguage; setLanguage: TLanguage; setLanguage: SetLanguage; setLanguage: TLanguage; setLanguage: TLanguage; setLanguage: TLanguage; setLanguage: SetLanguage; setLanguage: TLanguage; setLanguage: TLanguage; setLanguage: TLanguage; setLanguage: TLanguage; setLanguage: TLanguage; setLanguage: SetLanguage; setLanguage: TLanguage; setLanguage: SetLanguage; setLanguage: SetLanguage; setLanguage: TLanguage; setLanguage: TLanguage; setLanguage: TLanguage; setLanguage: TLanguage; setLanguage: SetLanguage; setLanguage: TLanguage; setLanguage: SetLanguage; setLanguage: TLanguage; setLanguage: TLanguage; setLanguage: SetLanguage; setLanguage: SetLanguage; setLanguage: SetLanguage; setLanguage: TLanguage; setLanguage: SetLanguage; setLanguage: SetLan
5 TTra nit Tran nterface System REST.Au Private FRes: FRes: FRes: FRes: FRes: FRes: FRes: FRes: FRes: FRes: FRes: FRes: FBas FSoul FDes: function	anslatorコンポーネントの宣言部 nslator: e REST関連のユニット、JSONユニット . SysUtils, System.Classes, System.JSON, REST.Client, uthenticator.Basic, Rest.Types, IPPeerClient; age = (Japanese, English): Lator = class(IComponent) ivate 宣言 / tClienct: INESICLient; tResponse; icAuthenticator: IHITPBasicAuthenticator: reeLanguage: ILanguage; ree: String; tination: String; tion GetLanguages: TLanguage): String; プロパティ
5 TTra nit Tran nterface ses System REST.A TLangua TLangua TIrans privat FRes: FRes: FRes: FRes: FRes: FSou FSou FSou FDes: func protec	anslatorコンポーネントの宣言部 nslator; e REST関連のユニット、JSONユニット .SysUtils, System.Classes, System,JSON, REST.Client, uthenticator.Basic, Rest.Types, TPPeerClient; age = (Japanese, English); lator = class(IComponent) ivate 宣言 / tClienct: TRESTClient; tResponse: TRESTClient; tRequest: TRESTClient; telanguage: TLanguage; getLanguage: TLanguage; tration: String; tion GetLanguageName(ALanguage: TLanguage); String: too GetLanguageName(ALanguage: TLanguage); String: too GetLanguage zo換元言語
5 TTra nit Tran nterface ses System REST.A private FRes FRes FRes FRes FRes FBas FSoun FTar; FSoun FDes funct Pr Public (Pr	anslatorコンポーネントの宣言部 nslator; e REST関連のユニット、JSONユニット .SysUtils, System.Classes, System.JSON, REST.Client, uthenticator.Basic, Rest.Types, TPPeerClient: age = (Japanese, English); lator = class(IComponent) ivato 宣言 / tClienct: IRESTRequest; tResponse; TRESTResponse; tResponse; TRESTResponse; treatanguage: TLAnguage; rec: String; tination: S
5 TTra nterface ses System REST.Au ype TLangua TTrans privato FRess FRess FRess FSou FDess func protec (Pr public (Pr protec	anslatorコンポーネントの宣言部 nslator; e REST関連のユニット、JSONユニット .SysUtils, System.Classes, System.JSON, REST.Client, uthenticator.Basic, Rest.Lypes, IPPeerClient; age = (Japanese, English); lator = class(IComponent); ivate 宣言 / tclienct: IRESTReguest; tRequest: TRESTReguest; tRequest: TRESTReguest; tRequest: TRESTReguest; translate メソッド: 変換実行処理 Destination プロパティ: 変換後テキスト プロパティ SourceLanguage: 変換元言語 TargetLanguage: 変換元言語 Source: 変換元テキスト
5 TTran nit Tran nterface ses System REST.A Ype TLangua ITrans privat FRes: FRes: FRes: FRes: FSour FTar; FSour For public Pro proce proce Pro	anslatorコンポーネントの宣言部 nslator; e REST関連のユニット、JSONユニット .SysUtils, System.Classes, System.JSON, RESI.Client, uthenticator.Basic, Rest.Types, TPPeerClient; age = (Japanese, English); Lator = class(IComponent); <i>Vata 宣言 /</i> tolienct : TRESTClient; tResponse; TRESTRequest; treatanguage: TLASRuage); reclanguage: TLASRuage); reclanguage: TLASRuage); setLanguage; TLASRUAGE); blished graf / blished graf /
5 TTra nit Tran nterface ses System <u>REST.A</u> ype TLangua TTrans privato <i>FResi</i> FResi FResi FResi FSour FTari FSour FDesi funci protecc <i>Pri</i> public <i>Prop</i> publis <i>Prop</i>	anslatorコンポーネントの宣言部 nslator; e REST関連のユニット、JSONユニット .SysUtils, System.Classes, System.JSON, REST.Client, uthenticator.Basic, Rest.Types, IPPeerClient; age = (Japanese, English); lator = class(IComponent) ivato 宣言 / tClienct: IRESIRequest; Response: IRESIRequest; Response: ILanguage; reclanguage: ILanguage; reclanguage: ILanguage; setLanguage: ILanguage; totation: String; tination: String; read FDestination; bdished 宣言 / erty SourceLanguage: ILanguage read FSourceLanguage write FSourceLanguage; ting for Concelanguage: TLanguage read FlarzetLanguage write FavetLanguage; tring FarzetLanguage write FarzetLanguage; tring FarzetLanguage write FarzetLanguage; tring FarzetLanguage write FarzetLanguage; tring FarzetLanguage write FarzetLanguage; FarzetLanguage write FarzetLanguage; FarzetLanguage write FarzetLanguage; FarzetLanguage write FarzetLanguage; FarzetLanguage write FarzetLanguage; FarzetLanguage; FarzetLanguage write FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage; FarzetLanguage;





福井 和彦 / 小杉 智昭

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

- 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
 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

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 エ 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

最後に

4.

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&flowNa me=GlifWebSignIn&flowEntry=Sign Up

(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 カウントヘログインした後、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/mapsplatform/

詳しい登録手順は、さまざまなサイト



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、InternetExplorer7.0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 InternetExplorer 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 メソッドを使ってブラン ク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 くつかあるが、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 プロパティヘセットし、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、SaveTo Stream 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。

0	oogle cloud		51 +				
← Al	リキー	₿ キーを再生成	■ 削除				
名前 API キー	1						
キーの制制限する。	限 ト 不正使田や割に	当ての次田を防止できます	Learn more				
▲ アプリ	リケーションの制限	:なし APIの制限: M	laps JavaScript API				
アプリケ・	ーションの制限	API の制限					
アプリケーオ	-ションの制限では	、このキーを使用できるウ	フェブサイト、IP アドレス	ス、またはアプリを指定	Eしま		
アプリケ・	ーションの制限						
	リファラー(ウェ	ブサイト)			社内ネットワー	-ケで使用する場合は	t,
● IP アト	ジレス (ウェブサー ロアプリ	バー、cron ジョブなど)	\supset —		外部入口のIP	アドレスを設定する	
0 ios 7	プリ				7		
リクエスト 例: 192.16	を受け入れるサーバ 3.0.1、172.16.0.0/12	ニーIP アドレス 、2001:db8::1、2001:db8::/6	4				
IP / FL	~~						
Google	Maps Platfo	rmの利用手続きに	ついて④				
Google = +	Maps Platfo Google Cl API キー	rmの利用手続きに・ oud Platform C キーを	ついて④ ひいて④ ひいて④ ● Test ▼ 再生成 盲 削	除	۹		
Google 三 (名前	Maps Platfo Google Cl API キー	rmの利用手続きに oud Platform C キーを	ついて④ ● Test ▼ 再生成 盲 削	除	٩		
Google 三 く API キ	Maps Platfo Google Cl API ≠— −1	rmの利用手続きに oud Platform C キーを	ついて④ ● Test ▼ 再生成 1 削	除	٩		
Google 三 く APIキ	Maps Platfo Google Cl API ≠—	rmの利用手続きに oud Platform C キーを	ついて④ ♣ Test ▼ 再生成 1 削	除	۹		
Google 三 く 名前 API キ キーの	Maps Platfo Google Cl API キー - 1	rmの利用手続きに oud Platform C キーを	ついて④ ● Test ▼ 再生成 1 削	除	٩		
Google そ 名前 APIキ キーの 制限する	Maps Platfo Google Cl API キー 	rmの利用手続きに oud Platform C キーを や割り当ての盗用を防 D制限: なし API	ついて④ ・ Test ▼ 再生成	除 more Script API	٩		
Google 三 く 名前 API キ キーの 制限する み フコー	Maps Platfo Google Cl API キー ー1 物限 ると、不正使用 ⁴	rmの利用手続きに oud Platform C キーを や割り当ての盗用を防 D制限: なし API	ついて④ ◆ Test ▼ 再生成 會 削 此できます。Learn の制限: Maps JavaS	除 more Script API	٩		
Google そ 名前 APIキ キーの 制限する アプリア	Maps Platfo Google Cl API キー ー1 *制限 ると、不正使用 ⁴ ^プ リケーションの制	rmの利用手続きに oud Platform C キーを や割り当ての盗用を防 D制限: なし API 限 API の制限	ついて④ ● Test ▼ 再生成 ● 削 止できます。Learn の制限: Maps JavaS	除 more Script API	٩		
Google 三 く 名前 APIキ キーの 制限する アプリイ APIの制 APIの制	Maps Platfo Google Cl API キー ー1 *制限 3と、不正使用 プリケーションの制 アーションの制 I限では、この=	rmの利用手続きに oud Platform C キーを や割り当ての盗用を防 D制限: なし API 限 APIの制限 Fーで呼び出すことが	ついて④ ・ Test ▼ 再生成	除 more Script API ます。	٩		
Google そ 名前 APIキ キーの 制限する アプリク APIの精 APIの精 APIの精 APIの精	Maps Platfo Google Cl API キー ニー1 ・制限 ると、不正使用 プリケーションの制 リアーションの制 眼では、このコ 眼では、このコ 眼では、このコ	rmの利用手続きに oud Platform C キーを や割り当ての盗用を防 D制限: なし API 限 APIの制限 キーで呼び出すことが	ついて④ ・ Test ・ 再生成	除 more Script API ます。	٩		
Google そ 名前 API キ キーの 制限す アプリ・ API の精 API の精 Maps Select	Maps Platfo Google Cl API キー ー1 *制限 ると、不正使用* [†] リケーションの制作 リケーションの制作 I限では、この= I限 JavaScript API	rmの利用手続きに oud Platform C キーを や割り当ての盗用を防 D制限:なし API 限 APIの制限 Fーで呼び出すことが	ついて④ ・ Test ・ 再生成 ・ 前 山できます。Learn の制限: Maps JavaS できる API を指定しま	除 more Script API ます。 章	Q ()		

それでは、次の手順で画面を作成して (3) フォームの OnShow イベントでは、 業所の詳細情報を表示する方法について いく。 Google マップ表示用 HTML を使用し 説明する。ただし、マーカーの情報を取 て TWebBrowser コンポーネントに 得するためには TWebBrowser コン (1) 【図12】 のように、フォームに Google マップを表示している。 ポーネントを経由して JavaScript を実 TDBGrid コンポーネントと OnShow イベントに【ソース5】【ソー 行して戻り値を取得する必要がある。し TWebBrowser コンポーネントを貼り ス6】のように記述する。それでは かし、TWebBrowser コンポーネント 付ける。そして、サンプルファイルを参 OnShow イベントで行っている内容に には IavaScript を実行するメソッドは 照するために FireDAC 関連のコンポー ついて詳しく見ていこう。 実装されていない。独自で実装すること ネントと TDataSource コンポーネント も可能ではあるが簡単ではない。 【ソース5のポイント①】 を貼り付ける。FireDAC 関連の各コン そこで本稿では、JavaScriptを実行 ポーネントの設定は次のとおり。 IBM i に接続し、TDBGrid コンポー するメソッドなど TWebBrowser コン ネントに営業所名の一覧を表示する。 ポーネントでは実装されていないメソッ ●TFDConnection コンポーネントの設 ドやプロパティが実装されている 【ソース5のポイント②】 「TEmbeddedWB| コンポーネントと 定 接続エディタにて設定する。 ここでは Google マップを表示する いうフリーのコンポーネントを利用す HTML を編集している。HTML の編集 ・接続定義名:CO400DEF る。TEmbeddedWB コンポーネントは Database : Delphi/400Configuration イメージは【図13】を参照していただ bsalsa productions にて開発・公開され の AS/400Name きたい。 ていたフリーのコンポーネント群の1つ ・User_Name: IBM i 接続ユーザー まず、サンプルファイルを全件参照し だが、現在は little earth solutions に ・Password : IBM i 接続ユーザーのパ ながら、各レコードの緯度 (SPLATI) よって GitHub 上でメンテナンス・公開 スワード と経度(SPLNGI)を取得する。そして されている。 【ソース3】で組み込んだ、マーカーを • ODBCAdvanced : LibraryOption 立てる JavaScript 「setMarker」の引 = 「サンプルファイルが有る Library TEmbeddedWB コンポーネントのダウ 数に取得した緯度と経度をセットする。 ンロードサイト 名] \mathcal{L} \mathcal{O} setMarker \mathcal{E} $[\mathcal{V} - \mathcal{X} 3] \mathcal{O}$ https://github.com/littleearth/Delphi-HTML の続きとして、サンプルファイ ● TFDQuery コンポーネントの設定 EmbeddedWB SQL プロパティにサンプルファイル ルに登録されているデータ全件分を結合 を参照する以下の SQL 文を埋め込む。 する。こうすることで、1回のHTML GitHub 上では Delphi 5 ~ Delphi 10 のリクエストで全件分の営業所のマー Seattle までが対象となっているが、 (SQL 文) SELECT * FROM [サンプルファイ カーが付いた状態の Google マップを表 Delphi 10.1 Berlin や Delphi 10.2 ル名] 示することができる。 Tokyo 向けの dpk ファイルも存在する ORDER BY SPEGCD (* dpk ファイルの読み込み前に res 【ソース6のポイント①】 ファイルを削除しなければ正しく取り込 【ソース6】の Google マップを表示す めない)。ダウンロードサイトより TDBGrid コンポーネントのカラムに は営業所名(SPEGNM)を設定する。 る処理については3章の内容と同じにな TEmbeddedWB コンポーネントをダウ る。ただし、Google マップを表示する ンロードして Delphi/400 の開発環境に HTML をセットする個所は、【ソース5 (2) Google マップ表示用 HTML は【ソー インストールをする。 ス3】 【ソース4】を参照。この章では 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。 それでは、次の手順に従って、4章で HTMLを2つに分割している。分割し 作成した画面に機能を追加していく。 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 完成した画面を実行すると完成イ カーを立てるために Google Maps (1) 【図 14】に従ってコンポーネントの メージ【図9】のように、各営業所の位 Platform の API を使用した JavaScript 置にマーカーが立っている Google マッ 追加/変更を行う。変更内容は、まず 「setMarker」を組み込んでいる。 プが表示される。 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。 【ソース4】では、Google マップ表示 3-2.クリックしたマーカーの情報を 後に地図の中心にしたい緯度と経度を組 次に、TTimer コンポーネントを追 取得する み込んでいる。本稿では「難波サンケイ 加し、Enabled プロパティを False、 ビル 本社」を中心にするため、その緯 この章では、3-1で作成した画面に機 Interval プロパティを 200 に設定する。 度と経度を組み込んでいる。そして3章 能を追加し、クリックした Google マッ TTimer コンポーネントの用途について

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

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。

78

と同様に、2章で取得した API キーを

HTML へ組み込む。

図5 APIキーの取得 のの * Test * 2 認証情報 2	
Ym Yest Q 認証情報 Buth 周星周 F×インの強認 医近硝酸乙酸酯 F×インの強認 一般時 Bith 一般的な API にアクセスするための証明書を作成します。詳しくは、API Fキュメントをご覧ください。 API キー 2010/00/14 1 P アドレス 1 個の API 1 2010/00/14 P アドレス 1 個の API To	
認証情報 ビビボ酸 Outh 同意面面 ドメインの確認 「夏夏傍吸を推惑 ● 耐油 一 対応な API にアクセズスするための証明書を作成します。詳しくは、API ドキュメントをご覧くたさい。 API キー ● 名称 作成日 ~ 解釋項 * ● API キー ● API キー ● 2010/00/14 ● アドレス 1 個の API OG Googleマップを表示する() レジストリのキーに値を追加する	
acade 16 年秋 WEIE (前秋: Outh 周東国 ドメインの強怒 「夏夏(前後を作成)」 新浩 市坊な API にアクセスするための証明書を作成します。詳しくは、API ドキュメントをご覧ください。 API キー ○ 名称 作成日 × 新慶野項 ● API キー ○ Orgole マップを表示する① レジストリのキーに値を追加する	
超磁情報 Outh 同意面面 F×インの強怒 一 一 一 一 小 小 一 クロ 市 一 クロ 市 一 クロ 市 ○ API #-1 2018/08/14 IP 7F FUZ.1 留の API □ API #-1 2018/08/14 IP 7F FUZ.1 留の API	
図値備報を作成・ 削除 特効な API にアクセスするための証明書を作成します。詳しくは、API ドキュメントをご覧ください。 API キー 金布 作成日 ~ API キー API キー 2018/00/14 IP アドレス 1 個の API	
APIキー 金柿 作成日 × 税厚専項 キー APIキー1 2018/08/14 IP アドレス.1 個の API 面 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	
 ○ 名称 作成日 × 税募項 まー ○ APIキー1 2018/08/14 IP アドレス.1 個の API ○ B 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 	
図6 Googleマップを表示する①	
図6 Googleマップを表示する①	
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>	
大臣/山 9 31世/ DWORD (32 ビット) 値の編集 X	
値の名前(1): TWebBrowserを使用するExe名	
(100 ⁻ -ク(0): 表記) (10 ⁻)(10	
OK キャンセル	
(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/ Googleマッフを表示する(2)	
C Form1	
TWebBrowser	

(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 ボーネントの 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ごとに、 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ごとに 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、無料で使用できる枠には限度 があり、無料で使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 を利用する際には注意



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		
platform/		
 (上記 URL のメニューにある"料金" のリンク生)		
 M		

_

_

_

_

_

_

_





営業所 CD	営業所名	営業所略称	郵便 番号		住所1	住所	2
100001 200001	難波サンケイビル 本社 霞が関 営業所	難波サンケイ 霞が関	556-0017 100-0013	大阪府大阪東京都千日	版市浪速区湊町~ 町区霞が関~	難波サンケイト 霞が関東急ビ	in~ 7
			06-	E話番号 6631-8601 5510-5701	FAX番号 06-6631-8603 03-5510-5702	緯度 34.6651420 35.6714410	経度 135.4955550 139.7447100
12 Goog	leマップにマーカーを立	73④					
	営業所名 TDBGrid					TWebBro	owser
1	NysCO4000riverLink1 1	TFDPhysCO4 TFDGUIxWait	100DriverLin tCursor	ık			
	1 conAS400 gryList 1 1	TFDConnecti TFDQuery	on				
1 1 サ:	dallet ! ノブルファイルへはFirel	TDataSource DACで接続する	\$				



ソース5 フォームのOnShowイベント procedure TfrmSample.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) usesにWinapi.ActiveXを追加する var sl: TStringList; ms: TMemoryStream; sMap: string; begin_____ !// IBMiへ接続 ポイントの conAS400.Connected := True; // 営業所リスト表示 gryList.Open; レーバー 「「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 ポイントの ----------sMap := GMapHTML_1; gryList.DisableControls; try gryList.Last; while not gryList.Bof do begin sMap := sMap + ' setMarker(' + qryList.FieldByName('SPLATI').AsString + ',' + qryList.FieldByName('SPLNGI').AsString + ');'; gryList.Prior; end; finally gryList.EnableControls; l_end:_____ ↓ [ソース6]へ続く 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; ポイント① 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;













株式会社ミガロ.

RAD事業部 技術支援課

[Delphi/400] RAD Serverを使った新しい 多層アプリケーション構築

はじめに
 RAD Server の特徴
 RAD Server の実装手順
 サーバーアプリケーション構築手順
 クライアントアプリケーション構築手順
 RAD Server の管理分析機能
 補足:他言語からの活用
 おわりに



Delphi/400 上級

1978年3月26日生まれ 2001年3月龍谷大学法学部卒業 2005年7月株式会社ミガロ、入社 2005年7月システム事業部配属 2007年4月RAD事業部配属

現在の仕事内容 Delphi/400を中心に製品試験およ

Delphi/400 を中心に裂品試験およ び月 100 件に及ぶ問い合せサポー トやセミナー講師などを担当してい る。

1.はじめに

ここ数年で Delphi/400 によるモバイ ルアプリケーションの業務開発も多く なってきた。そうしたモバイルアプリ ケーション開発において、IBM i を利用 するために重要となるのが中間サーバー を使った多層構成の仕組みである。【図 1】

Delphi/400 でこうした多層アプリ ケーションを開発する場合、サーバーも クライアントも Delphi/400 だけですべ て開発することができる強みがある。こ の中間サーバーのアプリケーション開発 では、DataSnap Server という技術が 使われているが、Delphi/400 の最新版 である 10.2 Tokyo では、新しく RAD Server というサーバー構築技術が利用 できるようになっている。

本稿ではこの新しい RAD Server に ついての特徴や実装方法、そして RAD Server を活用した他言語アプリケー ションとの連携テクニックなどの応用に ついて検証した内容を説明する。 なお本稿では RAD Server が利用で きる Delphi/400 10.2Tokyo の環境を 前提としている。

2.RAD Serverの特徴

新しく利用できるようになった RAD Server について、主な特徴を確認する。

まず RAD Server は、DataSnap Server と同様に、多層アプリケーショ ンの中間サーバー部分を実装する技術で ある。多層アプリケーションとは、複数 層で構成されたアプリケーションのこと で、たとえば Web アプリケーションや モバイルアプリケーションで IBM iに アクセスする場合には、中間層のサー バーを経由する 3 層構成となる。この中 間サーバーのアプリケーションを実装で き る の が DataSnap Server や RAD Server である。

ではこの2つの技術にどのような特 徴の違いがあるかをまとめてみる。

まず、従来のバージョンでも使用可能

な DataSnap Server は、次のような特 徴を持つ。

DataSnap Server

- ・多層アプリケーションの開発を可能に する SDK
- ・サーバー機能はプログラムで開発する 必要がある
- ・開発での実装となるため、プログラム の自由度が高い
- ・TCP/IP、HTTP(S)、REST、 JSON、COM などの 標準技術をサポート

次に RAD Server は次のような特徴 を持つ。

• RAD Server

- ・多層アプリケーションの REST API
 を公開するサーバー
- ・サーバーで必要となる高度な機能がい くつも提供されている
- ・ユーザー管理機能、認証機能、分析機 能などの標準機能を豊富に搭載

		for Business
クライアント	中間サー/	Ϋ́ IBM i
ີ 2 DataSnap ServerとRA	D Serverの違い	
	DataSnap Server	RAD Server
機能開発	全て開発で実装が必要	必要な部分のみ開発
標準通信	TCPIP/HTTP(S)	HTTP(S) (REST形式固定)
DBエンジン	FireDAC、 dbExpress	FireDAC
モバイル対応機能	開発が必要	Push通知、デバイス認証等が 標準機能
管理ツール	開発が必要	標準で付属(分析も可能)
ライセンス	開発ライセンスに含まれる (Enterprise以上)	開発ライセンスに 1サイトライセンス付属 (10. 2 Tokyo Enterprise以降)
S RAD Serverを使った多 Form1 10 NAME 1221 33ナッツマリンショップ2 1513 ダイブハウスタートあ5 3444 ダイビングベース新井巻 1231 アクアダイビングセンター 1351 亀山ダイブセンター 1360 ダイブショップブルーリーフ 1384 MHMダイバーズクラブ 1984 ADVENTURE UNDERSEA 2118 グリーンスポーツワラブ 2135 バイナップルダイバーズ	■ アプリケーションの実装例 ■ ロ × ● NAME 1221 ユヨナッツマリンショップ2 1513 ダイジハクスタートル5 3444 ダイビングペース新井8 1231 アクアダイビングセンター 1360 ダイブショップブルーリーフ 1360 ダイブショップブルーソーフ 1360 ダイブショップブルーソーフ 1360 ダイブショップブルーソーフ 1360 ダイブショップブルーソーフ 1360 ダイブショップブルーソーフ 1360 ダイブショップブルーソーフ 1360 ダイブショップブルーソーフ 1360 ダイブショップブル・ソーフ 1360 ダイブショップブルーソーフ 1360 ダイブショップブルー 1360 ダイブショップブルーソーフ 1360 ダイブショップブルーソーフ 1360 ダイブショップブルーソーフ 1360 ダイブショップブルーソーフ 1360 ダイブショップブルーソーフ 1360 ダイブショップブルーソーフ 1360 ダイジンター 1360 ダイジンター 1370 ダイレンター 1370 ダーム 1370 ダーム 1370 ダーム 1380 ダイブショップブルー 1380 ダイジンター 1380 ダイジンター 1380 ダイジンター 1380 ダイジンター 1380 ダイジンター 1380 ダイジェックラブ 2350 パーム 2155 ダーム 2155 ダーム 2155 ダーム 2155 ダーム 2155 ダーム 2155 ダーム 2155 ダーム 2155 ダーム 2155 ジーム 2155 ジ	DX RAD Server

細かい比較については【図 2】にまと めている。

比較するとわかるが、最も大きな違い は DataSnap Server はサーバーアプリ ケーションを細かく自由に開発すること を目的とした技術で、RAD Server は 大枠が完成されたサーバーアプリケー ションを、カスタマイズをして使う技術 ということである。

RAD Server は標準の機能がいろい ろ揃っている分、DataSnap Server と 違ってアプリケーションの仕組みは REST 形式で固定となっている。REST とは Representational State Transfer の略で、そのサービスの URL が持つメ ソッドにアクセスすることでデータの送 受信をステートレスで行う技術である。 汎用性が高く、Web やモバイルのアプ リケーションで広く使用されている。そ のため、Delphi/400 のアプリケーショ ンだけに限定されずさまざまなアプリ ケーションから活用することもできる (これについては後述する)。

どちらが優れているかは開発するア プリケーションの要件によっても異なる が、シンプルな機能の中間サーバーであ れば RAD Server の標準機能が強みを 発揮できる。

この2つの技術は、中間サーバーで 担っている役割は似ているが、実際の実 装手順は異なる部分も多い。そのため、 次章では新しい RAD Server の基本的 な実装手順について確認していく。

3.RAD Serverの 実装手順

RAD Server で構築するアプリケー ションは、DataSnap Server と同様に、 サーバーアプリケーションとそれを利用 するクライアントアプリケーションの2 種類で構成される。それぞれの実装手順 を2つのステップに分けて説明してい く。

・3-1.サーバーアプリケーションの構築
・3-2.クライアントアプリケーションの 構築 なお本稿では、【図3】に示すように PC やモバイルのクライアントアプリ ケーションから IBM i のデータを取得 する基本的な多層アプリケーションを題 材とする。

3-1.サーバーアプリケーション構築 手順

初めに中間サーバーに実装するアプ リケーションを構築する。構築は RAD Server にウィザードが用意されている ため、それほど難しいものではない。

【手順①プロジェクトの作成】

新規作成より、RAD Server (EMS) パッケージを選択してウィザードを起動 する。【図 4】

EMS という名称は、RAD Server 関 連の機能を意味する。

RAD Server が当初 EMS Server と いう製品名であったため、機能やコン ポーネント名で使われていることが多 い。

【手順②リソースの指定】

次にウィザードに従って設定を進め ることになるが、【図5】のように「リソー スを含むパッケージを作成する」を選択 して、そのリソース名を任意で命名する。 RAD Server は REST 形式になるた め、URL でアクセスする際に利用する 機能をリソースとして実装する。ここで 設定しておくと、指定したリソース名の ソースが作成される。たとえば【図5】 のように「CUST」というリソース名を 指定すると、実行時に下記のような REST サービスとして呼び出すことがで きる。

http (s) :// サーバー /CUST

【手順③機能の指定】

②で指定したリソースに対して、どう いった機能のテンプレートを作成するか を【図6】で設定する。デフォルトは基 本機能としてデータを取得する Get、 GetItem が選択されているが、Put、 PutItem、DeleteItem といった更新系 の機能も用意されている。なお名称に Item と付く機能は、パラメータを渡し て処理できることを意味している。 【手順④自動生成されるソース】

ここまでのウィザード操作が完了す ると、【図7】のように中間サーバーア プリケーションのソース一式が自動生成 される。

次の手順からはこのソースに必要な プログラムを実装していく。

【手順⑤コンポーネントの配置と設定】

自動生成されたリソースに機能を実 装するにあたって、まずは必要なコン ポーネントを配置する。RAD Server で使えるデータベースエンジンは最新の FireDAC のみに限定されている。その ため、ここではデータ取得(Get)の仕 組みを作ることを前提に FireDAC の主 要なコンポーネントを配置する。【図 8】

TFDConnection

- TFDPhysCO400DriverLink
- TFDTable
- TFDSchemaAdapter
- $\cdot \ {\rm TFDStanStorageJSONLink}$
- $\cdot \ {\tt TFDGUIxWaitCursor}$

TFDConnection ~ TFDTable およ びTFDGUIxWaitCursor については、 一般的なFireDACのアプリケーション で実装する内容である。本稿では FireDACの詳しい使い方は割愛するが、 【図9】のようにTFDConnection や TFDTable を設定する。

RAD Server の構築でポイントにな るのはTFDSchemaAdapterと TFDStanStorageJSONLinkコンポーネ ントである。これはRAD Serverが REST形式として動作することが前提と なるため、TFDTableで取得したデー タをJSON形式に変換する機能の実装 に使用する。【図9】でTFDTableの SchemaAdapterプロパティに TFDSchemaAdapterを設定している のは、そのためである。

【手順⑥データ取得の機能を実装】

次に RAD Server がリクエストに応 じてデータを返す機能をプログラムで実 装する。ウィザードでリソースの Get 機能を指定しておけば、自動的に Get メソッドが作成されているので、ここで 必要な処理だけを数行コーディングす る。【ソース1】



処理内容としては TFDTable でデー タを開き、そのデータを TFDSchema Adapter で JSON 形式のストリームに 変換するだけである。この時、JSON 形 式 を 指 定 す る た め に TFDStanStorageJSONLink が必要と なっている。また図 6 で Item を使った 機能を実装する場合は、パラメータであ る ARequest を 使 っ て ARequest. Params.Values ['XXXX'] という形で 簡単に利用することができる。

プログラムの実装はこれで一通り完 了である。

【手順⑦コンパイルと実行】

最後に作成した RAD Server の中間 サーバーアプリケーションをコンパイル して実行する。ただし初回のコンパイル 時にはコンポーネントのパッケージ開発 時と同様 (RAD Server もパッケージ 方式) にパッケージの参照確認や InterBase の設定確認のダイアログが表 示されるので応答が必要となる。【図 10】

InterBase は意識して使用するわけで はないが、RAD Server が標準で備え る管理分析機能等で内部的に使用してい る。なお InterBase のライセンスは RAD Server に含まれているので別途 購入の必要はない。

コンパイルが完了してアプリケー ションを実行すると、【図 11】のような 画面が起動して完成である。この画面で は、自動的にアクセスログが出力される ようになっており、クライアントからリ クエストがあるとこの画面にログが表示 される。

この起動画面はあくまでRAD Server に用意された標準の exe アプリケーショ ンである。DataSnap Server と違い、 作成したプログラム自体がサーバーとし て起動しているわけではない。

コンパイルしたリソースは bpl として 作成されて RAD Server に読み込まれ ている。運用環境の構築やリソースの配 布をする際には、エンバカデロ社の下記 オンラインヘルプに詳細情報が公開され ているため、参考いただきたい。

http://docwiki.embarcadero.com/ RADStudio/Tokyo/ja/

3-2.クライアントアプリケーション 構築手順

次に、前節で作成した RAD Server アプリケーションに接続してデータを取 得するクライアントアプリケーションを 構築する。

【手順①プロジェクトの作成】

新規作成よりマルチデバイスアプリ ケーションを選択して、FireMonkey の新規プロジェクトを作成する。【図 12】

今回はモバイルでの動作を見るため に FireMonkey で作成しているが、デ スクトップアプリケーションとして VCL で作成する場合も実装内容は同じ である。

【手順②コンポーネントの配置・設定】 新規作成したアプリケーションに必 要なコンポーネントを配置する。【図13】

TEMSProvider

- $\cdot \ {\rm TEMSFireDACClient}$
- $\cdot \ {\rm TFDSchemaAdapter}$
- TFDTableAdapter
- TFDMemTable
- TFDGUIxWaitCursor

このアプリケーションでも FireDAC コンポーネントを使用するが、直接 IBM i に接続するわけではなく、前節で 作成した RAD Server アプリケーショ ンに接続してデータを取得する仕組みと なる。そのため、TFDConnection では なく、TEMSProvider および TEMSFireDACClient の専用コンポー ネントを使用する (EMS は RAD Server 関連を意味する)。

また RAD Server より取得するデー タは JSON 形式で送られてくるため、 それを Delphi/400 のデータセット形式 に変換して戻すために TFDSchemaAdapter を使用する。そ れぞれのプロパティ設定は【図14】に 示す。

ポイントとしては TEMSProvider で 接続する RAD Server の IP やポートを 指定し、TEMSFireDACClient で呼び 出すリソース情報を設定する点である。 リソース名は RAD Server のウィザー

ドで指定したものを設定する。

なおデータの画面表示はFireMonkey なので、【図 15】のようにライブバイン ディングを使って実装できる。VCLの 場合は TDBGrid を使うこともできる。

【手順③ RAD Server の呼び出し機能の 実装】

最後に RAD Server からデータ取得 の Get 機能を REST で呼び出す処理を プログラムで実装する。RAD Server はほとんどのアクセス制御を前手順のコ ンポーネントで処理してくれるため、 コードは1行で済む。【ソース2】

これでクライアントアプリケーショ ンのデータアクセスのプログラムが完成 したことになる。RAD Server と DataSnap Serverの開発は似ているが、 RAD Serverでは標準機能がコンポー ネントなどで備わっている分、簡単に実 装ができる部分も多い。

【手順④コンパイル・実行】

完成したプログラムを目的のプラッ トフォームを指定してコンパイルして実 行すると、WindowsやiOS、あるいは Android 向けにアプリケーションを利 用することができる。【図 16】

全体的な仕組みは冒頭で説明した【図 3】のとおりである。

4.RAD Serverの 管理分析機能

前章で RAD Server の基本的な実装 手順を説明したが、RAD Server が持 つ機能について少しだけ補足を加える。

3-1. で実装した RAD Server のアプ リケーションは、起動画面でアクセスロ グが確認できることを説明したが、利用 できる機能はそれだけではない。

RAD Server は管理機能を標準で備 えているため、たとえばサーバーに対す るアクセス分析などの機能は開発しなく とも自動で実装されている。使い方とし ては起動画面メニューの「コンソールを 開く」ボタンをクリックする。【図17】

そうすると【図 18】のようなコンソー ルのログイン画面がブラウザで起動す る。このログインはユーザー名が consoleuser、パスワードが consolepass がデフォルトになっている。



このアカウント情報はRAD Server 上の下記iniファイルに設定されており、 テキスト編集で変更が可能である。

C:¥Users¥Public¥Documents¥Emba rcadero¥EMS¥emsserver.ini

ログインすると、【図 19】のように RAD Server にアクセスするクライア ントの状況や作成した REST リソース の使用頻度について、グラフを使ったビ ジュアル分析が行える。これによってど の機能がよく使われているか、どんな時 間帯に処理が集中しているか、などの把 握も容易である。

5.補足:他言語からの 利用

ここまで RAD Server を使った基本 的なアプリケーションの実装内容や機能 についてまとめてきたが、最後に RAD Server をさらに活用するための使い方 を考察していく。

RAD Server が REST 形式になるこ とは冒頭で説明したが、この REST 形 式はここまでの実装でも説明したとお り、JSON を利用できる。この REST/ JSON の組み合わせは、非常によく使用 される 通信方式 である ため、実は Delphi/400 以外の 他言語 アプリケー ションでも利用可能である。

本稿では他言語の例として、Delphi 言語と同じ開発元であるエンバカデロ・ テクノロジーズ社が提供する Sencha Architect を題材として RAD Server の利用を説明する。Sencha Architect とは、表、グラフ、ツリーなどの多彩な コンポーネントをドラッグ&ドロップし て簡単に HTML5 対応の Web アプリ ケーションを構築できる開発ツールであ る。【図 20】

こ の Sencha Architect は REST/ JSON データと連携してデータを扱うこ ともできるため、RAD Server とも容 易に連携ができる。

● RAD Server の実装調整

3章で実装した GET 機能の JSON は Delphi/400 のコンポーネントに特化し た作り方となっているため、もっと一般 的な JSON 形式に調整する。【ソース 3】 ● Sencha Architect からの利用

本稿では Sencha Architect の詳し い開発手順については割愛し、ポイント になる部分を説明する。

Sencha Architect に も Delphi/400 と同じようなコンポーネント(部品)が 用意されているため、RAD Server か ら取得したデータを表示するために Grid Panelというコンポーネントをプ ログラムに配置する。【図 21】

この Grid Panel は Delphi/400の TDBGrid と扱いが似ている。

次にGrid Panelのデータ参照元を 指定する。

これは Grid Panel に Grid Builder と いうウィザード機能が用意されており、 【図 22】のように JSON WebService 形 式で RAD Server の URL が参照元とな るように設定する。これによってフィー ルド情報なども自動で取り込んで定義で きる。

このデータを Grid Panel の参照元に セットすれば、Delphi/400 と同じよう に Sencha Architect 上で RAD Server からの情報を表示することができる。【図 23】

このプログラムを Sencha Architect でコンパイルして実行するとブラウザで Web アプリケーションとして実行され、 RAD Server の情報を表示することが 可能である。【図 24】

Sencha Architect の細かい設定手順 は省略しているが、RAD Server の REST/JSON 形式は非常に汎用性が高 く、別の言語でも簡単に扱えることがわ かる。

もちろん Sencha Architect に限ら ず、他の言語でも REST/JSON が扱え れば同じように使用が可能である。これ を理解していれば RAD Server をより 広い範囲で活用することができる。

6.おわりに

本稿では、新しい多層アプリケーショ ンの構築技術として RAD Server の特 徴や実装手順について説明した。従来の DataSnap Server と 違 っ て、RAD Server は開発形式がある程度限定され ているものの、あらかじめ専用コンポー ネントやサーバーの管理機能を備えてい るため、効率よくアプリケーションを実

装するには効果的である。

また5章で述べたように、この REST サービスは非常に汎用性が高く、 Delphi/400 以外の言語でも利用するこ とができる。

今後、クラウドやソリューションを含 めて多様化していくシステム連携の中心 となるサーバーとして活用が期待でき る。

Μ

パッケージの参照	(サーバ自体の管理データ)
Project1.dproj の変更	X 展 IMS セットアップ 2 パブード ー ロ X
このパッケージをインストール演みのパッケージと共存させるには以下	データベースの新規作業 新した10% デーラベースの補助ではメージを確定します。
の文化から来てす。 OKを押すと変更が通用されパッケージが再構築されます。	
dortf の追加 dortf は以下のユニットを含みます: Data.FmtScd, Data.D6Consts, Data.SgTmSt, Data.D6,	2 - 1 - 1 / 3 / 3 / 1 20 7 - (A S / 1 / 2) / 1 20 7 - (A S / 1 / 2) / 1 20 7 - (A S / 1 / 2) / 1 20 7 - (A S / 2) / 2 20 7 - (A
Cata.Decommoniypes FireDAC の適切 FireDAC (は以下のユニットを含みます:	(0 2) -97-82 (7) -92
FireDAC.Comp.DataSet, FireDAC.Comp.UI, v	Performent Learning
OK キャンセル ヘルフ	
	>>205 [7] 44526
ス1 データ取得の機能を実装	
procedure TCUSTResour	cel.Get(const AContext: TEndpointContext:
const ARequest: TEndp	ointRequest; const AResponse: TEndpointResponse);
ostr: TMemoryStream	
begin	
oStr := TMemoryStre	am.Create;
// クエリの実行結果をス	キーマアダプタから
// クエリの実行結果をス // メモリストリーム経由で	キーマアダプタから 『返す
<pre>// クエリの実行結果をス= // メモリストリーム経由で FDTable1.0pen;</pre>	キーマアダプタから 『返す
<pre>// クエリの実行結果をス= // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS</pre>	キーマアダプタから が返す .veToStream(oStr,TFDStorageFormat.sfJSON); .tream(oStr.'application/ison', True);
<pre>// クエリの実行結果をス= // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close;</pre>	キーマアダプタから 「返す LveToStream(oStr,TFDStorageFormat.sfJSON); Ltream(oStr,'application/json', True);
<pre>// クエリの実行結果をス= // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre>	キーマアダプタから 『返す weToStream(oStr,TFDStorageFormat.sfJSON); tream(oStr,'application/json', True);
<pre>// クエリの実行結果をス- // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre>	キーマアダプタから 『返す LveToStream(oStr,TFDStorageFormat.sfJSON); tream(oStr,'application/json', True);
<pre>// クエリの実行結果をス= // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre>	キーマアダプタから 「返す LveToStream(oStr,TFDStorageFormat.sfJSON); Ltream(oStr,'application/json', True);
<pre>// クエリの実行結果をス= // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre>	キーマアダプタから 「返す oveToStream(oStr,TFDStorageFormat.sfJSON); otream(oStr,'application/json', True);
<pre>// クエリの実行結果をス= // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre> 1 RAD Serverの起動	キーマアダプタから が返す oveToStream(oStr,TFDStorageFormat.sfJSON); otream(oStr,'application/json', True);
<pre>// クエリの実行結果をス= // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre> 1 RAD Serverの起動 RAD Server起動	キーマアダプタから T返す OveToStream(oStr,TFDStorageFormat.sfJSON); Otream(oStr,'application/json', True); 画面
<pre>// クエリの実行結果をス= // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre> 1 RAD Serverの起動 RAD Serverの起動	キーマアダプタから weToStream(oStr,TFDStorageFormat.sfJSON); tream(oStr, 'application/json', True); 画面
<pre>// クエリの実行結果をス= // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre> 1 RAD Serverの起動 RAD Serverの起動	キーマアダプタから T返す EveToStream(oStr,TFDStorageFormat.sfJSON); Stream(oStr, 'application/json', True); 画面
<pre>// クエリの実行結果をス= // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre> 1 RAD Serverの起動 RAD Serverの起動	キーマアダプタから Togy AveToStream(oStr,TFDStorageFormat.sfJSON); Atream(oStr, 'application/json', True); 画面 - © x
<pre>// クエリの実行結果をス= // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre> 1 RAD Serverの起動 RAD Serverの起動	キーマアダプタから Tiggす AveToStream(oStr,TFDStorageFormat.sfJSON); tream(oStr, 'application/json', True); 画面
<pre>// クエリの実行結果をス // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre>	キーマアダプタから Tiggす AveToStream(oStr,TFDStorageFormat.sfJSON); tream(oStr, 'application/json', True); 画面
<pre>// クエリの実行結果をス // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre>	キーマアダプタから r返す AveToStream(oStr,TFDStorageFormat.sfJSON); tream(oStr, 'application/json', True); 画面 - © x
<pre>// クエリの実行結果をス // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre>	キーマアダプタから riggす AveToStream(oStr,TFDStorageFormat.sfJSON); tream(oStr, 'application/json', True); 画面 - © ×
<pre>// クエリの実行結果をス: // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre>	キーマアダプタから E 返す AveToStream(oStr,TFDStorageFormat.sfJSON); tream(oStr, 'application/json', True); 画面 - 0 x - 0 x - 0 x - 0 x - 0 x - 0 x - 0 x
<pre>// クエリの実行結果をス // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre>	キーマアダプタから iSzgす werToStream(oStr,TFDStorageFormat.sfJSON); tream(oStr, 'application/json', True);
// クエリの実行結果をス // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.Sets FDTable1.Close; end; 1 RAD Serverの起動 RAD Serverの起動 Comparison ENCENT	キーマアダプタから Sizeす AveToStream(oStr,TFDStorageFormat.sfJSON); tream(oStr, 'application/json', True);
<pre>// クエリの実行結果をス // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;</pre>	キーマアダプタから Sizeす AveTOStream(oStr,TFDStorageFormat.sfJSON); tream(ostr, 'application/json', True); 面面
// クエリの実行結果をス // メモリストリーム経由で FDTable1.Open; FDSchemaAdapter1.Sa AResponse.Body.SetS FDTable1.Close; end;	キーマアダブタから fixed weroStream(ostr,TFDStorageFormat.sfJSON); tream(ostr, 'application/json', True); 面面



GR +++		× =	212	
パースノン FDMemT	DIクトインスペクター ロ		LiveB	Binding テザイナ
○検索			Form1	1 - デフォルトレイヤ
プロパテ	11/21	-		
» Active	False	* ^	110	
Active	StoredUcage [auDecignTime,auRunTime	e]	4	BindSourceDB1 StringGrid1
Adapte	gates (TFDAggregates)			EDMemTable1 *
Aggree	gatesActive 🔲 False		-	
AutoC	alcFields 🛛 True		5	•••
Chang	eAlerter		000	
Chang	eAlertName	19		Arl
Constr	aints (TCheckConstraints)	~		
フィールド データヤッ	エディタ ファイルから読み込み トの割り当て データヤットの編集			
レジュアル	「パイン」に、ケーノンクトの構築… にパイン」に、ケーノンクライク			
3.1004	RED SCALLUCY 6 7	-11		
ら プ :	ラットフォーム別にコンパイル・ m1	実行	0	×
ら プラ © Fon ID	ラットフォーム別にコンパイル・ m1 NAME	実行 -	0	× phone × @ O
ら プ . © Fon ID 1221	ラットフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2	実行 -		× ↓ IPhone × ▲ ○ ■ NAME 1221 □ □ ± ##₩2111 \\2 \# ## 272
ら プラ © Form ID 1221 1513	ラットフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5	実行 -		X
6 プ: © Form ID 1221 1513 3444	ラットフォーム別にコンパイル・ m1 ココナッツマリンショップ2 ダイブハウスタートル5 ダイビングベース新井8	実行 -		X
 デー デー ID 1221 1513 3444 1231 1351 	ラットフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 申山ダイブセンター	実行 -		X
6 プ: © Fon ID 1221 1513 3444 1231 1351 1380	ラットフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター ダイブショップブルーリーフ	実行 一		X
6 プラ © Form ID 1221 1513 3444 1231 1351 1380 1384	ラットフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター ダイブショップブルーリーフ MHMダイバーズクラブ	実行 -		X
© Form ID 1221 1513 3444 1231 1351 1380 1384 1984	ラットフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター ダイブショップブルーリーフ MHMダイバーズクラブ ADVENTURE UNDERSEA	実行		★ IPhone × ○ ○ D NAME 1221 ココナッツマリンショップ2 1513 ダイブハウスタートル5 3444 ダイビングペース新井8 1231 アクアダイビングセンター 1351 亀山ダイブセンター 1351 亀山ダイブセンター 1380 ダイブショップブルーリーフ 1384 MHMダイバーズクラブ 1984 ADVENTURE UNDERSEA 2118 グリーンスポーツクラブ
 デ マー・ デ Form ID 1221 1513 3444 1231 1351 1380 1384 1984 2118 	ラットフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター ダイブショップブルーリーフ MHMダイバーズクラブ ADVENTURE UNDERSEA グリーンスポーツクラブ	実行 -		X ▶ IPhone × ○ ○ ▶ NAME 1221 ココナッツマリンショップ2 1513 ダイブハウスタートル5 3444 ダイビングペース新井8 1231 アクアダイビングセンター 1351 亀山ダイブセンター 1380 ダイブショップブルーリーフ 1384 MHMダイバーズクラブ 1984 ADVENTURE UNDERSEA 218 グリーンスボーツクラブ 218 グリーンスボーツクラブ 2156 マリンハウスペムボム
© For ID 1221 1513 3444 1231 1351 1380 1384 1984 2118 2135	ラットフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター ダイブショップブルーリーフ MHMダイバーズクラブ ADVENTURE UNDERSEA グリーンスポーツクラブ バイナップルダイバーズ	実行 -		★ IPhone × ○ ○ D NAME 1221 ココナッツマリンショップ2 1513 ダイブハウスタートル5 3444 ダイビングベース新井8 1231 アクアダイビングセンター 1351 亀山ダイブセンター 1351 亀山ダイブセンター 1380 ダイブショップブルーリーフ 1384 MHMダイバーズクラブ 1984 ADVENTURE UNDERSEA 2118 グリーンスポーツクラブ 2135 パイナップルダイバーズ 2156 マリンハウスペんざん 2163 SCUBA HEAVEN
© For ID 1221 1513 3444 1231 1351 1380 1384 1984 2118 2135 2156	 ラットフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター ペブショップブルーリーフ MHMダイバーズクラブ ADVENTURE UNDERSEA グリーンスポーツクラブ バイナップルダイバーズ マリンハウスべんぎん 	実行 一		★ IPhone × ○ ○ D NAME 1221 ココナッツマリンショップ2 1513 ダイブハウスタートル5 3444 ダイビングペース新井8 1231 アクアダイビングセンター 1351 亀山ダイブセンター 1351 亀山ダイブセンター 1380 ダイブショップブルーリーフ 1380 ダイブショップブルーリーフ 1384 MHMダイバーズクラブ 1984 ADVENTURE UNDERSEA 218 グリーンスポーツクラブ 2135 バイナップルダイバーズ 2135 バイナップルダイバーズ 2136 SCUBA HEAVEN 2165 SHANGRI-LA SPORTS CENTER 2025 ダインジェイング
E 7: E Form ID 1221 1513 3444 1231 1380 1384 1984 2118 2135 2156 2163	 ラットフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター ダイブショップブルーリーフ MHMダイバーズのラブ ADVENTURE UNDERSEA グリーンスポーツクラブ バイナップルダイバーズ マリンパウスべんぎん SCUBA HEAVEN 	実行 -		★ IPhone × ○ ○ D NAME 1221 ココナッツマリンショップ2 1513 ダイブハウスタートル5 3444 ダイビングベース新井8 1231 アクアダイビングセンター 1351 亀山ダイブセンター 1380 ダイブショップブルーリーフ 1384 MHMダイバーズクラブ 1984 ADVENTURE UNDERSEA 2118 グリーンスボーツクラブ 2135 バイナッブルダイバーズ 2156 マリンハウスペんざん 2166 SHANGRI-LA SPORTS CENTER 2156 ダイビングショップマーメイド 2353 マリンセンター製精満
E 7° 7 E Form ID 1221 1513 3444 1231 1351 1380 1384 1984 2118 2135 2156 2163 2165	ラットフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター ペイブショップブルーリーフ MHMダイパーズクラブ ADVENTURE UNDERSEA グリーンスポーツクラブ パイナップルダイパーズ マリンハウスべんぎん SCUBA HEAVEN SHANGRI-LA SPORTS CENTER	実行		★ ★ IPhone × ○ ○ D NAME 1221 ココナッツマリンショップ2 1513 ダイブハウスタートル5 3444 ダイビングベース新井8 1231 アクアダイビングセンター 1351 亀山ダイブセンター 1350 ダイブショップブルーリーフ 1384 MHMダイバーズクラブ 1984 ADVENTURE UNDERSEA 2118 グリーンスポーツクラブ 2135 バイナッブルダイバーズ 2156 マリンハウスペんざん 2163 SCUBA HEAVEN 2165 SHANGRI-LA SPORTS CENTER 215 ダイビングショップマーメイド 2353 マリンセンター巣紡満 2975 ダイブリソートババイヤ
© For ID 1221 1513 3444 1231 1351 1380 1384 1984 2118 2135 2156 2163 2165 2315	フットフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター 亀山ダイブセンター ダイブショップブルーリーフ MHMダイバーズクラブ ADVENTURE UNDERSEA グリーンスポーツクラブ バイナップルダイバーズ マリンハウスべんぎん SCUBA HEAVEN SHANGRI-LA SPORTS CENTER ダイビングショップマーメイド	実行		★ IPhone × ○ D NAME 1221 ココナッツマリンショップ2 1513 ダイブハウスタートル5 3444 ダイビングペース新井8 1231 アクアダイビングセンター 1351 亀山ダイブセンター 1360 ダイブショップブルーリーフ 1380 ダイブショップブルーリーフ 1384 MHMダイバーズクラブ 1984 ADVENTURE UNDERSEA 2118 グリーンスポーツクラブ 2135 パイナッブルダイバーズ 2156 マリンバクスペんぎん 2165 SHANGRI-LA SPORTS CENTER 2155 ダイビングショッブマーメイド 2353 マリンセンター機構満 2975 ダイブリゾートババイヤ 2984 サンセットダイビングサービス 2041 マリンショップマーズ
© For ID 1221 1513 3444 1231 1380 1384 2135 2156 2165 2315 2353 2155	フントフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター 亀山ダイブセンター ダイジショップブルーリーフ MHMダイバーズクラブ ADVENTURE UNDERSEA グリーンスポーツクラブ バイナッブルダイバーズ マリンハウスペんぎん SCUBA HEAVEN SHANGRI-LA SPORTS CENTER ダイビングショップマーメイド マリンセンター単柿浦	実行 -		★
© For ID 1221 1513 3444 1231 1351 1380 1384 1984 2118 2135 2156 2163 2165 2315 2353 2975	 ラットフォーム別にコンパイル・ MAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター ダイブショップブルーリーフ MHMダイパーズのラブ ADVENTURE UNDERSEA グリーンスボーツクラブ バイナップルダイバーズ マリンハウスべんぎん SCUBA HEAVEN SHANGRI-LA SPORTS CENTER ダイブリゾートババイヤ ロットのボーンボー 	実行		★ IPhone × ○ ▶ NAME 1221 ココナッツマリンショップ2 1513 ダイブハウスタートル5 3444 ダイビングベース新井8 1231 アクアダイビングセンター 1351 塩山ダイブセンター 1360 ダイブショップブルーリーフ 1380 ダイブショップブルーリーフ 1384 MHMダイバーズクラブ 1984 ADVENTURE UNDERSEA 2118 グリーンスポーツクラブ 2135 バイナッブルダイバーズ 2166 マリンハウスペんぎん 2163 SCUBA HEAVEN 2165 SHANGRI-LA SPORTS CENTER 2353 マリンセンター巣柿浦 2975 ダイブリゾートババイヤ 2984 サンセットダイビングサービス 3041 マリンショッブアクア 3042 HAJIMAダイビングスクール 3051 ペンションあしか
© For ID 1221 1513 3444 1231 1351 1380 1384 2118 2135 2156 2163 2165 2315 2353 2975 2984	フントフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター 尊山ダイブセンター ダイブショップブルーリーフ MHMダイバーズのラブ ADVENTURE UNDERSEA グリーンスポーツクラブ バイナッブルダイバーズ マリンハウスペんぎん SCUBA HEAVEN SHANGRI-LA SPORTS CENTER ダイビングショップマーメイド マリンセンター単跡満 ダイブリゾートババイヤ サンセットダイビングサービス コリン・マーマーマー	実行		★ Phone × ○ ○ ▶ NAME 1221 コニナッツマリンショップ2 1513 ダイブ/ウスタートル5 3444 ダイビングペース新井8 1231 アクアダイビングセンター 1353 亀山ダイブセンター 1360 ダイブショッブブルーリーフ 1384 MHMダイバーズクラブ 1984 ADVENTURE UNDERSEA 2118 グリーンスボーツクラブ 2135 バイナッブルダイバーズ 2166 マリンハウスペムぎん 2165 SCUBA HEAVEN 2165 SHANGRI-LA SPORTS CENTER 2315 ダイブショッブアーメイド 2353 マリンセンター奥橋満 2975 ダイブリゾートババイヤ 2984 サンセットダイビングカーレズス 3041 マリンショッブアクア 3042 HAIMAダイビングスール 3041 マリンショッブレク 3041 マリンショッブレク 3041 マリンショッブレク 3041 マリンショッショッブ 3041 マリンショッブレク 3042 HAIMAダイビングシスー 3041 マリンショッブレット 3041 マリンショッブレット 3042 HAIMAS/Tビング 3043 マンヨンろしか
6 7 5 Form 1D 1221 1513 3444 1231 1351 1380 1384 1984 2118 2135 2156 2165 2315 2353 2975 2984 3041	アットフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター 魯山ダイブセンター ダイジショップブルーリーフ MHMダイバーズクラブ ADVENTURE UNDERSEA グリーンスポーツクラブ バイナップルダイバーズ マリンハウスべんぎん SCUBA HEAVEN SHANGRI-LA SPORTS CENTER ダイビングショップマーメイド マリンセンター単熱海浦 ダイブリジートババイヤ サンセットダイビングサービス マリンショップアクア	実行		★ Phone × ○ ○ ○ NAME 1221 ココナッツマリンショップ2 1513 ダイブハウスタートル5 3444 ダイビングペース新井8 1231 アクアダイビングセンター 1351 亀山ダイブセンター 1351 亀山ダイブセンター 1351 亀山ダイブセンター 1380 ダイブショッブブルーリーフ 1384 MHMダイバーズクラブ 1984 ADVENTURE UNDERSEA 2118 グリーンスポーツクラブ 2135 バイナッブルダイバーズ 2165 SHANGRI-LA SPORTS CENTER 2155 ダイビングショッブマーメイド 2353 マリンセンター機構満 2975 ダイブリゾートババイヤ 2984 サンセットダイビングサッーレ 3041 マリショッブアのア 3042 HAJIMAダイビングスクール 3051 ベンションあしか 3051 ベンションあしか
C 7 5 Form ID 1221 1513 3444 1231 1351 1380 1384 1984 2118 2135 2156 2163 2165 2315 2353 2975 2984 3041	フントフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター タイブショップブルーリーフ MHMダイバーズのラブ ADVENTURE UNDERSEA グリーンスボーツクラブ バイナッブルダイバーズ マリンハウスペんぎん SCUBA HEAVEN SHANGRI-LA SPORTS CENTER ダイビングショップマーメイド マリンセッター巣柿浦 ダイブリゾートババイヤ サンセットダイビングサービス マリンショップアクア	実行		★ ★ IPhone × ○○ ○ NAME 1221 ココナッツマリンショップ2 1513 ダイブハウスタートル5 3444 ダイビングベース新井8 1231 アウアダイビングセンター 1351 亀山ダイブセンター 1351 亀山ダイブセンター 1351 亀山ダイブセンター 1380 ダイブショップブルーリーフ 1384 MHMダイバーズクラブ 1984 ADVENTURE UNDERSEA 2185 パイナップルダイバーズ 2165 マリンハウスペム ぎん 2165 SHANGRI-LA SPORTS CENTER 2165 SHANGRI-LA SPORTS CENTER 2165 SHANGRI-LA SPORTS CENTER 2155 ダイビングショップマーメイド 2353 マリンセンター樂樹湯 2975 ダイブリゾートババイヤ 2984 サンセットダイビングスクール 3051 ペンションガレカ 1051 ペンションガレカ 1051 ペンションガレカ 1051 ペンションガレカ 1051 ペンションガレカ
6 7 For 1D 1221 1513 3444 1231 1351 1380 1384 1984 2118 2135 2156 2163 2165 2315 2353 2975 2984 3041	フントフォーム別にコンパイル・ m1 NAME ココナッツマリンショップ2 ダイブハウスタートル5 ダイブハウスタートル5 ダイビングベース新井8 アクアダイビングセンター 亀山ダイブセンター 魯山ダイブセンター 菊山ダイブショップブルーリーフ MHMダイバーズクラブ ADVENTURE UNDERSEA グリーンスポーツクラブ バイナッブルダイバーズ マリンハウスペんぎん SCUBA HEAVEN SHANGRI-LA SPORTS CENTER ダイビングショップマーメイド マリンセットダイビングサービス マリンショップアクア 裏示	実行		★ ★ IPhone × ● ● ● NAME 1221 ココナッツマリンショップ2 1513 ダイブハウスタートル5 3444 ダイビングベース新井8 1231 アクアダイビングセンター 1351 亀山ダイブセンター 1351 亀山ダイブセンター 1351 亀山ダイブセンター 1351 福山ダイブレンター 1360 ダイブショップブルーリーフ 1384 MHMダイバーズクラブ 1984 ADVENTURE UNDERSEA 2185 グイブッコンダイバーズ 2156 マリンハウスペんぎん 2165 SHANGRI-LA SPORTS CENTER 2155 ダイビングショップマーメイド 2353 マリンセンター県紡満 2975 ダイブリンートババイヤ 2984 ウンセットダイビングカービス 3041 マリンショップアクア 3042 HAJIMAダイビングスクール 3051 ペンションガレか ■ </td

S Configuaded : (Flename 'T C: 4 Users MP Juble NDocuments MEmbarcader of MEMS Nemser ver. in", "Thread '10856)); (Configuaded : (Flename 'T Flename 'T : 1: 4 Users MP Juble NDocuments MEmbarcader vBEMS Nemser ver. in", "Thread '10856)); (Configuaded : (Flename 'T : C: 4 Users MP Juble NDocuments MEmbarcader vBEMS Nemser ver. in", "Thread '10856)); (Configuaded : (Flename 'T : C: 4 Users MP Juble NDocuments MEmbarcader vBEMS Nemser ver. in", "Thread '10856)); (Configuaded : (Flename 'T : C: 4 Users MP Juble NDocuments MEmbarcader vBEMS Nemser ver. in", "Thread '10856)); (Configuaded : (Flename 'T : C: 4 Users MP Juble NDocuments MEmbarcader vBEMS Nemser ver. in", "Thread '10856)); (Configuaded : (Flename 'T : C: 4 Users MP Juble NDocuments MEmbarcader vBEMS Nemser ver. in", "Thread '10856)); (Configuaded : (Flename 'T : C: 4 Users MP Juble NDocuments MEmbarcader vBEMS Nemser ver. in", "Thread '10856)); (Configuaded : (Flename 'T : C: 4 Users MP Juble NDocuments MEmbarcader vBEMS Nemser ver. in", "Thread '10856)); (Configuaded : (Flename 'T : C: 4 Users MP Juble NDocuments MEmbarcader vBEMS Nemser ver. in", "Thread '10856)); (Configuaded : (Flename 'T : Kesource 'C : Cust') Endpoints 'T : Get 'T : Center (Flename T); 'Thread '10856); (Configuaded : (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Get 'T : Center (Flename T); 'Endpoints 'T : Center (Flename		停止	ブラウザを開く	コンソールを開く			
a Trade D7: [ConfigLoaded': ("Fiename': "C:HUsersHPubleHDocumentsHEmbarcaderoHEMSHemsserver.in", "Thread':10356)} [Decome: ChUsersHPubleHDocumentsHEmbarcaderoHEMSHemsserver.in", "Thread':10356)} [Decome: C:HUsersHPubleHDocumentsHEmbarcaderoHEMSHemsserver.in", "Thread':10356)} [ConfigLoaded': C:HUsersHPubleHDocumentsHEmbarcaderoHEMSHemsserver.in", "Thread':10356)} [ConfigLoaded': C:HUsersHPubleHDocumentsHEmbarcaderoHEMSHemsserver.in", "Thread':10356)} [ConfigLoaded': C:HUsersHPubleHDocumentsHEmbarcaderoHEMSHemsserver.in", "Thread':10356)} [ConfigLoaded': C:HusersHPubleHDocumentsHEmbarcaderoHEMSHemsserver.in", "Thread':10356)] [ConfigLoaded': C:HusersHPubleHDocumentsHEmbarcaderoHEMSHemsserver.in", "Thread':10357] [ConfigLoaded': C:HusersHPubleHDocumentsHEmbarcaderoHEMSHemsserver.in", "Thread':10356)] [ConfigLoaded': C:HusersHPubleHDocumentsHEmbarcaderoHEMSHemsserver.in", "Thread':10357] [ConfigLoaded': C:HusersHPubleHDocumentsHEmbarcaderoHEMSHemsserver.in", "Thread':10357] [ConfigLoaded': C:HusersHPubleHDocumentsHEmbarcaderoHEMSHemsserver.in", "Thread':10357] [ConfigLoaded': C:HusersHPubleHDocumentsHEmbarcaderoHEMSHemsserver.in", "Thre	-N8080						
9 RAD Server(LLS3) fitted with the set of	л. 1 <u>0000</u>						
8 コンソールのロダイン 8 コンソールのロダイン ************************************	{"ConfigLo	aded":{"Filename":"C:	¥Users¥Public¥Docum	ents¥Embarcadero¥EM	S¥emsserver.ini", "Thr	ead":10856}}	
8 コンソールのログイン 8 コンソールのログイン 9 Decent Light State St	{"DBConnel {"Licensinel	ction":{"InstanceName ":{"Licensed":false."De	:":"","Filename":"C:¥U efaultMaxUsers":5,"Th	Jsers¥Public¥Document hread": 10856}}	s¥Embarcadero¥EMS	emsserver.ib", "Threa	ad": 10856}}
8 コンソールのログイン ************************************	{"Loading"	:{"Filename":"C:¥Users	;¥Public¥Documents¥8 JST", "Endpoints": ["Ge	Embarcadero¥Studio¥1 t"."GetItem"]."Thread"	9.0¥Bpl¥Project1.bpl' :10856}}	,"Thread": 10856}}	
8 コンノールのログイン ************************************	1.1.2007.002	1		· (
8 コンノールのログイン Image: A market Marke							
8 コンソールのログイン ************************************							
9 RAD Server(LJSO) file in the file in	18 コンソー	-ルのロクイン					
9 RAD Server(L\$30/http://////////////////////////////////	Embaro	Adero MEAP Sol × + ×				- 0	×
9 RAD Server(L±30) filler (?)			_			001/20	1
9 RAD Server(Lt30) frite to the server of		Amba	サインイン		×		
9 RAD Server(L\$30) fill to the fill of the		CITICA	consoleuser				
9 RAD Server(L\$a\$) file (1)		エンタープライズモビ	[l		•		
9 RAD Server(こよる分析機能			Enter tenant socret				
S RAD Serverによる分析機能		D 2016 Embarcadera Technologies, Inc.	ロダイン				
デフォルトは consoleuser/consolepass 9 RAD Serverによる分析機能 **********************************							
9 RAD Server/こよる分析機能					デフォル	レトは	
9 RAD Serverによる分析機能 ************************************					consoleuser/	consolepass	
9 RAD Serverによる分析機能 ************************************							
9 RAD Serverによる分析機能 ************************************							
9 RAD Serverによる分析機能 * * * * * * * * * * * * * * * * * * *							
9 RAD Serverによる分析機能 ************************************							M
9 RAD Serverによる分析機能 ● 本 P (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)							8
 ▲ AN FORMULE/KROP × + v - ロ × ← → O @ isoahost 000 (Microsoblecality/strained) ▲ 本 L @ ··· ← → O @ isoahost 000 (Microsoblecality/strained) ▲ 本 L @ ··· ← → O @ isoahost 000 (Microsoblecality/strained) ▲ AN FORMULE/KROP × + v - 09/02/88 AN FORMULE/KROP × + v - 09/02/88 - 09/02/88							
 ← → ○ @ ○ locahost 000 (Initious Allocabins Street i → 3 ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲	19 RAD S	erverによる分析機					
APTIFUCIOLI DE DE PERFUSIO 20183/988(0) E 50000 BIECOREZO (CONDUCTORING DE PERFUSIO) © 271097020 BIECOREZO (CONDUCTORING DE PERFUSIO) • 07170100000 BIECOREZO (CONDUCTORING DE PERFUSIO) • 071701000000 BIECOREZO (CONDUCTORING DE PERFUSIO) • 071701000000 BIECOREZO (CONDUCTORING DE PERFUSIO) • 07170100000000000000000000000000000000	19 RAD S	erverによる分析機	能	 X ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● <lp>● </lp> <td>+ ~</td><td></td><td>- 5 X</td>	+ ~		- 5 X
	19 RAD S	erverによる分析機 JJF#f7+× + ~ ○ boahost 001/should calify for	能 	0 X 10 K 12 ···· 1 • • • • • • • • • • • • • • • • • • •	+ v estimateika 2 Antipatu Bit At the		× 6 - - \$ 1 \$ \$
	19 RAD S ● 45 AN FF0 E ← → ○ @ ■ 3-9- ● 70-7 ■ 72-7-0 ■ 52-7-0	erverによる分析機 ^{IDF#17+× + 〜} © Iocalhost 001/01/0040/Califordio API 呼び出しエンド ス amongstrates	能 ** □ * * * * *ポイント アナリティ	□ x w e w e + > 0 x + >	+ v. etwistech 2 wrgzz 28 A8 R8	20443/1463 (c)	
・ ・	19 RAD S ・ ・ 〇 @ ニュージー ビ クループ ロ インストール 田 Responses 9 アクションクロープ	erverによる分析機 Tyřefyh × + ~ o boshost 601 (Mirolauhochdrofo API 呼び出しエンド ス。 atticatzintee Ecov 797 (2000)	能 ** □ ☆ ☆ & *ポイント アナリティ	□ x e = 48556-13-71 x e2 € → 0 @ 0 bod • • • •	+ v etstruktusoon 2 AMADSU 28 A9 48	201823/1428 (c)	× 5 - 5 ×
BECOST - Gen	19 RAD S	erverによる分析機 DF#f7F× + ~ ○ boshost 800 (MirroutAncalatorio API 呼び出しエンド ス attributes 500 (MirroutAncalatorio API 呼び出しエンド ス attributes 500 (MirroutAncalatorio API 呼び出しエンド ス attributes 500 (MirroutAncalatorio API 呼び出しエンド	能 ** □ ☆ ☆ & *ポイント アナリティ ************************************	□ x € € #1958-2577 x € → 0 â 0 • - 57072148 1 • - 97072148 1 0 • - 97072148 1 0 • - 97072148 1 0 • - 97072148 1 0 • - 97072148 1 0 • - 97072148 1 0 • - 9 0 0 • - 9 0 0 • - 9 0 0 • - 9 0 0 • - 9 0 0 • - 9 0 0 • - 9 0 0 • - 9 0 0 • - - 0 0 • - - 0 0 • - - 0 0 • - - 0 0 • - - 0 • - 0	+	201483/1982) (12)	× 5 - × - 5 1 2 =
	19 RAD S	erverによる分析機 エンド#17+× + × ○ tosahost 800 (MinfoldAnCaldrodio API 呼び出しエンド ス 副前にの目21時間 こ API 時(3世しエンドボイント: 目間 こ API 時(3世しエンドボイント: 目間 こ 2018	能 ** ① ★ ★ & *ポイント アナリティ ************************************	x e </td <td></td> <td>(c) (c) 2008205020 2009205020 2009205020 2009205020 2009205020 2009205020 2009205020 2009205020 2009205020 2009205020 2009205020 2009205020 2009200 2009200 200000000</td> <td>- 5 X</td>		(c) (c) 2008205020 2009205020 2009205020 2009205020 2009205020 2009205020 2009205020 2009205020 2009205020 2009205020 2009205020 2009205020 2009200 2009200 200000000	- 5 X
* 2-9-5) 80990000	19 RAD S ・ ・ 0 0 ・ ・ 0 0 ・ ユーザー ダクループ ロインストール ■ EdgetHedue ■ Resources ゆ アナリタイクス ・ クライアンド線 ・ ARFT(78k) ・ メポイント ゆ Users Angl(2k) ・ ・ ボイント ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	erverによる分析機 JンF# f2h × + ~ © boolhost 001/shr0adAnCabiludio API 呼び出しエンド ス ase:off2tHitss © 000000000000000000000000000000000000	能 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	□ × 12 ···· (2) ···· (3) ···· (4) ···· (2) ····· (3) ····· <td></td> <td>20083/9802 (Q)</td> <td>- 3 X</td>		20083/9802 (Q)	- 3 X
2	19 RAD S ● セ An 時び出 ← → ○ @ ■ ユーザー ● クループ ■ インストール ■ Edgethedue ■ RAP (2021) ● クループ ■ インストール ■ Edgethedue ■ An 時び出し ● An 時び出し思 ● ユーザーの ● 出ザーの ● 出げの山尾	erverによる分析機 LVF#f7+× + ~ © boahost 001/04/04/04/04/04/04/04/04/04/04/04/04/04/	# # # # # # # # # # # # # #	Image: Second			
	19 RAD S	erverによる分析機 エアドモクト× + ~ ● boshoot 1041-0440/Andrews API 呼び出しエンド ス == 20100000 こ API 呼び出しエンド こ 2 API 呼び出しエンド 2 2010 = 20100000 こ API 呼び出しエンド 2 2011 = 20100000 こ API 呼び出しコンドボイント・ 目間 2 2011 = 2010 =	能 ** □ ★ ★ £ *ポイント アナリティ ************************************	□ X (2) (2) (3) (2) (3) (2) (3) (2) (3) (2) (3) (2) <td></td> <td>200833/982 (12)</td> <td>× t -</td>		200833/982 (12)	× t -
	19 RAD S ■ 40 APF020 ← → ○ 0 0 = ユ-サ- ● クループ □ インストール ■ Resources ○ アナリタイクス = クライアンド間 ● APF02020 → APF02020 → ユーザーの APF02020 → ユーザーの APF02020 → ユーザーの APF02020 → ユーザーの APF02020 → ユーザーの	erverによる分析機 エンド#17+× + ~ ○ bookhot 100 UNATOWARCabilution API 呼び出しエンドス 副前の記になっドポイント: 目間 2010 2011 第一回1000000 2011 第一回1000000 2011 第一回1000000 2011 第一回1000000 2011 100000000	:能 :ポイントアナリティ : #2	X X	+ v etryshowch 2 wrgozu 28 As te 	(0) (2018/3/302) (0) (2018/3/302) (0) (2019/3/3/302) (0) (2019/3/3/302) (0) (2019/3/3/302) (0) (2019/3/3/302) (0) (2019/3/3/302) (0) (2019/3/3/302) (0) (2019/3/3/3/302) (0) (2019/3/3/3/3/3/3/3/3/3/3/3/3/3/3/3/3/3/3/3	- 5 ×



Refresh	H Myviewport ×	My ViewportView	× tgj MyJsonSto	D	code grid		36
-					DEFA	ULT (7) C Grid	
My Grid I	Panel				Behar Butto	viors (0)	ind
String	Number	Date	Boolean		Chart	s (0) Grid Augi	ins ter
cell	10,000.00	05/19/2018	true		Data	(0) Filters	Plugin
cell	10,000.00	05/19/2018	true		Form		*= o
cell	10,000.00	05/19/2018	true			ligs_	
				My Gr	id Ranel	18	80 / 180
					/	P	
		\subset		TDBGrid(こ	以た表形式の		
				GridPanelコンホーネ	ノトを画面に	配直する	
2 Grid F	Panelのデ-		2				
		♪ ୬ mixxx					
Grid Builder							\oplus \otimes
Auto-opperate							
Auto generati	a grid view's column	ns, store, model, edito	ors, controller actions	d mock data			
	a grid view's column	ns, store, model, edito	ors, controller actions	d mock data			1
Data Source	e grid view's column e Choose	a data source type when	re the data will be load	form	fields will be used to create	this view. Please	
Data Source	e Choose Choose Ne	a data source type when w Model isting Model ON Web Service	re the data will be load	from Success: The following	fields will be used to create	this view. Please	
Data Source	e Choose Choose Ex S JS JS	a data source type when w Model isting Model ON Web Service ON Object	rs, controller actions	from Success: The following	fields will be used to create	^{this view Please} Service形式	
Data Source Model Nam	e Choose Choose States	a data source type when w Model isting Model ON Web Service <u>ON Object</u> class name for this mo	re the data will be load	from Success: The following	Telds will be used to create	^{this view Please} Service形式	
Data Source Model Nam	e Choose - Choose - Ex S JS JS e Enter the MyMod	is, store, model, edito a data source type wher w Model isting Model ON Web Service <u>ON Object</u> is class name for this mo	re the data will be load	from Success: The following	fields will be used to create	this view Please Service形式	
Data Source Model Nam	e Choose Choose Choose State S	a data source type when w Model isting Model ON Web Service ON Object e class name for this mo fel	re the data will be load	from Success: The following	Telds will be used to create	^{this view. Please}	
Data Source Model Nam	e Choose - Choose - Ex S JS JS e Enter the MyMoo Enter the http://fo	a data source type wher w Model isting Model ON Web Service <u>ON Object</u> e class name for this mo fet uRL of this service. * calihost 8080/CUST	re the data will be load	from Success: The following	fields will be used to create	this view Please Service形式	
Model Nam	e Choose - Choose - Ne Ext JS JS e Enter the MyMoo Enter the http://io	is, store, model, edito a data source type when w Model isting Model ON Web Service ON Object e class name for this mo fel cuRL of this service. * calihost 8080/CUST	re the data will be load	trom	JSON Web	this view Please Service形式 verのURL	
Data Source Model Nam	e Choose e Choose Ex S JS JS e Enter the http://io	s, store, model, edito a data source type when w Model isting Model ON Web Service ON Object e class name for this mo fel euRL of this service. * icalhost 8080/CUST proot data property if ne	re the data will be load odel * eeded (optional)	from Success: The following	JSON Web	this view Please Service形式 verのURL	
Data Source Model Nam URL Root Prope	e Choose - e Choose - Ne Ext y JS e Enter the http://fo	is, store, model, edito a data source type when w Model isting Model ON Web Service ON Object e class name for this mo- fel e URL of this service. * icalihost 8080/CUST e root data property if ne	re the data will be load odel * eeded (optional)	d mock data	Index will be used to create	this view Please Service形式 verのURL	
Data Source Model Nam URL Import Fields	e Choose - Oracies - Oraci	Is, store, model, edito	re the data will be load odel *	trom Success: The following	Telds will be used to create JSON Web RAD Ser ドの自動取り	this view Please Service形式 verのURL 込みも可能	
Data Source Model Nam URL Import Fields	e Choose Choose Sea grid view's column e Choose Sea Sea Sea Sea Sea Sea Sea Sea Sea Se	Is, store, model, edito a data source type when w Model isting Model ON Web Service ON Object e class name for this mo fel e URL of this service. * calhost 8080/CUST e rool data property if ne	re the data will be load odel * ceded (optional)	trom Success: The following rectance 7r-JU and mock data.	Telds will be used to create JSON Web RAD Ser ドの自動取り	this view Please Service形式 verのURL 込みも可能	
Data Sourc Data Sourc Model Nam URL Import Field Field Name custrio	e Choose - Ne Extension - Ne Extensio - Ne Extensio - Ne Extension - Ne Extension - Ne Extensio	Is, store, model, edito a data source type when w Model isting Model ON Web Service ON Object e class name for this mo fel URL of this service. * icalhost 8080/CUST or root data property if ne	re the data will be load odel * ceded (optional) mns, editors, model fiel Friendly Te Custno	trom Success: The following (Epine 71-JL and mock data.	Telds will be used to create JSON Web RAD Serr ドの自動取り	this view Please Service形式 verのURL 込みも可能	
Data Sourc Data Sourc Model Nam URL Import Fields Field Name custno company	e Choose - Choose - Ne Ext JS e Enter the MyMod Enter the http://fo ty Enter the Used to	Is, store, model, edito	re the data will be load re the data will be load odel * eeded (optional) mms, editors, model fiel Friendly Te Custno Company	trom Success: The following restriction 71-JL and mock data.	Telds will be used to create JSON Web RAD Ser ドの自動取り	this view Please Service形式 verのURL 込みも可能	
Data Source Data Source Model Nam URL URL Import Fields Field Name custrio company addr1	e Choose - P Choose - P Choose - P Exter the MyMoo Enter the http://fo rty Enter the Used to	Is, store, model, edito	re the data will be load re the data will be load odel * ceded (optional) mns, editors, model fiel Friendly Te Custno Company Addr1	trom Success: The following certification 71-JL and mock data.	Tekts will be used to create JSON Web RAD Ser ドの自動取り Type String String String	this view Please Service形式 verのURL 込みも可能	
Data Sourc Data Sourc Model Nam URL URL Import Fields Field Name custno company addr1	e Choose - Ne Extension - Ne Extensio - Ne Extensio - Ne Extension - Ne Extension - Ne Extensio	Is, store, model, edito	re the data will be load re the data will be load odel * reded (optional) mns, editors, model fel Friendty Te Custno Company Addr1	trom Seccess: The following Control of the fol	Tekks will be used to create JSON Web RAD Ser ドの自動取り Type String String String	this view Please Service形式 verのURL 込みも可能	
Data Sourc Data Sourc Model Nam URL Import Fields Field Name custrio company addr1	e Choose - Ne Ex e Choose - Ne Ex y JS e Enter the MyMod Enter the http://io	Is, store, model, edito	eeded (optional) mins, editors, model fiel Friendty Te Custno Company Addr1	trom Seccess: The following rest	Telds will be used to create JSON Web RAD Ser ドの自動取り	this view Please Service形式 verのURL 込みも可能	
Data Source Data Source Model Nam URL URL Import Fields Field Name custno company addr1	e Choose - Ne Existence - Ne Existe	Is, store, model, edito	re the data will be load odel * ceded (optional) mns, editors, model fiel Friendly Te Custno Company Addr1	trom Success: The following certified 71-JL and mock data.	Tekts will be used to create JSON Web RAD Ser ドの自動取り Type String String String	this view Please Service形式 verのURL 込みも可能	
Data Sourc Data Sourc Model Nam URL URL Import Fields Field Name custno company addr1	e Choose Ne Extended a grid view's column e Choose Ne Extended a JS e Enter the MyMoo Enter the http://lo	Is, store, model, edito	re the data will be load re the data will be load odel * reeded (optional) mins, editors, model fiel Friendly Te Custno Company Addr1	trom Seccess: The following Trom Carlos Trom Carlos And mock data.	Tekks will be used to create JSON Web RAD Ser ドの自動取り	this view Please Service形式 verのURL 込みも可能	
Data Sourc Data Sourc Model Nam URL Import Fields Field Name custrio company addr1	e Choose - Ne Existence State	Is, store, model, edito	re the data will be load re the data will be load odel * eeded (optional) mms, editors, model fiel Friendly Te Custno Company Addr1	trom	Telds will be used to create JSON Web RAD Ser ドの自動取り	this view Please Service形式 verのURL 込みも可能	
Data Source Model Nam URL URL Import Fields Field Name custno company addr1	e Choose - Ne Extended States - States	Is, store, model, edito	re the data will be load re the data will be load odel * ceded (optional) mns, editors, model fiel Friendly Te Custno Company Addr1	trom	Tekts will be used to create JSON Web RAD Ser ドの自動取り	this view Please Service形式 verのURL 込みも可能	
Data Sourc Model Nam URL Root Prope Import Field Field Name custrio company addr1	e Choose Ne Existence Ne Ne Ne Ne Ne Ne Ne	Is, store, model, edito	re the data will be load re the data will be load odel * reded (optional) mis, editors, model fiel Friendly Te Custno Company Addr1	trom Seccess: The following Trom Seccess: The following Trom Trom	Tekks will be used to create JSON Web RAD Ser ドの自動取り	this view Please Service形式 verのURL 込みも可能	
Data Sourc Model Nam URL Import Fields Field Name custrio company addr1	e Choose - Ne Ex S JS e Enter the MyMod Enter the http://io	Is, store, model, edito	re the data will be load re the data will be load odel * eeded (optional) mns, editors, model fiel Friendly Te Custno Company Addr1	trom Seccess: The following THE Seccess: The following THE Seccess: The following THE Seccess: The following Seccess: The following Secce	Telds will be used to create JSON Web RAD Ser ドの自動取り	this view Please Service形式 verのURL 込みも可能	


吉原 泰介 / 國元 祐二

RAD事業部 技術支援課

SP4i 初級

【SmartPad4i】 JC/400からSP4iへのマイグレーションノウハウ

はじめに
 システム環境・設定の違い
 プログラムのマイグレーションポイント
 プロジェクトのマイグレーション
 3-2. IBM i プログラムのマイグレーション
 3-3. 画面プログラムのマイグレーション
 マイグレーションに伴う拡張性
 補足:自動ログオン機能の強化
 おわりに



 略歴
 吉原泰介

 1978年3月26日生まれ

 2001年3月龍谷大学法学部卒業

 2005年7月株式会社ミガロ、入社

 2005年7月システム事業部配属

 2007年4月RAD事業部配属

現在の仕事内容 Delphi/400を中心に製品試験およ び月 100 件に及ぶ問い合せサポート やセミナー講師などを担当している。



略歴 國元 祐二 1979年3月27日生まれ 2002年3月追手門学院大学 文 学部アジア文化学科卒業 2010年10月株式会社ミガロ、入社 2010年10月RAD事業部配属

現在の仕事内容

SmartPad4i(JC/400)、Business4 Mobile、Valenceの製品試験やサ ポート業務、導入支援などを行って いる。

1.はじめに

最近のWebアプリケーションはC/S アプリケーションと近い機能や操作性を 持ち、基幹システムの一部として利用さ れることも多くなっている。C/Sアプ リケーションと比べて便利な点は、PC にあらかじめ搭載されたWebブラウザ で動作するため、運用環境の構築が非常 に容易な点である。

JC/400 は、そうした Web アプリケー ションを IBM iの RPG を中心に開発で きるツールとして実績がある。

しかし Web アプリケーションもここ 数年で大きく環境が変わってきている。 それは Web ブラウザが多様化したこと である。

以前は Windows にインストールされ ている Internet Explorer (以下、IE) が標準 Web ブラウザとして使われるこ とが圧倒的に多く、Web アプリケーショ ンも IE の動作を基準としていた。しか し 最 近 は、Google Chrome (以下、 Chrome) や Safari、Firefox などさま ざまな Web ブラウザが使われるように なっている。【図 1】

これはモバイルの普及が大きく影響 しており、単純に Windows を標準とし た Web ブラウザよりも、モバイルを含 めたさまざまな OS に対応した高機能な Web ブラウザが標準となってきている。

そのため、Web アプリケーションも 各種 Web ブラウザに対応した動作を求 められることが多くなってきた。いわゆ るクロスブラウザ対応である。

JC/400 の Web アプリケーションの 動作環境は、従来の IE に限定されるが、 こうした背景に合わせた Web アプリ ケーションへの対応方法が用意されてい る。 それは JC/400 の後継 である SmartPad4i(以下、SP4i)へのマイグ レーションである。

SP4i では前述のクロスブラウザ対応 に加え、モバイルでもハイブリッド Web アプリケーションとして使用可能 である。

モバイルでは、カメラや GPS などの ネイティブ機能も活用することが可能で ある。もちろん SP4i は JC/400 の後継 であるため、仕組みとしては大きく変わ らず、ほとんどの部分は SP4i がアーキ テクチャの違いを吸収してくれる。その ため、マイグレーションでは JC/400 の プログラムを若干手直しすることで、そ のまま移行できる。

本稿では、JC/400 から SP4i ヘプロ グラムのマイグレーションを題材に手順 やポイントを説明する。

*マイグレーション対象の SP4i は 2018 年 8 月 時 点 で 最 新 バ ー ジョ ン の SmartPad4iV2.1.8D とする。



JC/400 から SP4i へのプログラムの マイグレーションを行う前に、環境の違 いを把握しておく。RPG を中心として アプリケーションを開発・実行できる大 きな仕組みは SP4i でも変わらないが、 サブシステム名やポートなどは製品が異



なるため、刷新されている。(* SP4i の初期バージョンでは JC/400 と共通の 部分もある)

まず、サブシステムはログインメ ニューとアプリケーションの2つが稼働 しており、次のような違いがある。【図2】

●ログインメニューのサブシステム
 CO405JTCP → CO406JTCP

●アプリケーションのサブシステム JACI400 → SP4I

実際のプログラムでこのサブシステ ム名の違いを考慮する必要はないが、実 行するジョブの確認などで重要になって くるので、新しいサブシステム名を把握 しておく必要がある。

また接続するポート番号も 19003 か ら 19004 に 変 更 と なって い る の で、 Web サーバー環境や Designer (後述) の接続設定では注意していただきたい。

これらの新しい環境を前提に、次の章 からはプログラムのマイグレーションに ついて詳しくポイントを説明する。

3.プログラムのマイグ レーションポイント

JC/400 のプログラムを SP4i のプロ グラムにマイグレーションするにあたっ て、プログラムを3つのカテゴリに分け てポイント整理する。

①プロジェクト

プログラム

- アプリケーション全体の構成や設定

② IBM i プログラム

- 環境設定の CL プログラムやメイン となる RPG プログラム

③画面プログラム - 画面を構成する HTML や JavaScript

これら3つのカテゴリのプログラム が JC/400 のアプリケーションを構成し ているので、これらに若干の変更を加え れば SP4i のアプリケーションとして使 用可能である。 本章では、この3つをカテゴリごと に変更ポイントとしてまとめている。

3-1.プロジェクトのマイグレーション

アプリケーションのプロジェクトに は全体のソース構成や設定が保存されて いる。JC/400 では [JC/400 Designer] というツールを使ってプロジェクトを作 成したり、IBM i への RPG プログラム の自動生成を行うが、SP4i でも同様の ツール [SmartPad4i Designer] が用 意 されている。この [SmartPad4i Designer] を使って JC/400 の既存プロ ジェクトを読み込み、再配布を行う。 詳しい手順は次のとおりである。

SmartPad4i Designerの操作手順

①プロジェクトを読み込む

SmartPad4i Designerの「ファイル」 メニューから拡張子が jdp のプロジェク トファイルを選択して読み込む。【図 3】

②配布の接続設定を行う

読み込んだプロジェクトの設定は JC/400 のままになっているため、接続 はキャンセルして、接続設定のポート番 号を変更する。ポート番号は前章で述べ たとおり 19004 になる。【図 4】

③配布先の設定を行う

SmartPad4i Designer の「オプショ ン」メニューから HTML ファイルのパ スと WEB サーバーの配布先のパスを設 定する。【図 5】

HTML ファイルパスは SP4i の場合、 次のようなパスになるので変更が必要で ある (SmartPad4i の製品パスが含まれ る)。

「Web サ ー バ ー \htdocs\ja_JP\ smartpad4i\html\ ライブラリ名」

WEB サーバーへの配布ルートの選択 については、JC/400 と同じであれば変 更の必要はない。

④配布を行う

SmartPad4i Designer の「配布」メ ニューから IBM i へ配布を行い、既存 の RPG を更新する。【図 6】

同様に「配布」メニューから WEB サー バーへ配布を行う。【図 7】

この作業によって、JC/400で構成さ

れている RPG プログラムを SP4i の形 式へ自動的に組み換えることができる。 これでプロジェクトの変更は完了であ る。

3-2.IBM iプログラムのマイグレー ション

次に、IBM iのプログラムについて SP4iで変更すべき点を確認する。IBM i のプログラムは CL と RPG の2つで構 成されている。

①環境設定用の CL プログラム

CL プログラムでは、実行時にライブ ラリリストを設定している。JC/400 で は JACI400DEV、JACI400 というライ ブラリを使用しているが、SP4i では SP4I というライブラリに変わるため、 この記述を変更する必要がある。【ソー ス1】

XXXLIB はプロジェクトのライブラ リを指す。

②メインの RPG プログラム

RPG プログラムのロジックは変更す る必要がなく、定型的な変更作業になる。 これは JC/400 で用意されている RPG 上の API やフィールド変数の名前が SP4i 用に変わっているために行う作業 である。ただし、3-1 の作業で基本的に は自動で変更が適用されているので、ほ とんど変更の必要はない。

プログラムの内容によっては自動変 換できない部分も残ってしまうが、コン パイルすると必ずエラーになるため、エ ラーになった部分を定型的に置き換える 作業となる。API やフィールド変数は 数が多いため、ソースの変更例ではなく、 リストとして【図8】【図9】【図10】に まとめているので作業時に参考いただき たい。

コンパイルが無事通れば、IBM i プロ グラム変更は完了である。

3-3.画面プログラムのマイグレー ション

画面プログラムはHTMLとJava Scriptで構成される。独自に作り込ん でいる画面プログラムは自由度が高いた め、本稿ですべての変更点を挙げること

図4	接続設定
<u>×</u> 14	1 ★ 1 1 1 1 1 1 1 1 1 1 1 1 1
	CCSIDの選択: 5026 - Japan Katakana/Kanji (extended) 文字コード(for DBCS): Shift_JIS □ 128桁のパスワード 接続確認 Select Cancel
図5	接続設定
	SP4i Designer - C:¥Program Files (x86)¥IBM¥HTTPServer¥b************************************
	IBM iの選択 HTMLフ WEBサーバーへの配布ルートを選択 プロジェクトの詳細 WEBサーバー配布先のパス 現在のHTMLファイル: WEBサーバー配布先のパス
	Root Used? HTML Type HTML ID IBM SMP010.HTML SMP011.HTML SMP020.HTML

はできないが、SP4i に直接関連する関 数の使用について説明する。

JC/400 利用時に JavaScript を記述し ている場合、画面の項目値(要素)を取 得 す る 関 数 と し て、document.get ElementById 関 数、document.get ElementsByName 関数を使用すること が多い。この関数を SP4i で使用する場 合は、それぞれ、SP4i.getElementById 関数、SP4i.getElementsByName 関数 として変更する必要がある。【ソース 2】 【ソース 3】

また画面プログラムについては、基本 的に変更を必要としないが、IE に限定 した JavaScript の機能(たとえば ActiveX など)を独自に使っている場 合は、ほかの Web ブラウザでは機能し ないので注意が必要である。

3-1 ~ 3-3 の 修 正 が 完 了 す れ ば、 JC/400 からマイグレーションした SP4i のアプリケーションが完成である。

これにより、今まで IE だけで利用し ていた Web アプリケーションが、他の Web ブラウザでも実行できるようにな る。【図 11】は、IE で実行する JC/400 のアプリケーションの例、【図 12】が SP4i ヘマイグレーションして、Chrome や Edge で実行した例である。

SP4i で実行すれば IE はもちろん、 Windows10 に搭載の新 Web ブラウザ Edge、Chrome、Safari、Firefox 等、 企業で使用されるであろう、ほぼすべて の Web ブラウザで利用することができ る。

4.マイグレーションに 伴う拡張性

前章までの内容で基本的なマイグ レーションは完了である。SP4iではIE 以外のWebブラウザで実行できるため、 クロスブラウザ対応ができたことになる が、メリットはそれだけではない。SP4i では、HTML5 に対応しているため css3を利用してデザインが可能となり、 レスポンシブデザインで画面も実装でき る。【図 13】

レスポンシブデザインとは1つの HTMLからデバイスの画面サイズに合 わせて複数の見え方で表現する手法であ る。

また jQuery などのオープンソースを

組み込んだカスタマイズを行うこともで きる。

jQueryとは、アメリカのプログラマー John Resig (ジョン・レッシグ)氏によっ て開発・公開された JavaScript 用のラ イブラリである。jQuery は著作権表示 を消さなければ、商用・非商用を問わず、 誰でも自由に利用することができるメ ジャーなオープンソースである。 jQueryを使うメリットは大きく2つあ る。

- I. jQuery の JavaScript を使うとコー ディング量が減らせる
- II. jQuery に対応したオープンソース の部品が利用できる

I については、JavaScript で複数行 にわたるソースコードも、jQuery では 1 つのメソッドで実現できる場合も多 く、ソースコードを簡略化することがで きる。

II については、jQueryを利用したオー プンソースの部品も Web 上には多く公 開されており、たとえば 「OVERSCROLL.JS」というオープン ソースを利用すると、ヘッダーを固定し たままスクロールできる便利なサブファ イルの表部品を利用可能である。【図 14】【図 15】

あくまで jQuery の一例ではあるが、 活用すると使いやすい画面を労なく作成 できる。jQuery などを使った詳しい拡 張方法については、2015 年のテクニカ ルレポート No.8 に掲載されている「ス マートデバイス開発で役立つ 画面拡張 テクニック」を参照いただきたい。

このように SP4i にマイグレーション したアプリケーションでは、使用できる Web ブラウザの種類が増えるだけでな く、新しい画面設計や機能拡張にも対応 できるメリットがある。

5.補足:自動ログオン 機能の強化

最後に JC/400 から SP4i で強化され た環境の違いとして自動ログオン機能に ついて補足する。自動ログオン機能とは、 ログオン画面のユーザー/パスワードの 入力を省略してアプリケーションを使用 できる機能である。SP4i ではこの自動 ログオンの機能が便利に強化されてい る。

自動ログオン機能を使用する場合、ロ グオンの入力を省略する代わりに、固定 のアカウントが使用される。

JC/400 では、signon.txt というアカ ウントファイルで固定のユーザープロ ファイルを指定できる。これで自動ログ オン可能だが、ログオンするユーザーは signon.txt のユーザープロファイルに固 定されてしまう。

この機能が強化された SP4i では自動 ログオンに使うアカウントファイルを HTML ごとに指定できるようになって いる。具体的には name 属性 SIGNON の value 値に「自動ログオンするアカウ ントファイル」を設定できる。【ソース4】 この SP4i の自動ログオンの仕組みに

よって、JC/400 で固定されていたユー ザープロファイルを自由に使い分けるこ とができる。【図 16】

たとえば部門や役職によって権限や メニューを自動ログオンするユーザープ ロファイルで制御ができるので、SP4i では利用ユーザーにも柔軟に対応できる ようになる。

6.おわりに

本稿では、新しい Web ブラウザ環境 への対応方法として、JC/400 から SP4i へのマイグレーションのポイントを説明 した。

マイグレーションの内容としては、 JC/400(IE限定)のプログラムがその まま SP4i(別のWebブラウザ)で動か せる内容となっているが、4章で例示し たようにHTML5などの機能やjQuery などのライブラリを組み込めば、今まで 以上に綺麗なデザインで高機能なWeb 画面にカスタマイズしていくことができ る。

また冒頭でも述べたとおり、SP4i は ハイブリッド Web アプリケーションと しての機能を持っているため、単純に Chrome や Safari で動かせるようにな るだけでなく、モバイル向けのアプリ ケーションも開発できる。本稿では SP4i のモバイルアプリケーション開発 までは説明をしていないが、RPG をベー スとしたアプリケーションでカメラや バーコード、マップなどの機能が活用で

● IBM iプログラムの生成	× @ レポート ×
	'请奉張:
IBM iへの配布:	Insert: SP4I/PAGEFILE record :
192.168.0.21	
	ブログラムの起動
オブジェクト・ライブラリー名	IBM I OBJECT LIDIARY JCSMPLIB
JCSMPLIB	戻り値:0
いーフ・ライブラリータ	E
	rapop.
HTMLファイル IBM iブログラム名	ERROR:
SMP010.HTML SMP010	Tue Aug 14 11:57:59 JST 2018
SMP020.HTML SMP020	エラーなし。
「IBM i言語選択	
₩ RPG	
COBOL	エラーがなければOK
Deploy Exit	
配布	
1 BL 20	
55 1	
WEBサーバーへ配布	
WEBサーバーへ配布	
WEBサーバーへ配布	×
WEBサーバーへ配布 ・ レポート	×
WEBサーバーへ配布 ● レポート 谐報:	×
WEBサーバーへ配布 ● レポート	
WEBサーバーへ配布 ● レポート 猫報: Tue Aug 14 12:00:37 JST 2018 ====================================	
WEBサーバーへ配布 ● レポート	× htdocs\ja_JP\smartpad4i\html\JC
WEBサーバーへ配布 ● レポート	× htdocs\ja_JP\smartpad4i\html\JC
WEBサーバーへ配布 ● レポート	Image: Additional and the second s
WEBサーバーへ配布 ● レポート 猫報: Tue Aug 14 12:00:37 JST 2018 ====================================	htdocs\ja_JP\smartpad4i\html\JC htdocs\ja_JP C:\Program Files (x86)\IBM\HTTPs
WEBサーバーへ配布 ● レポート	htdocs\ja_JP\smartpad4i\html\JC \htdocs\ja_JP C:\Program Files (x86)\IBM\HTTPs
WEBサーバーへ配布 ・ レポート	× htdocs\ja_JP\smartpad4i\html\JC \htdocs\ja_JP C\Program Files (x86)\IBM\HTTPs
WEBサーバーへ配布 ● レポート	× htdocs\ja_JP\smartpad4i\html\JC \htdocs\ja_JP C:\Program Files (x86)\IBM\HTTPs ▼
WEBサーバーへ配布 ● レポート	Intdocs\ja_JP\smartpad4i\html\JC Ntdocs\ja_JP C:\Program Files (x86)\IBM\HTTPs
WEBサーバーへ配布 ● レポート	A htdocs\ja_JP\smartpad4i\html\JC \htdocs\ja_JP C:\Program Files (x86)\IBM\HTTPS
WEBサーバーへ配布 ● レポート 諸報: Tue Aug 14 12:00:37 JST 2018 Project C:\Program Files (x86)\IBM\HTTPServerN Send to C:\Program Files (x86)\IBM\HTTPServerN Dont' forget to check that your html file(s) are in C ■ ERROR: Tue Aug 14 12:00:37 JST 2018	Image: With the second seco
WEBサーバーへ配布 ● レポート	Implementation Implemen
WEBサーバーへ配布 シレポート 猪糠: Tue Aug 14 12:00:37 JST 2018 Project C:\Program Files (x86)\IBM\HTTPServerN Send to C:\Program Files (x86)\IBM\HTTPServerN Dont forget to check that your html file(s) are in C Import forget to check that your html file(s) are in C Import forget to check that your html file(s) are in C Import forget to check that your html file(s) are in C Import forget to check that your html file(s) are in C Import forget to check that your html file(s) are in C Import forget to check that your html file(s) are in C Import forget to check that your html file(s) are in C Import forget to check that your html file(s) are in C Import forget to check that your html file(s) are in C	Intdocs\ja_JP\smartpad4i\html\JC htdocs\ja_JP 2:\Program Files (x86)\IBM\HTTPS
WEBサーバーへ配布 シレポート	htdocs\ja_JP\smartpad4i\html\JC \htdocs\ja_JP _:\Program Files (x86)\IBM\HTTPs
WEBサーバーへ配布 シレポート 猪報: Tue Aug 14 12:00:37 JST 2018 Project C:\Program Files (x86)\IBM\HTTPServer\ Send to C:\Program Files (x86)\IBM\HTTPServer\ Dont' forget to check that your html file(s) are in C ・ ERROR: Tue Aug 14 12:00:37 JST 2018 エラーなし。	Image: wide wide wide wide wide wide wide wide
WEBサーバーへ配布 シレポート 猪報: Tue Aug 14 12:00:37 JST 2018 Project C:\Program Files (x86)\IBM\HTTPServerN Send to C:\Program Files (x86)\IBM\HTTPServerN Dont' forget to check that your html file(s) are in C Immediate State	Image: wide wide wide wide wide wide wide wide
WEBサーバーへ配布 ・ レポート 体報: Tue Aug 14 12:00:37 JST 2018 Project C:\Program Files (x86)\IBM\HTTPServerN Send to C:\Program Files (x86)\IBM\HTTPServerN Dont forget to check that your html file(s) are in C ERROR: Tue Aug 14 12:00:37 JST 2018 	A htdocs\ja_JP\smartpad4i\html\LC \htdocs\ja_JP C\Program Files (x86)\IBM\HTTPs
WEBサーバーへ配布 ・ レポート	htdocs\ja_JP\smartpad4i\html/UC htdocs\ja_JP 2:\Program Files (x86)\IBM\HTTP
WEBサーバーへ配布 ● レポート	X htdocs\ja_JP\smartpad4i\html\UC \htdocs\ja_JP C:\Program Files (x86)\UBM\HTTPs
WEBサーバーへ配布 ● レポート	X htdocstja_JPtsmartpad4i\htmlUC htdocstja_JP C:\Program Files (x86)\IBM\HTTPS
WEBサーバーへ配布 ・ レポート	Intdocs\ja_JP\smartpad4i\htmlUC htdocs\ja_JP C\Program Files (x86)\IBM\HTTPS
WEBサーバーへ配布 ・レボート 体報: Tue Aug 14 12:00:37 JST 2018 	Intdocs\ja_JP\smartpad4i\htmlJC Intdocs\ja_JP C\Program Files (x86)\IBM\HTTPS

きるので、RPG で実現できるシステム		
の幅が大きく広がる。【図 17】		
 SP4i を新しいアプリケーション開発の		
 開拓にも役立てていただければ幸いであ る。		
M		

DOO2.00 CHGLIBL LIBL(XXXL	IB JACI400DEV JACI400 QTEMP QGPL)	
JUU3.UU ENDPGM		
P4iのライブラリリスト		
P4iのライブラリリスト		
P4iのライブラリリスト 0001.00 PGM		
P4iのライブラリリスト DOO1.OO PGM DOO2.OO CHGLIBL LIBL(XXXL	IB SP41 QTEMP QGPL)	

図8 変更が必要なAPI名

JC/400	SP4i	備考
JACIINIT	SP4IINIT	初期化
JCSEND	SPSEND	ブラウザ側に送信
JCRECV	SPRECV	ブラウザ側から受信
JACISATR	SP4ISATR	SETATR の機能
JACIPHONE	SP4IPHONE	クライアント端末情報取得機能
JACIIPADR	SP4IIPADR	IP アドレス取得
JACISPFM	SP4ISPFM	CSV ファイル出力
JACIOFCK	SP4IOFCK	DATAQ KEY を取得
JACIOFCF	SP4IOFCF	データベースファイルを送る
JACIOFCO	SP4IOFCO	クライアント PC 上に新しいファイルを作る/開く
JACIOFCW	SP4IOFCW	開いたファイルの書き込み
JACIOFCC	SP4IOFCC	開いたファイルを閉じる
JACIOFCD	SP4IOFCD	クライアント PC のファイルを実行、開く
JACIOFCACK	SP4IOFCACK	OFFICE 機能からの通知を待つ

_

-_ _ _ --

--_

図9 変更が必要なフィールド変数名①

JC/400	SP4i	備考
JCLIBC	SPLIBC	ライブラリ
JCHNDL	SPHNDL	ハンドル
JCLIB	SPLIB	ライブラリ
JCFILE	SPFILE	HTML ファイル名
JCRETN	SPRETN	リターンコード
JCRCDN	SPRCDN	レコード名
JCMULT	SPMULT	複数行のフラグ
JCRCDL	SPRCDL	レコード長
JCCSRF	SPCSRF	カーソルフィールド
JCCSRL	SPCSRL	カーソル行
JCFLGS	SPFLGS	処理フラグ
JCNBRL	SPNBRL	ループ行
JCLOOP	SPLOOP	ループ変数
JCTIMO	SPTIMO	タイムアウト
JCACTN	SPACTN	アクションコード
JCACTL	SPACTL	アクションコード行

図10 変更が必要なフィールド変数名②

JC/400	SP4i	備考
JCSFLS	SPSFLS	サブファイルループ開始行
JCSFLR	SPSFLR	サブファイルループ終了行
JCDUMY	SPDUMY	予備フィールド
JCTABN	SPTABN	タブ用フィールド
JCFILC	SPFILC	HTML ファイル
JCBUFF	SPBUFF	データ送信用
JCBUF1	SPBUF1	データ送信用
JCFLDN	SPFLDN	SETATR HTML の ID を設定
JCORDF	SPORDF	SETATR 対象行を指定
JCORD1	SPORD1	SETATR サブファイル対象開始行
JCATRC	SPATRC	SETATR 属性コード
JCCLAS	SPCLAS	SETATR クラス名
JCELEM	SPELEM	SETATR ラジオボタンの要素指定
JCATRB	SPATRB	SETATR バッファー
JCRETC	SPRETC	SETATR 結果⊐ード
JCL10X	SPL10X	サブファイルレコード開始行
JCL90X	SPL90X	サブファイルレコード終了行

var inp01 = null;	
function initpage(){	
Inpol = document.getElementById(INPOL);	
4iのgetElementById	
var inp01 = null;	
function initpage(){	
inp01 = SP4i.getElementById('INP01');	
3 JC/4000getElementsBvName	
var items = null:	
function initpage(){	
items = document.getElementsByName ('MYNAME');	
}	
P4iのgetElementsByName	
ar items = null;	
tunction initpage(){	
items = 3r4i.gettiementsbyname (WithAlvie),	

回#791340·元注重用·Inte	net Explorer	- U X	
受注問い合わ	世党録	ち メニューに戻る	
◎ 基本情報			
ብጻ	塩トマト×3 胡麻豆腐×2 太陽スイカ×1		
売上日	4993		
SEL	2018/07/20 #428		
× 525H10198			
101			
succes	© 111/0201 0 111/18		
FAX(8.97			
Height (SHI) ()		n (an - set 1) and	
99998991 (millA0)			
INSTERN (SILVI)	2018/08/13		
連絡区分 (お用入力)	1:TEL ¥		
105M	999 佐藤 一郎		
作业生	000030 Q 有級好菜ファクトリー		
潮送会社	`		
	平日は仕事で荷物を受け取ることができません。 キト 今週の日間にはおよるため、一間日ののためたに取りた	~	
受付内容	S.C. 7 MY/CHECK/TG 7 G/CO/ IMEC//TSHPICROET		
		~	
	今週十個日前で配送するように手配する絵を伝えました。		
		<u>^</u>	
96998			
	 メニューに戻る ● 構造書 		
	● メニューに戻る Copyright(C) 2018 MIGARO. Corporation. All ri	ghts reserved. €, 105% ~	
	● メニューに戻る Copyright(C) 2018 MIGARO. Corporation. All n	phts reserved. 氡 105% ▼	
2 SmartPad	● メニューに戻る Copyright(C) 2018 MGARO. Corporation. All n 4iアプリ実行画面	ghts reserved. 考 105% ▼	
2 SmartPad	 ◆ メニューに戻る ◆ 株認書 Copyright(C) 2018 MGC#R0. Corporation. All in 4iアプリ実行画面 	A model = 105% + √ - 0	×
2 SmartPad	 メニューに戻る ● 病認ま Copyright(C) 2018 MGARO. Corporation. All n 4iアプリ実行画面 ■ = □ ■ = □ ■ = □ 	Image: State of the second distribution of the second dis	×
2 SmartPad	● メニューに戻る Copyright(C) 2018 MGARO. Corporation. All n 4iアプリ実行画面 Market All アプリまた画面 Market All アプリまたので も、の うたニーに戻る	> >	×
2 SmartPad	 メニューに戻る ・ 市設ま Copyright(C) 2018 MGARG. Corporation. All in 4iアプリ実行画面 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・		ě.
2 SmartPad	● メニューに戻る Copyright(C) 2018 MGARO. Corporation. All n 4iアプリ実行画面 Main - □ All アプリ実行画面 Main - □ All アプリまたのの All Resolution All Resoluti	Image: State of the state Image: State of the state Image: State of the state Image: State of the state Image: State of the state Image: State of the state Image: State of the state Image: State of the state Image: State Image: State </td <td>×</td>	×
2 SmartPad	・メニューに戻る ・ 防沈さ Copyright(C) 2018 MGGARG. Corporation. All re 4iアプリ実行画面 ・・・・・・・・・・・・・・・・・・・・・・・・・		×
SmartPad SmartPad * C			×
2 SmartPad	・ メニューに戻る ● 病認書 Copyright(C) 2018 MGGARO. Corporation. All re 4iアプリ実行画面 ************************************	Image: second dimension Image: second dimension Image: second dimension Image: second dimension </td <td>×</td>	×
2 SmartPad	・ メニューに戻る ・ 病沈さ Copyright(C) 2018 MGGARO. Corporation. All n 4iアプリ実行画面 discriteConsequent(SPEConset) edeerviceConsequent(SPEConset) edeerviceConset <td>Image: Second of the second of the</td> <td>× ii š</td>	Image: Second of the	× ii š
SmartPad SmartPad	・ ・ ・ ・ ・ ・ ●	> >	×
C SmartPad C C C C C C C C C C C C C C C C C		> > </td <td>×</td>	×
2 SmartPad		Image: State of the state o	× III S
SmartPad SmartPad		Image: Second State Image: Second State Imag	×
SmartPad SmartPad		>http://www.second. >http://www.second. € 0	×
2 SmartPad		Image: State of the served. Image: State	× ×
2 SmartPad	・ メニューに戻る ・ 市法: Copyright(C) 2018 MGGARO. Corporation. All re 4iアプリ実行画面 (************************************	Image: Provider + #128# Image: Provider + #128# Image: Provider + #128# Image: Provider + #128# <td>× ::: 8</td>	× ::: 8
2 SmartPad	Copyright(C) 2018 MGGARO. Corporation. All n Copyright(C) 2018 MGGARO. Corporation. All n AirCrCllass Copyright(C) 2018 MGGARO. Corporation. All n Copyright(C) 2018 MGGARO.	phs reserved. Image: served.	×
2 SmartPad	Copyright(C) 2018 MGCARC. Corporation. Al n Copyright(C) 2018 MGCARC. Corporation. Al n AI TO D ⁰ UEETOBE AI TO D ⁰ UEETOBEETOBEETOBEETO AI TO D ⁰ UEETOBEETOBEETO AI TO D ⁰ UEETOBEETOBEETOBEETOETO AI TO D ⁰ UEETOBEETO AI TO D ⁰ UEETOBEETOBEETOETO AI TO D ⁰ UEETOBEETO AI TO D ⁰ UEETOBEETOBETO AI TO D ⁰ UEETOBEETO AI TO D ⁰ UEETOBEETO AI TO D ⁰ UEETOBEETOBETO AI TO D ⁰ UEETOBEETOBETO AI TO D ⁰ UEETOBETO	Image:	×



		ヘッダコ	項目固定
Migar	o.Technica	l Report	終了
No.	~	· ○男性	: ○女性 ◎全て
入会日	(検索 条件クリア
No.	会員名(漢字)	会員名(カナ)	性別 生年月日 入会日
00000009	寺脇 育二	テラワキ イクジ	男性 1947/01/09 2012/05/22
00000010	高見 浩正	タカミ ヒロマサ	男性 1955/02/15 2010/05/28
00000011	村井 莉央	477 JA	女性 1992/06/09 2010/06/12
00000012	高井 雄太	974 179	男性 1932/04/18 2010/06/12
00000013	福沢 愛梨	87 714 h	女性 1984/12/06 2010/00 4
滑らかに.	スクロールが	「可能 ヒロコ ヨシマサ	スクロールバーが 女性 男性 スクロールバーが 表示される
00000017	吉田 徹	ヨシダ トオル	男性 1949/05/14 2012/05/11
00000018	令责 十陸	27777 VD+	甲性 1078/05/15 2012/05/20
HIG HIG?	±ミガロ. Copyright(C)	MIGARO. Corporation. All rights reser	rved.
1 <html> 2 <head> 3 <scr< td=""> 4 <scr< td=""> 5 <scr< td=""> 6 7 <body> 8 <for< td=""> 9 <i< td=""> 10 11 12 <scrip< td=""> 13</scrip<></i<></for<></body></scr<></scr<></scr<></head></html>	은 아이지 Copyright(C) 언 IPT type="te IPT type="te IPT type="te Ye M method=POS nput type="h RM> Ye Ye T type="text Ye	ext/javascript" src="sp4 ext/javascript" src="sp4 ext/javascript" src="sp4 ext/javascript" src="sp4 ext/javascript" src="sp4 idden" name="SIGNON" va	imsg.js">┵ iparm.js">┵ ilogon.min.js">┵ lue="profile/signonJCSMPLIB.txt">┵ t();┵



Migaro. Technical Report 既刊号バックナンバー

電子版・書籍(紙)媒体で提供中! http://www.migaro.co.jp/contents/support/technical_report/

No.1 2008 年秋

お客様受賞論文

●最優秀賞 直感的に理解できるシステムを目指して一情報の"見える化" の取り組み 石井 裕昭様 / 豊鋼材工業株式会社

●ゴールド賞
 運用部間にサプライズをもたらした Delphi/400
 春木 治様/株式会社ロゴスコーポレーション

●シルバー賞 JACi400 使用による Web アプリケーション開発工数削減 中富 後典様/日本梱包運輸倉庫株式会社

Delphi/400 を利用した Web 受注システム 飯田 豊様/東洋佐々木ガラス株式会社

●優秀賞 Delphi/400 による販売管理システム(FAINS)について 藤田 建作様/株式会社船井総合研究所

技研化成の新基幹システム再構築 藤田健治様/技研化成株式会社

SE 論文

はじめての Delphi/400 プログラミング 畑中 侑/システム事業部 システム 2 課

Delphi/400 と Excel との連携 中嶋 祥子/ RAD 事業部 技術支援課

連携で広がる Delphi/400 活用術 尾崎 浩司/システム事業部 システム 2 課

フォーム継承による効率向上開発手法 吉原 泰介/ RAD 事業部 技術支援課

API を利用した出力待ち行列情報の取得方法 鶴巣 博行/ RAD 事業部 技術支援課

Delphi テクニカルエッセンス Q&A 集 吉原 泰介/ RAD 事業部 技術支援課

JACi400 を使って RPG で Web 画面を制御する方法 松尾 悦郎/システム事業部 システム 2 課

あなたはプラインドタッチができますか? 福井和彦/システム事業部システム1課

No.2 2009 年秋

お客様受賞論文

●最優秀賞
 JACi400 で 既存 Web サービスの内製化を実現
 佐々木 仁志様/株式会社ジャストオートリーシング

●ゴールド賞
 .NET 環境での Delphi/400 の活用
 福田 祐之様 / 林兼コンピューター株式会社

●シルバー賞 5250 で動作する「中古車 在庫照会プログラム」の GUI 化 佐久間 雄様/株式会社ケーユー

●優秀賞
 Delphi による 輸入システム「MISYS」の再構築
 秦 榮禧様 / 株式会社モトックス

Delphi/400 による物流システムの再構築 仲井 学様/西川リビング株式会社

Delphi/400 で開発し 3 台のオフコンを 1 台の IBM i へ統合 島根 英行様/シルフ

SE 論文

JACi400 環境でマッシュアップ! 岩田 真和/ RAD 事業部 技術支援課

Delphi/400 を利用したはじめての Web 開発 福岡 浩行/システム事業部 システム 2 課

Delphi/400 を使用した Web サービスアプリケーション 尾崎 浩司/システム事業部 システム 3 課

Delphi/400 によるネイティブ資産の応用活用 吉原 泰介/ RAD 事業部 技術支援課 顧客サポート

RPG でパフォーマンスを制御 松尾 悦郎/システム事業部 システム 1 課

MKS Integrity を利用したシステム開発 宮坂 優大・田村 洋一郎/システム事業部 システム 1 課

No.3 2010年秋

お客様受賞論文

●最優秀賞 建物のクレーム情報管理システム「アフターサービス DB」 について 大橋 良之様 / 東レ建設株式会社

 ●ゴールド賞
 Delphi/400 で「写真管理ソフト」と「スプールファイル の PDF 化ソフト」を自社開発
 寒河江 幸喜様 / 日線産業株式会社

●シルバー賞 Delphi/400 で鉄鋼受発注業務を統一し 鉄鋼 EDI も実現 柿本 直樹様/合鐡産業株式会社

●優秀賞 Delphi/400 で EIS (Executive Information System)の高速化 小島 栄一様/西川計測株式会社

イントラでの PHP-Delphi-RPG 連携 仲井 学様/西川リビング株式会社

Delphi/400 を使った取引先管理システム 大崎 貴昭様/森定興商株式会社

SE 論文

Delphi/400 ローカルキャッシュ活用術 中嶋 祥子/ RAD 事業部 技術支援課

Delphi/400 帳票開発ノウハウ公開 尾崎 浩司/システム事業部 システム 3 課

Delphi/400 でドラッグ&ドロップを制御 辻林 涼子/システム事業部 システム 2 課

Delphi/400 のモジュールバージョン管理手法 前田 和寛/システム事業部 システム 2 課

Delphi/400 Web からの PDF 出力 福井 和彦・清水 孝将/システム事業部システム 3 課・システム 2 課

Delphi/400 で Flash 動画の実装 吉原 泰介/ RAD 事業部 技術支援課 顧客サポート

No.4 2011 年秋 [創立 20 周年記念号]

お客様受賞論文

●最優秀賞 全社の経費処理業務を効率化した「e総務システム」 鈴木 英明様 / 阪和興業株式会社

●ゴールド賞 「Web 進捗管理システム」でリアルタイム性を実現 堀内 一弘様/エスケーロジ株式会社

●シルバー賞 「**営業奨励金申請書」をたった2日間で開発** 簑島 宏明様/株式会社ケーユーホールディングス

液体輸送における「配車支援システム」の構築 桂 哲様/ライオン流通サービス株式会社

SE 論文

グラフ活用リファレンス 中嶋 祥子/ RAD 事業部 技術支援課

Web サービスを利用して機能 UP ! 福井 和彦・畑中 侑/システム事業部 システム 2 課

OpenOffice 実践活用 吉原 泰介 / RAD 事業部 技術支援課 顧客サポート

VCL for the Web 活用 TIPS 紹介 尾崎 浩司/システム事業部 プロジェクト推進室

JC/400 で JavaScript 活用 清水 孝将/システム事業部 システム 1 課

jQuery 連携で機能拡張 國元 祐二/ RAD 事業部 技術支援課 顧客サポート

No.5 2012 年秋 [創刊 5 周年記念]

お客様受賞論文

【部門1】 ●最優秀賞 JC/400 による取引先との Web-EDI システム構築 久保田 佳裕様 / 極東産機株式会社

●ゴールド賞
 Delphi と Excel を使用した帳票コストの削減
 大久保 治高様/合鐵産業株式会社

もっと見やすく、もっと使いやすい画面を 新谷 直正様/株式会社アダル

【部門 2】 ●優秀賞

Delphi/400 で確認業務の効率化 為国 順子様/ベネトンジャパン株式会社

取引先申請システムでの稟議書作成ワークフロー 大崎 貴昭様 / 森定興商株式会社

Delphi/400 で IBM i のストアードプロシージャを利用 し、SQL 処理を高速化 島根 英行様 / シルフ

SE 論文

InstallAware を使った Delphi/400 運用環境の構築 中嶋 祥子/RAD事業部 技術支援課 顧客サポート

カスタマイズコンポーネント入門 Delphi/400 開発効率 向上 前田 和寛/システム事業部 システム2課

Delphi/400 スマートデバイスアプリケーション開発 吉原 泰介/ RAD 事業部 技術支援課 顧客サポート

DataSnap を使用した3層アプリケーション構築技法 尾崎 浩司/システム事業部 プロジェクト推進室

JC/400 でポップアップウィンドウの制御&活用ノウハウ 清水 孝将・伊地知 聖貴/システム事業部 システム 1 課

【創刊5周年記念】 ミガロ.SE座談会一お客様と共に歩む、お客様への熱い思い

No.6 2013 年秋

お客様受賞論文

【部門 1】 ●最優秀賞 **自社用開発フレームワークの構築** 駒田 純也様/ユサコ株式会社

●ゴールド賞
 Delphi/400 で CTI 開発および関連機能組み込み
 仲井 正人様/株式会社スマイル・ジャパン

●シルバー賞
 IBM WebFacing から JC/400 への移行・リニューアル手法
 八木 秀樹様/極東産機株式会社

Delphi/400 と Delphi を利用した IBM i 資源の有効活用 小山 祐二様 / 澁谷工業株式会社

発注システムを VB から Delphi へ移植しリニューアル 川島 寛様/株式会社タツミヤ

【部門2】 ●優秀賞 5250 画面を使用せずに AS/400 スプールファイルをコ ントロールする 白井 昌哉様/太陽セメント工業株式会社

Delphi/400 を利用した 承認フロー導入による IT 内部統制構築 塚本 圭一様/ライオン流通サービス株式会社

SE 論文

FastReport を使用した帳票作成入門 尾崎浩司/RAD事業部営業推進課

Delphi/400 で開発する 64bit アプリケーション 吉原 泰介/ RAD 事業部 技術支援課 顧客サポート

Web コンポーネントのカスタマイズ入門 佐田 雄一/システム事業部 システム 1 課

 Indy を利用したメール送信機能開発

 辻野 健・前坂 誠二/システム事業部 システム 2 課

Windows テキストファイル操作ノウハウ 小杉 智昭/システム事業部 プロジェクト推進室

JC/400 Web アプリケーションのユーザー管理・メ ニュー管理活用術 吉原 泰介・國元 裕二/ RAD 事業部 技術支援課 顧客サポート

No.7 2014 年秋

お客様受賞論文

【部門1】 ●最優秀賞 Delphi/400 による生産スケジューラの再構築 柿村 実様/東洋佐々木ガラス株式会社

●ゴールド賞
 Delphi/400 および Delphi を利用したオンライン個人別
 メニューの構築
 小山 祐二様/ 澁谷工業株式会社

シルバー賞
 IBM i と Delphi/400 のコラボレーション
 新谷 直正様/株式会社アダル

●シルバー賞 荷札発行システムリプレースについて 仲井 学様/西川リビング株式会社

【部門2】
 ●優秀賞
 Delphi/400 バージョンアップのためのクライアント環境
 構築
 普入 弘様/株式会社エイエステクノロジー

●優秀賞 外出先からメールでリアルタイム在庫を問い合せ 島根 英行様/シルフ

SE 論文 iOS/Android ネイティブアプリケーション入門 吉原 泰介/ RAD 事業部 技術支援課

ファイル加工プログラミングテクニック 小杉 智昭/システム事業部 プロジェクト推進室

FastReport を使用した帳票作成テクニック 前坂 誠二/システム事業部

大量データ処理テクニック 佐田 雄一/システム事業部

スマートデバイス WEB アプリケーション入門 尾崎 浩司/ RAD 事業部 営業推進課 國元 祐二/ RAD 事業部 技術支援課

No.8 2015 年秋

お客様受賞論文

【部門1】 ●最優秀賞 iPod Touch の業務利用開発と検証 石井 裕昭様/豊鋼材工業株式会社

●ゴールド賞 ブランク加工図管理システムの構築 小山 祐二様/澁谷工業株式会社

●シルバー賞
 Delphi/400 でスプールファイル管理(WRKSPLF コマンドの活用)
 三好 誠様/ユサコ株式会社

●シルバー賞 予算管理システムの構築 川島 寛様/株式会社タツミヤ

シルバー賞
 送状データ送信システムの Web 化について
 仲井 学様/西川リビング株式会社

【部門2】 ●優秀賞 繰り返し DB 参照時の ClientDataSet の First 機能に ついて 牛嶋 信之様/株式会社佐賀鉄工所

●優秀賞
 IBM i のカレンダーを基準に他のシステムを稼働
 福島利昭様/株式会社ランドコンピュータ

SE 論文

フレームを利用した開発手法 前坂 誠二/システム事業部 システム2課

Windows タブレット用にカスタムソフトウェアキーボー ドを実装 福井 和彦/システム事業部 プロジェクト推進室

マルチスレッドを使用したレスポンスタイム向上 尾崎 浩司/RAD 事業部 営業・営業推進課

Android アプリケーションの NFC 機能活用 吉原 泰介/ RAD 事業部 技術支援課 顧客サポート

スマートデバイス開発で役立つ画面拡張テクニック 國元 祐二/ RAD 事業部 技術支援課 顧客サポート

No.9 2016年秋

お客様受賞論文

【部門1】
 ●最優秀賞
 IBM i の見える化で実現するアジャイル開発
 吉岡 延泰様/日本調理機株式会社社

●ゴールド賞
 Windows Like 5250 への道のり
 小山 祐二様/澁谷工業株式会社

●シルバー賞
 Delphi プログラム管理ソフトの開発
 牛嶋 信之様/株式会社佐賀鉄工所

【部門 2】 ●優秀賞

Delphi/400 を利用した定型業務の PDF 化 佐藤 岳様/ライオン流通サービス株式会社

●優秀賞
 ちょい足しモバイル
 仲井 正人様/株式会社スマイル・ジャパン

●優秀賞
 AS/400 の受注データを Web で社員に公開
 福島利昭様/株式会社ランドコンピュータ

SE 論文

iOS モバイルアプリ開発のデザイニングテクニック 前坂 誠二/システム事業部 システム 2 課

新データベースエンジン FireDAC を使ってみよう! 福井和彦/システム事業部 プロジェクト推進室

Delphi/400 最新プログラム文法の活用法 尾崎 浩司/ RAD 事業部 営業・営業推進課

FastReport を活用した電子帳票作成テクニック 宮坂 優大/システム事業部 システム 1 課

Beacon 技術による IoT 活用の第一歩 吉原 泰介/ RAD 事業部 技術支援課 顧客サポート

 Web & ハイブリッドアプリ開発で役立つ IBM i & ブラウ

 ザデバッグテクニック

 國元 祐二/ RAD 事業部 技術支援課 顧客サポート

No.10 2017 年秋

Migaro.Technical Report 創刊 10 周年記念

パートナー様からの祝辞 武藤 和博様/日本アイ・ビー・エム株式会社 藤井 等様/エンバカデロ・テクノロジーズ日本法人 Serge Charbit 様/ SystemObjects Corporation 飯田 恭子様/アイマガジン株式会社

お客様からの祝辞・お客様の声

石井 裕昭様/豊鋼材工業株式会社 牛嶋 信之様/株式会社佐賀鉄工所 大崎 貴昭様/森定興商株式会社 川島 寛様/株式会社タツミヤ 久保田 佳裕様/極東産機株式会社 駒田 純也様/ユサコ株式会社 駒田 純也様/ユサコ株式会社 集河江 幸喜様/日綜産業株式会社 集河江 幸喜様/日綜産業株式会社 住々木 仁志様/株式会社ジャストオートリーシング 佐藤 岳様/ライオン流通サービス株式会社 白井 昌哉様/太陽エコブロックス株式会社 仲井 学様/西川リビング株式会社

お客様座談会

石井 裕昭様/豊鋼材工業株式会社 駒田 純也様/ユサコ株式会社 寒河江 幸喜様/日綜産業株式会社 仲井 学様/西川リビング株式会社 上甲 將隆/株式会社ミガロ. 司会 飯田 恭子様/アイマガジン株式会社

	お客様受賞論文
	【部門 1】
	●最優秀賞
	貸金庫と鍵のマッチング業務を Delphi/400 で実現
,	―文字認識データと基幹システムデータを統合
	佐藤 正様/株式会社富士精工本社
	●ゴールド賞
,	Windows タブレット導入による工作部材料受入業務改革 小山 祐二様/澁谷工業株式会社
	【部門 2】
	●優秀賞
	Delphi/400 を利用した各拠点 PING コマンド簡素化 松垣 秀昭様/ライオン流通サービス株式会社
	汎用的な帳票出力画面
	牛嶋 信之様/株式会社佐賀鉄工所
	バーコードリーダー読み取り後 次の入力位置にカーソルを
	自動遷移させる技術
	ーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーー
	IDM: のフプー II.ファノ II. 弁昭烨的た Wob で構筑
	福島 利昭様/株式会社ランドコンピュータ
	SE 論文
	デスクトップアプリケーション開発でも
	役立つ FireMonkey 活用入門
	尾崎 浩司/ RAD 事業部 営業・営業推進課
	Delphi/400 バージョンアップに伴う文字コードの違いと 制御
	┲уш₽ 宮坂 優大/システム事業部 システム1課
	FastReport への効率的な帳票レイアウトコンバート 畑中 侑/システム事業部 システム2課
	IBM i トリガー機能を活かしたセキュリティログ対応
	八木沼 幸一/システム事業部 プロジェクト推進室
	アプリケーションテザリングを利用した PC &モバイルア
	プリケーション連携
	吉原 泰介/ RAD 事業部 技術支援課 顧客サポート
	SmartPad4iの運用で役立つ WEB サーバー機能

МЕМО	



Migaro.Technical Report No.11 2018 年秋 ミガロ.テクニカルレポート

2018年12月1日初版発行

◆発行
 株式会社ミガロ.
 〒 556-0017
 大阪府大阪市浪速区湊町 2-1-57 難波サンケイビル 13F
 TEL: 06(6631)8601 FAX: 06(6631)8603
 http://www.migaro.co.jp/

◆発行人

上甲 將隆

◆編集協力

アイマガジン株式会社

◆デザインフォーマット

近江デザイン事務所

©Migaro.Technical Report2018 本誌コンテンツの無断転載を禁じます 本誌に記載されている会社名、製品名、サービスなどは一般に各社の商標または 登録商標です。本誌では、TM、®マークは明記していません。

