

Migaro.Technical Report Sta.F7=カルレポート

ごあいさつ

CONTEN_IT_xS

Migaro.Te	echnical Award 2010 お客様受賞論文/ミカロ.テクニカルアワード	
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	04
ゴールド営	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	10
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●合鐵産業株式会社	16
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	20
	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	23
	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	27
Migaro.Tec	chnical Report 2010 ミガロ .SE 論文/ミガロ.テクニカルレポート	
初級者向け	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	32
	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	41
中級者向け	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	49
	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	57
上級者向け	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	66
	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	72

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。

さて、2008年10月に"ミガロ.製品をご利用のお客様に、より多くの技術情報の発信を行う"ことを目的に技術論文集『Migaro.Technical Report』を創刊、昨年2009年には第2号を発刊いたしました。お陰さまで、この論文集は多くのお客様に非常に高いご評価をいただいており、たいへん感謝をしております。

そして、今回『Migaro.Technical Report』の第3号を無事に発刊する運びとなりました。 従来と同じく2部構成になっています。

第1部は「Migaro.Technical Award 2010 お客様受賞論文」を掲載し、第2部は「ミガロ.SE 論文」を掲載しています。

「Migaro.Technical Award」とは、日々アプリケーション開発・保守に携わるエンジニア の方々の努力と創意工夫の効果を顕彰することを目的とし、「Delphi/400」「JC/400」「MKS Integrity」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。今回の論文は、IBM i とOracle などの他 DB サー バーとの連携を生かしたシステムや、IBM i と連携した画像管理、スプールのPDF 化といっ た、Delphi/400の特長を見事に業務課題の解決に結びつけられた事例が揃っております。

また「ミガロ.SE 論文」は、弊社 SE による技術論文を掲載しております。今回はローカ ルキャッシュの活用からWeb でのPDF 出力など、より実践的な技術情報を揃えており、初 級 / 中級 / 上級向けに分けてご紹介しております。

本レポートが少しでも皆様の開発・保守のお役に立てれば幸いです。

最後に『Migaro.Technical Report』第3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 御礼申し上げます。

2010年秋

株式会社ミガロ. 代表取締役社長 上甲 將隆

MIGARO.

REPORT

TECHNICAL

1



Migaro. Technical Report 2010

お客様受賞論文/ミカロ.テクニカルアワート

Migaro.Technical Award 2010 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 技術部技術室 課長



東レ建設株式会社 http://www.toray-tcc.co.jp/

ファッションから宇宙開発まで、高 いテクノロジーで暮らしを支える東 レグループの中の建設会社である。 総合建設事業(ゼネコン)と総合不 動産開発事業(デベロッパー)を併 せ持つ数少ない企業でもある。

開発背景

従来の建物のクレーム対応は、受付部 署でシステムへの登録を行ってはいた が、この既存システムの対象は重大なク レームだけですべてのクレーム管理はで きていなかった。

そのシステムについては利用すること 自体の負担が大きく、日常の業務とは完 全に独立しているうえ、建物の施工デー タなど詳細を調べないと入力ができず、 利用者からは時間と労力がかかり非常に 使いづらいという声があがり、徐々に活 用頻度も減っていた。

蓄積されたデータは手集計を行い、分 析後、結果を現場にフィードバックをし ていたが、不定期かつ部分的なもので あったので、的確にすべてのクレームに 対応できていないなどの問題点があった。

その他にも下記(①~⑤)のような問 題点があったため、「アフターサービス DB」という新たにシステム名を付けてそ れらの問題解決や業務改善を検討した。

- ①お客様のクレーム内容と、対応の経 過が共有化されていないため担当者 にしか分からない。
- ②重大なクレームが発生した場合、関係者による対策会議を行う必要があるが、その通知や招集がリアルタイムで行えていない。
- ③日常業務の忙しさの中で、クレーム をまとめる管理資料の作成が後回し となっている。
- ④クレームに対する会社の対応が、一 部の関係者にしか伝わっていない ケースがあった。
- ⑤半年ごとに作成する報告書は、クレー ムの集計と分析をすべて手作業で 行っていたため、作成の負担が大きい。

アフターサービス DB の主な特徴

 Delphi/400を使用して「アフターサー ビス DB」としてシステム化したこと により、クレーム発生時の受付から、 担当者が行った対応履歴やクレーム 箇所の写真を貼り付けることが可能 となり、1つのクレームを病院のカル テに例えた「建物カルテ」という概 念で管理が可能となった。

「建物カルテ」では1件のクレームご とに作成され、クレームの対応の履歴 や担当者名など内容の把握に必要な項 目が網羅されていて、必要なデータを 既存システムから引っ張ってくるシス テムである。

②担当者の日常業務に直結したシステム であるため、業務効率を大幅に向上さ せ利用者の負担を軽減させる自動化 が使用頻度を上げることとなった。

③クレーム担当者がお客様ヘクレーム対応するのに必要な建物・施工情報や類似クレーム内容を簡単に入手参照ができる。

④重大なクレームについては、クレーム担当者の上長である各地域の担当事業部長がリアルタイムに内容を確



認でき、社長などへ報告したり重大 クレーム対策会議を招集指示するこ とができる。

- ⑤必要な書類や報告書などを自動作成 させることで、クレーム担当者の業 務効率向上ができる。
- ⑥クレーム内容を、施工した関係者(設計・現場所長や担当者など)に対してメールで配信してクレーム情報を共有化できる。
- ⑦クレームの再発防止策として必要な 社内書類を自動作成し、クレーム防 止策として活用することができる。
- ⑧「建物カルテ」に添付する写真や資料 のデータ容量が大きくなると、作業レ スポンスが極端に悪化するため、各 地域に専用サーバーを設置して写真 や資料データを保管させてレスポン スを大きく改善した。また、写真デー タは自動縮小をして保管している。

Delphi/400 採用の 理由

以下に、Delphi/400 採用の理由を述 べる。【図 1】【図 2】

①現状では独立したシステムであるため、それぞれのデータを照会するために専用端末(パソコン)が必要であったが、Delphi/400は異なるプラットフォームとの連携性・拡張性に優れている。

②末端の使用ユーザーにとって使いやす いビジュアル的な画面開発が可能。 既存の経理管理 DB は、AS/400 を使 用しておりビジュアル的にわかりに くい等の問題があったが、Delphi/ 400 を使用することで解決ができる。

③Webシステムである「文書管理システム」へ「アフターサービス DB」で作成した「建物カルテ」の内、クレーム対応を完了したものだけ自動登録可能。

システム導入前

お客様から受け付けた重大クレーム は、既設システムへの登録を行い、その 後上長へワークフローで報告を行ってい た。報告のあった重大クレームの内、必 要に応じて関係部署でクレーム対策会議 を行い、結果をフィードバックすると共 に、今後の再発防止策に活用していた。

しかし、システム登録時、必要な情報 をマスタから検索することができず、必 要情報を紙のファイルから転記する必要 があった。【図 3】【図 4-1】【図 4-2】

システム導入後

「アフターサービス DB」は、部分的 なクレーム管理や手作業がなくなり全自 動化することができるシステムである。 すべてのクレームを管理するのですべて の情報を共有できるようになり、過去情 報の検索も容易に行えるようになった。

クレーム発生時の受付から、担当者が 行った対応履歴、またクレーム箇所の写 真を貼り付けることが可能になり、1つ のクレームを「建物カルテ」という画面 1つで管理するようになった。その結果、 建物カルテ1つに入力するだけですべて の情報を共有できるようになり、また、 操作性の高い入力画面で登録処理をス ピードアップすることができた。【図5】 【図6】【図7】

また、重大なクレームの場合は、関 係部署や上司に「クレーム対策会議」を 開催するワークフローを自動で召集依頼 したり、報告内容の承認作業もスピード アップできるようになった。

登録データは、いくつもの関係書類を 自動作成し、クレーム防止策としてメー ル配信したり再発防止策として活用でき るようにもなった。

システム化の効果と 今後について

「アフターサービス DB」システムの 導入で、すべてにおいてスピーディな対 応が可能になった。「建物カルテ」への 登録時、簡単にクレーム箇所の写真等も 貼り付けることができ、今まで管理でき ていなかった全クレームの対応履歴や、 クレームに関するその他すべての情報を 一元管理できるようになった。

「アフターサービス DB」は、クレー ム情報を現場にフィードバックし、起こ りやすいクレームを事前に絶滅させる 「クレームのゼロ化」活動が行えるよう になり、クレームの再発防止と、高品質 サービスを実行することによって、お客 様の顧客満足度を向上させることができ るようになった。

現在、導入して約2年6ヶ月が経過し ており、1,100件を超えるデータベース を蓄積すると共に、システム使用者から の意見を反映して約66項目の改良を加 えた。

現在では、社内でも活発に「アフター サービス DB」の改善に取り組んでいる。

業務に直結したシステムを開発でき て、クレームに携わっている担当者や部 署から評判が良く感謝している。

今後も、「アフターサービス DB」に よる業務効率の向上とさらなるシステム の充実を図って、お客様にクレームのな い建物を提供していきたい。

Μ

東レ建設株式会社(詳細)

総合建設事業(ゼネコン)と、総合不動産開発事 業(ディベロッパー)の2つの顔を兼ね備えた企業。 建物の企画・設計・施工からアフターフォローまで を行うゼネコンの機能と、マーケティング活動から 用地開発・企画・供給までを手がけるディベロッパー としての機能を持つ。建設のすべてのプロセスを1 社で担える企業であることが、東レ建設の最大の特 色。また「東レ株式会社」の100%子会社として、 東レグループの建設・不動産事業を支えている。 図3 システムフロー(現状のクレーム対応)



図4-1 クレーム処理状況報告書

90.	ークスペース		動請負案件クレーム処理;	×			
り閉	じる 日作成						
請	「魚」「「魚」	レーム処理状況報告書		申請番号 申請日 申請者	;	TCC	
氏	氏名	Macan	所属	東京不動産事業部 建	は サG(エ 役職		
		(R98998)		争) (DJOJ)	電話		
承	包情報						
履	虚情報						
100000000000000000000000000000000000000	A 100 107						
w	全腐歴						
~	全腐歴	and the later of the second data and the second data and					
	「 全腹歴 氏名 / 所	属 / 役職 / 日付 / 状》	こく コメント				
	「 室廠歴 氏名 / 所	属 / 役職 / 日付 / 状》	2 / コメント	711.加油+4:口本	招生章		
	主旗歴 氏名 / 所 請者用記 / 二	属 / 役職 / 日付 / 状》 、 欄	2 / コメント (クレーム処理状	7レーム <u>処理状況執</u> 況報告書の内容を以って担保工程	<u>現任書</u> 事施工申請書に替え	5)	
● 単 部作	王腐歴 氏名 / 所 請者用記 『請者用記 『 書整理番号 成日:200	<u>属 / 役職 / 日付 / 扶</u> 济 桶 5/10/30	2 / コメント (クレーム処理状	フレーム <u>処理状況</u> 況報告書の内容を以って担保工	祝 <u>告書</u> 事施工申請書に替え	5)	
● 部作 東	王腐歴 氏名 / 所 諸者用記 常成日:200 京建築サー	<u>属 / 役職 / 日付 / 状況</u> 備 き: J-061030 6/10/30 ビス課	2 / コメント (クレーム処理状	フレーム <u>処理状況</u> 況報告書の内容を以って担保工	<u>祝告書</u> 事施工申請書に替え	5)	
■ ■ 部作 東 123	主席歴 氏名 / 所 請者用記 常成日:200 京建築サー 受問客名:	<u>属 / 役職 / 日付 / 状況</u> 備 *: J-061030 8/10/30 ビス課 目 : 2006/7/3 (受付者	R / コメント (クレーム処理状 音): (新知)	フレーム <u>処理状況</u> 況報告書の内容を以って担保工	<u>祝<i>告:書</i></u> 事施工申請書に替え	5)	
■ 単 部作 東: 123	主席歴 氏名 / 所 請者 者 用 記 えの 京 建 受付 案 で の 京 建 受 前 客 で の 定 受 騎 客 る () () () () () () () () () (<u>属 / 役職 / 日付 / 状況</u> 備 *: J-061030 6/10/30 ビス課 目 : 2006/7/3 (受付者 供 業業のの方法業業)	R / コメント (クレーム処理状 音): (新知)	クレーム <u>処理状況</u> 4 況報告書の内容を以って担保工 (完工後4	<u>祝<i>告:書</i></u> 事施工申請書に替え ² 名称): (()線線線	5)	
■ ■ 部作 東 123	主席歴 氏名 / 所 諸者 用 記 示 成 定 受付年月 : 200 京 建築 (工事等) : 200 京 建築 (工事等)	<u>属 / 役職 / 日付 / 状</u> 済 補 ま:J-061030 6/10/30 ビス課 目:2006/7/3 (受付者 供 課番号):[] 電番号):[]	2 / コメント (クレーム処理状 音) : (新知)	クレーム <u>処理状況</u> 況報告書の内容を以って担保工 (完工後4)	<u>祝告書</u> 事施工申請書に替え 名称): (0 線機構	5)	
▶ ■ 部作 東 123	王 麻 歴 氏名 / 所 請 者 用 記 フ	属 / 役職 / 日付 / 状況 (欄) 5: J-061030 6/10/30 ビス課 日: 2006/7/3 (受付者 (報) 理番号): [] 年月日: 1996/9/30 1: 1. 構造 S道 2. 階数 地上 2階 均 3. 住戸数 1戸	2 / コメント (クレーム処理状 音) : (新会) 也下 一階	フレーム <u>処理状況</u> 況報告書の内容を以って担保工 (完工後 都屋番号 -	<u>祝告書</u> 事施工申請書に替え 名称): (() ()	5)	

図4-2 クレーム処理状況報告(承認状況)

(F) 編集(D. HX 日 一 /	() 作成(C) ア	(A) = 10105 N	しての しつ(H)						
Alex	ታ杂ۍ	친슈: 이 이 :			BISE		• = (A 111	/ P # 2		
	アドレス	• • • •					and your states			
フークスペーン	スシクレーム	処理状況報告 ×	10請負案件クレー	ム処理×						
, <i>b</i> n 18	し、表示の	更新 ② 全て展開	乳全て省略 4	「閉じる						
レーム処理状況	状部	8署、部署整理番、	所属名	申請者 ◇	申請日 👌	申請番号 ☆	承認日。	現在の承認者	建物名称 👌	完成後建物
ークフロー申	*		東京建事部 設部(建築	1 200 200						
申請書入力			サ <i>ー</i> ビス)(ご 事)	<u>E</u>						
承認状況 認待ち一覧	*		東京建事部 設部(建築 サービス)(MARKED
状態順	*		事) 東京建事部 設部(建築							
旧文書			サニビス)(1 事)	C						
事業部別	*		東京建事部 設部(建築 サービス)(e ling and						得在神秘
8告宛先登 事業部別			事) 大阪不動産 業部 建サG						18013 (1818)	Maret
scel出力	*		大阪) 品質管理室 (大阪)(管							
社 大阪 滋 名古屋	*		理) 大阪不動産 業部 建サG	≢ #*#3#₩						
島東京		051128	大阪) 大阪不動産 業部 建サロ							
		061 023	大阪) 大阪不動産 業部 建サG							
		061 030	大阪) 東京不動産 業部 建サG							
			<u>工事</u>) 大阪不動産 業部 建サG							16 72 0
		4/24	大阪) 大阪不動産	事 法保持领援					THE REAL PROPERTY OF	
	*	20080205	 業部 建サG 関西建設事 部(土木)(3) 							2 5028 96
			事)							

図5 システムフロー(アフターサービスDBシステムについて)



裔[CL1010] 建物かテ 入力	_ _ ×
加示No 75~9° 重大判定状况	新規 🛆
加行担当者 ****** 大橋良之 建物CD 000-0001 大橋邸	
所長名 竣工年月日 2008/02/28 0年2ヶ月 住所 滋賀県大津市	
練名 部屋番号 場所 台所排水 区分((事業) 不動産 💌 地区 京滋
氏名 大橋良之 連絡先1 自宅 ▼ 123-4567 2 携帯 ▼ 123-4567	費用区分 ○ AS ○ DH ○ TG ○ RF ////
Xモ(カルテ)	クレーム ● 軽微 ● 重大
受付	料 3/21
表示対象 対応日 対応担当者 対応 〇 2008/04/15 大橋良之 調査に伺った。入居者は怒っている。	内容
○ 2008/04/16 大 橋 良 之 ご主人に経過説明し、納得された。後日取	り替え予定。
対応日 対応担当者 日 建物加近へ多	表示対象とする(6件まで)
対応内容	※対応内容はから発生内容書に
	反映されません。 建物加炉には最大6件まで
	反映されます。
☆:行追加(1) 約行編集(E) ☆:行削除(D)	
	A STREET AN UNCONSTR
 ロ 建物加好を開く(0) 四 重大申請(0) 合 完了(1) 合 文書(3)(9) アフターサービスDB2 	注即除(Z) ■保存(S) 直閉じる(C)
回 建物加げを開く(0) (周重大車請(2) 自完了(1) 自文吉(3)(3) アフターサービスDB2 ALFNO. 「」、「大橋良之」 建物00 [000-0001]「大橋邸 が長名 録工年月日 [2008/02/28]0年2ヶ月 住所 [滋賀県大津市 東名 部屋番号 場所 台所排水 区分(事業) 氏名 大橋良之 連絡先1 自宅 [123-4567] 2 携帯 [123-4567] 費用	注印除(Z) ■保存(S) ①閉じる(C) ●●●●● ●●●●● ●●●●● ●●●●● ●●●●● ●●●● ●●●● ●●●● ●●●● ●●●● ●●●● ●●●● ●●● ●●● ●● ●● ●● ●● ●● ●● ●
□ 建物加好を開く(0) ③ 重大車請(2) ③ 完了(1) ③ 文書(3)(9) アフターサービスDB2	(二回来) (前時(乙)) (前日にる(C)) (前日にる(C)) (前日にる(C)) (前日にる(C)) 「「前規 (四)) 「「前規 (四)) 「「前規 (四)) 「「前速 (回)) 「「「前速 (回)) 「「「前速 (回)) 「「前速 (回)) 「「前速 (回)) 「「「前速 (回)) 「「「「「「「「「「「「「「「」」(□)) 「「「「「「「」」(□)) 「「「「「「「」」(□)) 「「「「」」(□)) 「「「」」(□)) 「「「」」(□)) 「「」(□)) 「」(□)) 「「」(□)) 「」(□)) 「」(□)) <p< td=""></p<>
回 建物加好を開く(0) (当重大車請(2) (当完了(1) (当文吉13(9)) アフターサービスDB2 [CLIO10] 建物カルテ入力 加行し、「一」 びージ 重大判定状況 加行担当者 ****** …」大橋良之 建物00 000-0001 …」大橋邸 新長名	(2) 日保存(S) ①閉じる(C) FT期除(Z) 日保存(S) ①閉じる(C) FT規定 ●
回 建物加尔 宏開 < (0) (3 重大申請 (2) (3 完了 (3) (3 文書的 (8) アフターサービスDB2	(二) 「日除(Z)」「日保存(S)」「日にる(C)」 ●
回 建物加炉を開く(0) 自 重大申請(2) 自 京子13(8) アフターサービスDB2 [CL1010] 建物加炉 入力 加水10。 - ファウ* 重大判定状況 - 加水10。 - ファウ* 重大判定状況 - - 加水10。 - ファウ* 重大判定状況 加水10。 - - ・ 加水10。 - ファ・ ・ 大橋取 加水10。 - - ・ ・ 加水10。 - ・ ・ ・ 加水110 - ・ ・ ・ ・ 受付 対応 最終調査・処置 ・ ・ ・ ・ 受付 対応 最終調査・処置 ・ ・ ・ ・ ・ ・ ・ ・ <td>注前除(Z) ■保存(S) ①閉じる(C) 「「前焼」(Z) 「「「「「「」」」」 「「「」」」」 「「」」」 「「」」」 「「」」」 「「」」」 「」」 「」」」 「」」 「」」 「」」」 「」 「」」 「」」 「」」 「」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」 「」 「」」 「」」 「」」 「」 「」 「」」 「」」 「」 「」 「」」 「」 」」 」」 」」 」 」」 」」 」」 」」 」」 」」 」」 」」 」」 」 」」 」」 」」<</td>	注前除(Z) ■保存(S) ①閉じる(C) 「「前焼」(Z) 「「「「「「」」」」 「「「」」」」 「「」」」 「「」」」 「「」」」 「「」」」 「」」 「」」」 「」」 「」」 「」」」 「」 「」」 「」」 「」」 「」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」 「」 「」」 「」」 「」」 「」 「」 「」」 「」」 「」 「」 「」」 「」 」」 」」 」」 」 」」 」」 」」 」」 」」 」」 」」 」」 」」 」 」」 」」 」」<
回 建物加好を開く(0) 息重大車請(2) 自文書(3)(9) アフターサービスDB2 [CL1010] 建物加テ 入力 加7103 五ヶ沙* 重大判定状況 加7103 万ヶ沙* 重大判定状況 加7103 ブラジ* 重大判定状況 加7103 「大橋印 加7103 「本****** … 「大橋印 加7103 「二 フラジ* 重大判定状況 加7103 「二 フラジ* 重大判定状況 加7103 「二 フラジ* 重大判定状況 加7103 「二 フラジ* 重大判定状況 加7104 「三 フラジ* 重大判定状況 加7105 「三 ジェーク・ 「大橋印 近日 2008/02/28 0年2ヶ月 住所 原名 部屋番号 場所 「「「「「」」」」」「大橋印 原名 「日を<」	(二) 日保存(S) ①閉じる(C) ● <
回 建物加水子 在内 自 安吉19(8) アフターサービスDB2 Alight	(二回文) 所規 (二回文) 所規 (二回文) 所規 (金) (金)
	(二回文) 所規 (二回文) 所規 (①) (○) 不動産 ↓ 地区 京滋 用区分 ○ AS ○ DH ○ TG ○ RF _ 997 クレーム ○ 軽微 ○ 重大
四建物肪所を開く(0) (3重大申請(2) (3克了(2) (日文書33(2) アフターサービスDB2 [CLIOIO] 建物カルテ 入力 加わし、 ひっか、 重大判定状況 加力目当者 ******	■ 開除(Z) 具保存(S) 直閉じる(C) ●
回 建物加炉を開く(0) (日本大申请(0) (日本下市) (日本下市) アフターサービスDB2 [CL1010] 建物加炉 入力 加水10:5 [万・ジ [*]) 加水10:5 [万・ジ [*]) [大橋印 加水10:5 [万・ジ [*]) [万・(1) [万・(1) [新屋 [新屋 [新屋 [新屋 [日を、][123-4567 [日本、][123-4567 [日本、][123-4567 [日本]	前時(2) ■保存(S) ①開じる(C) 「前席(2) ■保存(S) ①開じる(C) 「新規 ① ○ 不動産 」 地区 京滋 用区分 ○ AS ○ DH ○ TG ○ RF 297 り~4 ●軽微 ○重大 わト
回建物加炉を閉く(0) (日東大申請化) (日京了作) (日文吉33(9) アフターサービスDB2 [CL1010] 建物加ルテ 入力 加水り。	Implication Implication <tr< td=""></tr<>

Migaro.Technical Award 2010 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 日綜産業株式会社 電算室 室長



日綜産業株式会社 http://www.nisso-sangyo.co.jp/

建設用仮設機材の開発、設計、製造、 販売、レンタルの会社。「建設労働 災害の撲滅」を目的に、"足場"を 中心とする仮設機材のメーカーとし て創業。仮設業界のリーディングカ ンパニーとして、さまざまな新しい 取り組みを行っている。

はじめに

従来の 5250 画面のアプリケーション では画面や機能に制限があり、不便な面 が多い。

そこで Delphi/400 を導入し、まず費 用対効果の大きな 2 つのシステム「写真 管理ソフト」と「スプールファイルの PDF 化ソフト」を自社開発した。



開発経緯

仮設機材などの商品が返却された際、 AS/400 に検収データを入力して、修理 が必要な商品や全損(修理不能)の商品 は、破損部分を撮影して検収写真として 添付、お客様に対して請求を行っている。 この業務を改善するために「写真管理 ソフト 検収写真くん」を開発した。【図1】

従来の問題点

検収写真は多い時で1日300枚以上 と大量なこともあり、作業が煩雑で、以 下のようなさまざまな問題点が発生して いた。

- ●機材センターから営業事務担当者への写真データ受け渡し方法は、メールでの送受信であった。そのため、メールサーバー容量の圧迫・ネットワークトラフィックが増加。
- ●写真データの保存や添付作業が大変。 さらに、保存場所が適切でないこと が多く、後から必要な写真を探し出 すことが困難。
- ●印刷は写真管理ソフトが必要な上に、 手作業で写真を見ながら現場ごとに仕 訳し、A4 サイズの用紙に写真を4 つ ずつレイアウトして印刷などの手間が かかる。
- ●検収写真と検収データ(AS/400)は リンクされておらず、突きあわせ作業 はすべて目視であった。また、写真の

信憑性に対する裏付けがシステム化さ れておらず、回収時に不利になること もある。

システムの追加 / 変更点

タスクの流れとシステムの追加 / 変更 点は、表の通りである。

営業店において、Delphi/400のアプ リケーションを使用し、AS/400に保管 している検収データと検収写真を自動で 印刷/保管できるようにした。【表1】

創意工夫した点

このアプリケーション作成にあたり、 以下のような点を工夫した。

● JPEG と AS/400 の連携

AS/400 とマッチングできる修理情報 のキーを、JPEG (画像データ)の EXIF 情報 (付加情報) に持たせた。

この JPEG の EXIF 情報を Delphi/ 400 で読み込み、AS/400 上のデータと マッチングさせて、ファイルサーバー上 の検収写真(JPEG)と AS/400 の検収

表1	流れ図
----	-----

番号	タスク	追加・変更点	担当
1	写真を撮影(ファイル名は返却番号+連番)	今までと同じ	機材センター
2	修理・全損情報を入力(5250 より)	今までと同じ	機材センター
3	日次処理で修理・全損の明細に、写真用番号を自動採番	RPG バッチ処理	夜間バッチ
4	自動採番された番号を、JPEG の EXIF 情報に書きこむ	新たな手作業	機材センター
5	上記の写真を、営業店ごとのファイルサーバー上に保存	メールの添付を廃止	機材センター
6	「検収写真くん」で写真データと AS/400 のデータをマッチング。 写真の印刷・保存を自動化	Delphi/400	営業店

図1



11

●フォルダの自動作成

保存フォルダのカテゴリ(営業担当者・ 得意先・現場・返却日等)や階層を、任 意の組み合わせで自動作成できるように した。【図 2】

● INI ファイルの利用

営業店や担当者により使用方法や保存 方法が異なるので、INIファイルにより 対応した。

- ・保存フォルダのカテゴリや階層
- ・4 枚印刷 /8 枚印刷
- ・保存機能の可 / 不可
- ・過去のフォルダを記憶 など

●写真管理にサムネイル

写真管理ソフトのような、サムネイル を見ながらの作業を実現した。また、 Shift キーやポップアップメニューも採 用した。【図 3】

●「検収写真一覧」の出力

VB-Report を使用し、検収写真の印 刷(自動レイアウト)を実現した。【図4】

- ・ A4 用紙に、4枚/8枚の写真を自動リ サイズしながらレイアウト形成。
- ・ 伝票番号が変わると、自動的にページ 替え。
- ・商品名と修理内容を自動印字。

問題点 / 解決方法

リリースにあたり発生した問題とし て、同じネットワーク環境の同一スペッ クパソコンにもかかわらず、パソコンに よってはパフォーマンスが悪く、数秒で 終わる処理が数分かかってしまった。 JPEG データの EXIF 情報の取得がボ トルネックになっていた。

原因を調査したところ、Delphi/400 ではなく、クライアント PC の OS 環境 で左右されていることが分かった。パ フォーマンスの悪いクライアント PC は、すべて Windows XP の SP2 だっ たため、SP3 をインストールすること で解決することができた。

成果

従来の問題点の多くを解消し、予算を かけずに自前で構築できたことが大きい。 ユーザーのさまざまな要望に対して、 Delphi/400 による開発で満足のいく結 果が出すことができた。

スプールファイルの PDF 化ソフト「スプールくん」

開発経緯

従来から、印刷コスト削減のため、印 刷物を減らしたい、スプール帳票をメー ルに添付したい、チェックだけなら印刷 せずに画面上でチェックしたいなど、さ まざまな課題を抱えていた。また、5577 系プリンターからの脱却も大きな課題と なっていた。

AS/400上のスプールファイルを PDF 化するツールを検討したが、各種のツー ルはあるが高価だったり、専用サーバー が必要だったりして導入には至らなかっ た。今までは、AS/400の帳票を PDF 化するには、一度印刷してからスキャ ナーで読み込んで作成していた。

Delphi/400 での開発に挑戦

Delphi/400を使用して、スプールファ イルの PDF 化する機能のプログラム開 発に挑戦してみようと考えた。

ミガロ.のテクニカルサポートを活用 して、Delphi/400 でプログラミングし たところ、意外に簡単に構築することが できた。

実装したプログラム画面と出力結果 は、図の通りである。【図5】【図6】 また、プログラムの主な処理内容は、

以下のような流れとなっている。

- ●ListSpool400 → DataSource → DBGr id を使い、OUTQ(出力待ち行列) の中身をプログラム画面上に表示。
- ●該当スプールファイルをダブルクリッ クすると、Spool400の各種プロパティ を セ ッ ト し、Active 化。1 行 ず つ Memo コンポーネントに貼り付け。
- ●Memoコンポーネントに貼り付ける ことにより、スプールファイルの内 容をプログラム画面上で表示。
- ●「PDF 作成ボタン」を押すと、該当 Spool400 のページ範囲を VB-Report にセット。VB-Report の出力先プリン ターを、PDF 出力ツール (PrimoPDF や Adobe PDF 等) に設定し、ファイ ル名を現在日時で出力。

創意工夫した点

開発にあたっては、以下のような点を 工夫している。

- ◆VB-Reportで使うExcelフォーマットは 2 種類で対応 桁数 198 用と 132 用(固定幅フォント)
- ●クライアントに依存する設定値は
 INIファイルに持たせる
 ・OUTQ(出力待ち行列)
 ・PDF 作成ツール
- ファンクションキーの多用
 ページ送りやフォントのサイズ変更
 を、ファンクションキーに割り当てた。

問題点

APW には対応していない。APW を 使用している帳票は、個別にデザインす る必要がある。

成果

スタートしたばかりだが、ユーザーか らは好評である。実際に、印刷コストも 少しずつだが減少している。なお、作成 した PDF は「テキスト検索」が可能なの で、保存しておけば検索が柔軟にできる。

最後に

予算があれば、市販の BI ツールや電 子帳票ツールを導入できるが、Delphi/ 400 を使用して開発すれば、自社に特化 したこのようなツールを構築可能だ。

今後は、GUI 化によってユーザービ リティを向上させ、AS/400 上に蓄積さ れているデータを、売上戦略に貢献でき るデータへと変換し、ユーザーに提供で きるようシステムを構築していきたい。

Μ





図5

1 h that a m	KORTLIB /RORTO	, nonny	a	Bac/rr)		- 756	(m. sa)		_	
7-41		at. 27	Sta / H	/ 取利(F3)		C 0H	成 12			
DINT	Seb3618	H 2-2	รถสวาว	ADEADY	10/09/26	17.55.20	nee	ragemiuch		
RINT	SSR1630	1 #51	,)	READY	10/08/28	18.54.94	000	132		
RINT	SSR1630	1 *STI)	READY	10/08/26	18:55:17	000	132		
RINT	SSR2610	2 *STI)	*READY	10/08/26	18:56:07	066	198		
RINT	SSR2610	3 *STI)	*READY	10/08/26	18:56:12	066	198		
R0010		46 *ST)		10/08/29	10:04:27	086	182		
2VDS V5R4M	0 060210	IBM RPG/400		0	TEMP/MTR0010	10/08	/29 10:04:26	<u>^*~)</u> ″	1	
レバイラー		IBM RPG/400								
マンド・オ	プション:									
プログラム	アイル	QTEMP/MTR00 NISS/ORPGSR	10 C							
강드중: 김	ンバー	*PGM *SOURCE	#XREF	RGEN	#NODUMP	#MOSECL.VL	#NOSRCORG	#MOLISTORG		
生成オラシ	1	#NOLIST	*NOXREF	#NOATR	*NODUMP	*NOOPTIMEZE				
タイラ変換	ストのチトケー・・・・・ オブション ・・・・・	*NONE #NONE								
ショーに開閉		#HEX # LODDUN								
SAA 255	56°	*NOFLAG								
生成重大度	1	9 al TRI /06V6P	рт							
ブログラム	の置き換え ・・・・・・・	*YES								
ターゲット	· UU-Z	*CURRENT								
権限		*LIBCRTAUT								
국부スト.	h1-7	*SRCMBRTXT								
千蘭テキス	- gggg : : : : : : : :	HNONE								
スナップ	황건코 • • • • • • • • •	HNONE								
10進データ	エラーの無視	*00								
ヌル値可能		490								
調整のソース	・ ブログラム:									
メンバー		MTR0010 ORPGSRC								
ライブラリ		NISS								
策務変更 記述 		- U5/09/02 1 見緒入力(0:32:19 キー・基本)							
manas • •		2000-0-2-0								
- 82			17-		PDE4L 71-			_		•
F8) 前(FI	(0) 1 / 46)大(F11)	緣(F12)	(F1) 大(F	2) 画面大		● 全て ● 指定	46 3	 ①、検索 	?{}揀作脱剧	前閉じる(0)
					1 <u>200</u>	1876 1				
100000 100										
100829 100	isusipar - Adobe Reader E) 表示(V) 文書(D) ツー	ルロ ウィンドウ(し ヘルプ(日							×
	(16) · · · · · · · · · · · · · · · · · · ·			CON. BREAMOR						

図6

5722WDS V5R4WO 060210	IEM RPG/400	07EMP/MTR0010 10/08/29 10:04:26 **-9* 1	
コンジャッチ・オブション: フログラム ソース・ファイル ソース・ファイル リース、メンバー ソース、リスト(PF) 生成オプション ファスを除すプション 支援目の 装置目の 34数、フラグラけ 生成要大皮ペル 和明ファイル フログラムの置き換え	IDER PREVIOUS IDER PREVIOUS NISS/OPPOSIC +CDM +CDM	-NCOLMP -NOISECL.VL -NOISECOBG -NOLSTOBG NCOLMP -NOIGPTIMIZE	
ターゲット・リリース ユーザー・ブロファイル 権利 デキスト・レース 中国サデフスト・レース 中国サデフスト・シンプ スード・リスト 10 道テータ・エラーの無視 スル値可能 実際のソース、プログラム メンパー			
27イ25 yー 2175 yー 2175 yー 2175 yー	- 10 0P6586 NIS3 - 66 19202 10:32-18 - 見越入力 (キー・基本)		

Migaro. Technical Award 2010

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 合鐵産業株式会社 経理部 システム担当 係長



合鐵産業株式会社 http://www.go-sun.co.jp/

70年余りの歴史を持つ鉄鋼メー カー「合同製鐵(東証1部)」の販 社として設立。鉄鋼材の専門商社と して事業を営む。特に、建築用の鋼 材に関しては原料仕入から仕上げま で一貫して自社で加工を行い、製品 の付加価値を高めている。

受発注の全社管理

当社は、合同製鐵グループの鉄鋼商社 として事業を行っており、現在に至るま でグループ内での統合を繰り返してき た。そのため、業務手順や管理資料が部 署・個人ごとにバラバラで"標準化"が 全く図れていなかった。

特に、受発注に関しては、営業担当者 がプライベートに管理資料を作成/管理 している状態が多く見受けられ、外部か らの状況把握が困難で大きな人的リスク を伴っていた。

ビジネスの基軸である鉄鋼製品に関し て言えば、当社の流通という立場からも、 受発注の管理は欠かせない。そこで、正 確な受発注の状況を把握し、情報を会社 全体でオフィシャルに管理/共有できる 仕組みを導入して、システム化をするこ ととなった。もちろん、業務手順の統一、 資料等の様式の標準化を行うことも必須 であった。【図1】

Delphi/400の 選定理由

Delphi/400 は、現状の Excel 資産を 生かすことができ、リアルタイムに収集 される情報を活用する手段として導入を 行った。

多々あるアプリケーションの開発ツー ルの中から、Delphi/400を選定したの は、主に以下のような理由である。

(1) IBM i との高い連携力

現状の RPG や CL を流用できること。

(2) 開発時間の短縮

RPG や CL を混在させたまま開発が できるため、Delphi で複雑なプログラ ムを開発する必要がなく、結果として全 体の開発工数を抑えることができる。

(3) データ取得の容易性

IBM i 内のデータを、Excel の形で容 易に取得できる。

受発注管理システムの 再構築

システムを再構築する上で、次の3項 目が必要最低限の前程条件となった。

【受発注管理の前程条件】

- (1)業務手順の統一、資料等の様式の 標準化
- (2) 事務作業の効率化
- (3) 鉄鋼 EDI によるデータ送受信

(1)業務手順の統一 資料等の様式の標準化

業務手順を統一することは管理上、必 要不可欠であるが、反面、営業担当の個々 人の営業スタイルを否定するようなこと にもつながる。

営業の担当者には誰しも特有のスタイ ルがあり、それに自信を持ってビジネス に取り組んでいる。実績さえ上げていれ ばかまわないではないかとか、業務プロ セスに第三者が口を挟むのを快く思わな い担当者も少なくなかった。



そんなユーザーサイドを動かすには、 仕事を管理されている感をできるだけ感 じさせないことが肝要だ。

「システム化するとこんなにも便利に なる」ということを前面に押し出しての、 受発注管理システムの導入アピールを開 始した。

(2) 事務作業の効率化

現在の Excel 作業や 5250 画面での入 力等からの変更によって、以前より負荷 がかかってはいけない。当然、Delphi/ 400 の新システムは使い勝手の悪いもの であってはいけない。

現在の煩雑な入力を簡易にするべく、 Delphi/400の機能を生かしたアプリケー ションを開発する必要があった。作業の 効率化という面から次の工夫をしている。

①入出力操作の簡素化(容易性)
 ②既存システムとの連携

【改善ポイント:見積書からのデータ入力】 アプリケーション開発における独創 性・創意工夫について、詳細を述べる。

まず、営業担当者が使用していた見積 書を統一した。見積書はExcelフォーマッ トに統一されており、新しい受発注管理 システムには、事務作業者がその見積書 をそのまま取り込める方式を採用した。

入力フォーマットを Excel にするこ とで、営業担当者は、今までと同様の作 業で受発注処理が行える。また、事務作 業者は、対象の Excel データを受発注 システムのデータに再利用できるので、 入力時間の削減、入力ミスの削減を実現 できる。【図 2-1】【図 2-2】

(3)鉄鋼 EDI による データ送受信

鉄鋼業界には、メーカーとの取引を統 一されたデータ規格でやり取りする「鉄 鋼 EDI (Electronic Data Interchan ge)」がある。

規定フォーマットでデータのやり取り をするのであるが、EDIのフォーマット に変換するソフトは現況、PC上動作す るものばかりであった。また、データの やり取りを行う際、当社には当時、IBM iの全銀手順しかない状態であった。 当社の受発注業務は、FAX や電話等

でのやり取りが中心であったが、取引先

の電炉メーカーからの要望により EDI 経由で発注する必要が出てきた。

実際に EDI の発注処理を動作させる と、以下の流れとなる。

 IBM i (ジョブスケジュール):EDI 用のデータを作成
 PC (タスク):FTP での GET 処理
 PC (タスク):EDIのフォーマット変換
 PC (タスク):FTP での PUT 処理
 IBM i (ジョブスケジュール):全銀 手順での送信

現在のシステム環境下では、①~⑤の プロセスは、ジョブスケジュールやタス ク任せの連携のない動きとなり、トラブ ル時の解決をスムーズに行えない。また、 臨時に手動で対応する場合には、非常に 不便である。

上記内容を一連で行えるサーバー起動 のソフトがあるが、価格が高価である。

Delphi/400 を使えば、上記の問題は 解決できる。

Delphiのアプリケーションを主体に 動作させることにより、上記の流れを一 連で動作させることを実現した。

発注処理以外にも出荷請求、PCの時 刻を監視させた Delphi アプリケーショ ンを常駐させ、定時刻になると、Delphi から発注の送信や出荷請求の取得を一連 で行うようにした。

データ変換については、変換ソフトの DLL を Delphi のアプリケーションから Callすることによって実現している。【図3】

エンドユーザー評価

【プラス評価】

●事務作業の時間短縮とミス削減

これまで、会議資料等を定期的に作成 するためには、印刷帳票や個人の Excel データをベースにするしかなく多大な時 間を費やしていた。今回のシステムの導 入により、時間をかけることなく、最新 に近い業務データにもとづく会議資料の 作成が可能となった。

機能的には、特に、Excel データで出 力できるという利便性が評価されてい る。また、手作業時に比べ集計ミスも減 少した。

今後は、Excel 連携がこのように簡単 にできるのなら、現在、5250 画面で管 理しているリストについても画面照会や リスト出力ではなく、Excel データに出 力し、業務データの加工を自分で行いた いという要望が多数あった。

●アプリケーションの操作性と表現力 の向上

新アプリケーションでは、マウスによ る直感的な操作が可能になった。若い世 代に好評である。

また、従来の 5250 画面では不可能な、 柔軟な表現が可能になり、非常に好評で ある。例えば、項目数の多いサブファイ ル(3行で1列)が単行で表示できる等 である。【図 4-1】【図 4-2】

【マイナス評価】

●画面表示のパフォーマンス劣化

遠隔地(東京、福岡など)から、従来 より画面表示のスピードが遅いというク レームが入った。今後、PC 自体の性能 アップやネットワーク回線の改善などで 対応の予定である。

●クライアントPCへのアプリケーション配布(システム部門)

従来は IBM i 上にアプリケーション をコンパイルするだけで OK だったが、 今回よりクライアント PC へのアプリ ケーションの配布の手間が発生した。自 動配布もしくは自動取り込みが今後の課 題となっている。

今後の展望

Delphi/400 は、AS/400 のレガシーシ ステム全般を GUI 化していく目的で導入 した。今後は、照会画面および帳票を中 心にDelphi/400の開発を行っていきたい。 照会画面は、使用頻度の高いものから優 先的に GUI 化していく予定である。また、 帳 票 について は、専用 帳 票 以外 は、 Delphi/400 を利用して Excel 形式でエン ドユーザーに提供し、情報活用はエンド ユーザーに自由に行ってもらうことを基本 方針として、開発する予定となっている。

運用面では、モジュールのクライアン ト PC 配布の仕組みを改善していきたい。 プログラミングについては、当初不慣れ な面はあったが、今後開発を進めていく 中で、生産性を向上できると考えている。



Migaro.Technical Award 2010 優秀賞

Delphi/400で EIS (Executive Information System)の高速化 —リアル性の高いデータの提供へ

小島 栄一 様

西川計測株式会社 営業統括本部室 情報企画部



西川計測株式会社 http://www.nskw.co.jp/

横河電機の総合代理店。技術商社と して、制御情報機器、計測器、分析 機器等の販売と、それに伴うエンジ ニアリング、ソフトウェア開発、計 装工事、保守サービスなどを提供し ている。

EISとは

「Executive Information System」と は、日々の営業活動で発生する財務系情 報および販売管理系情報を加工し、経営 情報として経営幹部層に提供するシステ ムである。

EIS 情報は、AS/400 内の財務系情報 と販売管理系情報を集計し、ミドルウェ ア(転送ソフト)を使って1時間に1回 のバッチ処理で、EIS 用 DB (Oracle) に転送を行っている。

問題点

次のようなことが、課題になっていた。

●AS/400内の集計バッチ処理の終了時 間が不明のため、集計処理の終了時 間を最大時間に設定して転送ソフト を起動しなければならなかった。(全 体処理時間の遅延)

- ●集計するバッチ処理において転送障 害が発生することがあり、正しいデー タが提供できない状況があった。
- ●2009年に財務系システムのSuper Streamを導入したことにより、シス テムが分断された。そのため、勘定 系情報の集計操作が手動になり、運 用が煩雑となった。(集計漏れの発生)
- ●一度締めた月を再集計する場合は、 複雑な手順を踏む必要があった。
- SuperStream 側において前月分を再 配賦処理した後、結果をAS/400 に 取り込む。
- ②取り込んだ後、AS/400 側で再集計する。
 ③集計結果を、EIS 用 DB (Oracle) に 転送する。

Delphi/400/400 に よる解決ソリューション

課題解決のために、Delphi/400/400 を用いて、EISの集計システムを全面的 に再構築することを決定した。さらに、 ミドルウェア(転送ソフト)を使って連 続処理することで、一連の処理時間を短 縮することも検討した。【図 1】【図 2】

- ●Delphi/400 で AS/400 から集計対象 月を読み込み、SuperStream 内の経 費を配賦するプログラムを、SQL コ マンドで起動する。
- 経費配賦したファイルを Delphi/400 で AS/400 に書き込み、全体を集計 する CL を起動する。
- ●前月分の締めが未完了の場合、当月の月初の数日間は、今まで前月分しか集計できなかった。
 - →前月分の締め前でも、前月分だけ でなく今月分も集計ができる。
- ●一度締めた月を再集計する場合、今までは複雑な手順を踏む必要があった。
 → Delphi/400 で再集計したい月を指定するだけで、簡単に再集計処理ができる。
- ●上記処理中に発生したエラーは、すべてエラーログとして出力する。さらに、異常の発生時は、処理担当者にメールで異常を通知する。



苦労 / エ夫したところ

ミドルウェアで、本プログラムを連続 処理するようにした。そのため、夜中の AS/400 停止中にアクセスすると、エ ラーが発生して処理が停止してしまう。 対策として、AS/400 が再起動する時刻 まで待機することにした。

導入メリット

- ●現状1時間掛かっていた処理時間が、 15分~20分に短縮された。
- ●転送障害がなくなり、正しいデータ を提供できるようになった。
- ●経理部門での手動の集計操作がなく なり、集計漏れがなくなった。
- ●月締め後の再集計処理が簡単になり、 工数が大幅に削減した。

今後の計画

EIS を全体的に再構築することによ り、EIS 用 DB(Oracle)を AS/400 で 構築し、基幹システムとのシームレスな 連携を実現する。それにより、今よりさ らにリアルタイムで、精度の高い経営情 報を提供することを目指している。

Μ

Migaro.Technical Award 2010 優秀賞

イントラでの PHP-Delphi-RPG連携

■中ティーテー1家 西川リビング株式会社 経営システム室 課長代理



西川リビング株式会社 http://www.nishikawa-living. co.jp/

「眠り」から「健康」を創造し、より快適な暮らしを提案する西川リビング。時代のニーズにあわせた健康 機能商品や新商品の開発を行ってい る。創業 1566 年、本年 2010 年 は 444 周年である。

はじめに

西川リビングは、本年 2010 年に 444 周年を迎えた、創業 1566 年の寝具・寝 装品の製造卸業社である。

当社商品の流通業務の対象は、寝具専 門店様、量販店様、百貨店様など非常に 幅広い。

今回のシステム開発は、とある大手量 販店様への納品関連業務で必要となっ た、システム開発案件である。

背景 : イントラ環境で RPG を再利用

当社では、IBM i で Delphi/400 や RPG の社内システムを自社開発し、運 用している。

Web 関連のプログラミングについて は、当初 Java 言語習得を目指しチャレ ンジした時期もあったが、スキル面や開 発工数の面から PHP を採用し、現在に 至っている。 今回のシステムの環境としては、Win dows サーバー上に Apache + PHP 環 境を構築し、PHP アプリケーションか ら ODBC 経由で DB2/400 にアクセス している。

そういった状況で、課題として頭の中 にひっかかっていたのは、イントラ環境 でビジネスロジックがつまった RPG を 活用し、再利用できないかということで あった。

難解な RPG のビジネスロジックを PHP で書き直すには、あまりに無駄が 多く、テスト工数も当然かさんでくる。

クライアント / サーバー型アプリケー ションでの展開という選択肢もあったの だが、アプリケーション配布は配布作業・ バージョン管理などの手間がかかってし まうため、できるだけ避けたかった。

さらに、IBM i 内ではサブシステムご とに区画が分かれており、Delphi/400 が導入されている区画以外のデータにも アクセスしたいという課題も抱えていた。

要件:物流センター納品 情報システム

いくつか課題がある中で、当社物流部 より本格的にシステム化への要望の声が あがった。

要件としては「大手量販店様からの依 頼で、先方物流センターへの納品分につ いて最終納品センター別の個数、才数を 集計したい」というものであった。【図1】

現行業務としては、先方の統合物流セ ンターへ一括納品しているのだが、最終 納品センター(各店舗に納品する物流セ ンター)単位の物量を、先方システムで ある Web 画面で報告(入力)する、と いう必要が発生したのである。

ソリューション : PHP + Delphi/400 + RPG

要件を実現するには、Delphi/400 導 入区画である基幹システム区画(以降、 基幹区画と記載)以外の、物流サブシス テム区画(以降、物流区画と記載)にア クセスする必要があった。加えて、イン トラ環境で稼働させたいという要件も 持っていた。

この2点をポイントとして考察した結 果、「PHP + Delphi/400 + RPG」とい う組み合わせで実現できないか、を検討 してみることにした。【図2】

まず、物流区画の出荷予定ファイルか ら、基幹区画のワークファイルを作成す る RPG プログラムを作成する。続いて、 TCall400 コンポーネントを使用して、 RPG を起動する Delphi/400 プログラム を作成した。さらに、Delphi/400 プロ グラムを起動し (exec 関数を使用する)、 ワークファイルを読み込み、表示する P HP プログラムを作成した。

ちなみに、厳密にいうと実際に Delphi/400でキックするオブジェクト は、OVRDBFを定義したCLプログラ ムを利用している。【図 3】

結論: TCall400 コンポーネントは有効

個人的には、今回のシステム開発に独創 性や工夫はそこまでないかと考えている。

ただ、「PHP から RPG を動かせれば いろいろと使えるのに・・・」という悩 みが、実際に「Delphi/400 と連携させ てみたら動いた」というきっかけになっ たと思う。

TCall400 コンポーネントは非常に便 利で、価値のある IBM i を最大活用す ることができる Delphi/400 の機能だと 思う。「これが PHP(イントラ環境)か ら使用できたら・・・」という思いで、 今回のシステムの開発に取り組んだ。

TCall400 コンポーネントを利用する ことで、パラメータのやり取りも非常に 容易に、かつコーディング量も少なくて CLプログラムを起動することができた。 TCall400 コンポーネントだけに限らな いが、Delphi/400 の持つ IBM i 専用コ ンポーネントの存在に改めて感心した。 【ソース 1】

TCall400 コンポーネントが、Delphi/ 400 でのクライアント / サーバー型アプ リケーションでも、PHP などの Web ア プリケーションからでも使用できること は、当社にとって非常に有効な機能とい う結論に至った。 展望:「難解なビジネスロ ジック」を使用したイントラ

Delphi/400 が PHP (イントラ環境) で使用できる、ということは確認できた。 ただ、不特定多数のユーザーからのリ クエストにどこまでたえうることが可能 なのか、という検証はできていない。今 回のシステムのユーザーは2~3名程度 で非常に少ない。排他制御を含め、検討 課題はまだまだあると考えている。

引き続きこれらの課題に取り組み、課 題をクリアできれば、冒頭で記述した「難 解なビジネスロジック」を Delphi/400 で使用したイントラ、そして、インター ネットシステムに構築し、展開したいと 考えている。

今回使用した PHP 環境自体は無料で ある。低コストな環境をフルに生かして、 高品質な開発を検討していきたい。

また、Delphi/400 での Web アプリ ケーションにも、一度チャレンジしてみ たいと考えている。

評価:可能性に期待

「PHP + Delphi/400 + RPG」の新シ ステムは、レスポンス面では、稼働速度 も問題なく、非常に単純な画面というこ ともあり、これといったトラブルもなく 稼働している。

Delphi/400 ミドルウェアが高速であ るので、エンドユーザーにとってもスト レスのないレスポンスを実現できている と思う。

今回 PHP と組み合わせることで、ま た1つ新しい Delphi/400 の活用手段を 発見することができた。今後も工夫しな がら Delphi/400 のさまざまな機能を活 用していく予定で、Delphi/400 の可能 性に多いに期待している。

Μ



図2



図3

-	Ŷ	-			9	▶ sān []= * = congore
BRIG C	個数·	才数集計				
SッチNa.]取引先CD(44	4iî) :	持込センタ		[集計]
出荷倉庫	112F	取引先CD	持込センタ	NG-X0名	個數	才数
	140		-1	関西	-	
				関西	1	-
				九州		
				広島		
				EE.		
				情感纳		
				開岡		
1				中部		
				東北		
				兵庫		
				北海道		
				北陸		

```
ソース1
```

program Project1;

{ \$APPTYPE CONSOLE}

uses

SysUtils, Unit1 in 'Unit1.pas' {DataModule1: TDataModule};

var

i,n:integer; p1,p2,p3:string; strSQL:string;

begin

```
// パラメータ取得
 n:=ParamCount;
 if n > 0 then
 begin
  for i:=1 to n do
  begin
   if i = 1 then
    begin
     p1 := ParamStr(i);
    end;
    if i = 2 then
    begin
     p2 := ParamStr(i);
    end;
    if i = 3 then
    begin
     p3 := ParamStr(i);
    end;
  end;
 end;
```

{TODO -oUser -cConsole Main : この下にコードを記述してください} //RPG プログラムキック

datamodule1 := TDataModule1.Create(nil); try unit1.DataModule1.AS4001.Userid := 'USERID'; unit1.DataModule1.AS4001.PWD := 'PASWRD'; unit1.DataModule1.Call4001.Connect; unit1.DataModule1.Call4001.libraryname := 'LIB'; unit1.DataModule1.Call4001.ProgramName := 'PG001C'; unit1.DataModule1.call4001.Value[0] := p1; unit1.DataModule1.call4001.Value[1] := p2; unit1.DataModule1.call4001.Value[2] := p3; unit1.DataModule1.call4001.Execute;

finally

datamodule1.Free; end;

unit1.DataModule1.con.Close; end.

26

Migaro.Technical Award 2010 優秀賞

Delphi/400を使った 取引先管理システム

森定興商株式会社 システム運用部 係長



森定興商株式会社 http://www.morisada.jp/

森定興商は、パイプや鋼材、建材な どの生産材を取り扱う商社として、 今年で80周年。日本のインフラ整 備や"サステイナブル(持続可能)" な社会のために「変革と挑戦」の姿 勢を重視、80余年変わらない誠意 で変化に柔軟な組織づくりを目指し ている。

1. Delphi/400 に 巡り合う

当社の商事部門では、新たに顧客と取 引するためにまず、「取引稟議書」とい う Word 文書を作成・印刷して、印鑑決 裁にて上層部に回り最終決裁がおりる。

取引開始の決裁がおりたら今度は、「取 引先カード」という Excel 文書を作成・ 印刷する。そして基幹系販売システムで 使えるようにするために、「取引先コー ド申請書」を作成する。「取引稟議書」「取 引先カード」で入力した同じ内容を、基 幹系端末画面から入力・印刷し、印鑑決 裁にて上層部に回る。その後、私が所属 するシステム運用部に「取引先コード申 請書」が回ってきて、ようやく販売シス テムの「取引先マスター」に登録が完了 し、ユーザーが利用できるようになる。

最終的に販売システムで使えるまでに ユーザーは、同じ内容を3回も入力しな ければならない。

こんな姿を苦々しく見ていたある日、 Delphi/400というものに巡り合う時が 来た。2006 年夏のことである。

当社の基幹系システムは、RPGでの 自社開発を行ってシステム化がほぼ完了 していた。そんな基幹系システムだが、 グリーンスクリーンの画面は上層部から の受けが悪く「今風の Windows ライク な画面にせよ!」という声が挙がっていた。

そんな折、ある展示会で見かけたのが Delphi/400 で、これなら既存のプログラ ムにさほど手を加えずに満足いくものが つくれそうだと直感した。早速ミガロ. 主催の「Delphi/400 体験セミナー」に 申し込んで実際に触れてみて、これなら ばと確信。上司を説得し即購入に至った。

2. Delphi/400の トレーニングコース

私は入社以来 RPG 開発一本槍で来て おり、Delphi/400 での開発は遅々とし て進まなかった。そんな中、2007 年に 我がシステム運用部に新人社員が配属と なった。

当初彼女には、基幹系システムの開発

に携わってもらうつもりでいたが、Del phi/400を導入したので CL や RPG の 開発ではなく、Delphi/400の開発要員 として育てようと思った。

何とか予算を獲得して、彼女を Delphi /400 のトレーニングコースに派遣した。 入門トレーニングと基礎トレーニングを 終えた彼女は、仕事に意欲的に遊び心も 生かしながらアプリ開発を学んでいった。

私はというと、既存の RPG アプリを Delphi 化するためのプログラム仕様書 をせっせと作っていた。そして Delphi/ 400 の開発を彼女に託した私はプログラ ム仕様書を彼女に渡し、必要とあれば RPG でサブプログラムを作った。

そして Delphi/400 を使ったアプリが 徐々に増えていった頃である。

「取引先管理を Delphi で一元化する」 という話が上司から持ちかけられた。



取引先管理を Delphi で一元化。冒頭

でも述べたが、当社の取引先管理という のは、書類のために担当者が振り回され 無駄な労力を使っている。そこに Delphi /400 を使って「取引稟議書」「取引先カー ド」「取引先コード申請書」が i5 でデー タベースを共用し一元管理できるように なればどれだけユーザーの負担が減るこ とか。 早速「取引先管理システム」の開発チー ムを発足し開発着手するわけだが	露目を行い、忌憚のない意見を頂戴し修 正を行った。 構想から数年を経て今ようやく、ユー ザーの負担を軽減するという第1の目標 はクリアした。今後ユーザーからの声を 反映してまた、メンバーへの仕事の配分 を上手く行い、よりよいものに育ててい きたい次第である。
ムを発足し開発着手するわけだが、 Delphiの開発ができる者は彼女1名し かいない。その彼女は RPG は分からな いので、私からのプログラム仕様書待ち となる。 その他に、一貫して RPG での開発を 行ってきた上司に、データベース設計と システム全体像についてのアドバイスを 依頼。 上司の思い描くイメージを私は咀嚼 し、Delphi/400 で開発する彼女へそれ を伝える橋渡し役を担った。 当初私は、Delphi/400 でインター フェース部分をつくり、パラメータの受 け渡しで CL や RPG プログラムを動か すつもりでいた。そのために Delphi /400 と CL/RPG の連携をいかにスムー ズに行うか、に頭を悩ませていた。 データベースの更新処理を行う部分 で、レコードを呼び出した後にキー フィールドの内容が変更される場合のレ コード取得方法をどうするか? どうに も行き詰まった私は彼女に相談した。 「Delphi でレコードのアップデートがで きますよ」 RPG のようにキーで CHAIN してと いうことではなく、1 レコードそのもの がキーのような Delphi ではキーフィー ルドの内容が変更されても問題がないの だという。	森定興商株式会社(詳細) 1927年(昭和2年)5月に「森定雄商店」と じて鉄鋼販売業をスタート。戦後の1950年(昭和 25年)からは鋼管ならびに一般鋼材の販売を行い、 1964年3月に「森定興商株式会社」に商号変更し、 2000元を行う商事部門、大径溶接鋼管・異型管・高 欄・防護柵・鋼構造物の製造を行う主産部門、住宅 300元支合う商事部門、大径溶接鋼管・異型管・高 欄、防護柵・鋼構造物の製造を行う主産部門、住宅 300元支合う商事部門、大径溶接鋼管・異型管、高 欄、防護柵・鋼構造物の製造を行う主産部門、住宅 300元支合、 200元支合、 </td
以後私は一人で抱え込むのを止めた。 4. 取引先管理を Delphi で一元化 メンバーに対し適切な仕事の配分を行 うことにより、取引先管理システムは 着々と進んでいった。また、当初予定し ていたサブプログラムが大幅に減った。 これは Delphi/400 でできる部分のサブ プログラムが不要になったためである。 こうして 2010 年の初夏。我が部内と 管理部署および営業部署の上層部でお披	

ド」「耳 タベー なれば とか。 早速 ムを発 Delphi かいな

4. § Del

	図1 メニュー画面		
	新規申請データ作成		
	変更申請データ作成 《基本情報·取引法事業所情報変更》		
	限度額変更申請データ作成 《增粋+1 很活+ 臨時枠+特別/枠》		
	申請データ変更 《訂丁正·削印念》		
	申諸データー関係者意見		
	申請書名印刷		
	図2 新規データ作成画面		
	() 新規申23子→内存成 () 基本価格入力 (**は入力必須得です () 基本価格入力 (**は入力必須得です () 第24年の1997 (***********************************		
	Ref (地名) (1000-1000-1000) (1000-1000-1000-1000) (1000-1000-1000-1000-1000-1000-1000-100		
	(注例(*) 愛知毎名古屋市中40区を祝(その他) 電話番号(*) 「52-000-0000 「52-000-000 「52-000-0000 「52-000-0000 「52-000-0000 「52-000-0000 「52-000-0000 「52-000-0000 「52-000-000 「52-000-0000 「52-000-000 「52-000 「52-000-000 「52-000-000 「52-000-000 「52-000		
	(代表者 保護協会 (代表者 主年月日 1950/01/01 (代表者主年月日 1950/01/01 (代表者自治		
	代表省略語 二 <u>ゴ</u> 波本会 30000 万円		
	制護準月日 [1971,0210 設立準月日 1971,0210 決算月 ロッゴ 月 専業所 東京、名古里、大阪、道明		
	(25月) 300 名 (男)(20 名·女)(20 名) 25年10日 「即 アル杰 ア加江 ア製造		
	振発先 第二カルロ®、第三カルロ®と対応 は入え 第二カルロ®、第三カルロ®と対応 は入え 第三カルロ®、第三カルロ®と対応		
	☆独築種 	※ 学習 (100 mm : 二)	
	図3 関係者意見入力画面	図4 既存照会画面	
	④申請データー開係者意見 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■		
	与信限度額と意見入力 ・ 担当 ド可お卵いします。 ・		
	3,000万円 課長 C 所長		
	氏名 0万円 1 ① 初田 1 1	63	
	氏名山田 涼 10 万円		
	 ● 副支店長 C 部長 C 工場長 氏名 福田 陰司 10 万円 		
	 ○ 支店長 ○ 名古屋工場担当役員 ▲ 氏名 福山文彦 		_
	0万円 マ で 宮菜本部長 エ		_
1			



Migaro. Technical Report 2010

ミガロ.SE 論文/ミガロ. テクニカルレポート

中嶋 祥子

株式会社ミガロ.

RAD事業部 技術支援課

Delphi/400:ローカル キャッシュ活用術

ClientDataSetの利用を中心に Delphi/400 でのローカルキャッシュの有効な活用方法を紹介する。

●はじめに ●使用コンポーネント ●データ操作 更新処理 ●注意点 ●まとめ

1. はじめに

クライアントデータセットを使用する と、サーバー上のデータをローカル PC のキャッシュ上に保持できるが、そのメ リットとは何だろうか。

Delphi/400 は、クライアントデータ セットを使用しなくても、IBM i (AS/ 400) との通信パフォーマンスが高い。 だが、キャッシュ上のデータにアクセス することで、サーバー (IBM i) との 通信回数を減らすことが可能になり、さ らにパフォーマンスのよいアプリケー ションが作成できる。それだけでなく、 キャッシュ上にあるという特性を生かし た機能が用意されているため、さまざま なオペレーションが簡単に実現できる。

2. 使用コンポーネント

ローカルキャッシュ上でサーバー上の データを扱う場合、TClientDataSet を 使用する。その時にサーバーへ接続する データセット(TTable、TQuery 等) とこの TClientDataSet をつなぐため に、TDataSetProvider もあわせて利用 する。

TTable のデータを TDBGrid に表示 するプログラムを例にしてみよう。

通常であれば、4つのコンポーネント がプロパティで関連付けられている。こ れに、TDataSetProvider、TClientData Set の2つのコンポーネントを加え、プ ロパティの関連付けを行うだけで、ロー カルキャッシュ上でデータを扱える。

なお、TDBGridに表示するデータセッ トは TTable から TClientDataSet にな るため、TDataSource のプロパティを TClientDataSet に変更しておく。【図1】

以上の処理で、TClientDataSetを操 作すると、キャッシュ上のデータを操作 することになる。また、TTableと基本 的な操作は同じであるため、違和感なく 利用できる。



1968 年 2 月 23 日生まれ 1990 年奈良女子大学大学家政学部卒 2002 年株式会社ミガロ . 入社 2002 年 11 月 RAD 事業部配属

現在の仕事内容

お客様からのDelphi/400 に関す る技術的な質問や問い合わせに対応 している。また、メルマガ「Migaro News」やホームページのTipsなど、 開発に役立つ情報も担当。

3. データ操作

実際に、どのようなことが行えるのか を説明していこう。

レコード移動

First、Last、Next、Prior などの標準 のデータセットメソッドが使用できる。

また、キャッシュ上にデータを保持す る TClientDataSet 独自の方法として、 RecNo プロパティでレコード番号を指定 して位置付けを行える。【ソース1】【図2】

なお、この RecNo プロパティは、現 在のレコード位置を取得することもでき る。【ソース 2】【図 3】

また、リレーショナルデータベースで は、基本的に RecCount プロパティで レコード件数の参照は行えないが、 TClientDataSet では取得可能である。 【ソース 3】 [図 4]

フィルタ処理

TTable はフィルタ処理でレコード抽



出を行えるが、TClientDataSet でも可 能である。【ソース 4】【図 5】

画面で条件を指定して、そのデータを 抽出するケースを考えてみよう。

SQLを使用して条件抽出する場合、 SQLは条件を変更する度に「サーバー への要求」→「サーバー上での処理」→ 「サーバーからの送信」という処理が発 生する。

一方、TClientDataSet は、キャッシュ 上にデータがすでに取り込まれているた め、それが不要になる。つまり処理が高 速で行える。

インデックスの作成

TClientDataSet は自由にインデック スを変更できるので、データの並び替え も行うことができる。 方法には、次の2通りがある。

(1) IndexFieldName プロパティで指定

1つのフィールドで昇順に並び替えを 行う時、このプロパティにフィールド名 を文字列で設定する。

例えば、DBGrid のタイトルクリック 時にその列フィールドで並び替えを行う のであれば、ソースのように記述するこ とで、インデックスの切り替えが可能に なる。【ソース 5】

(2) IndexDefs プロパティで指定

こちらのプロパティを使用すれば、複 数のフィールドで並び替えができる。イ ンデックスを作成し、その Fields プロ パティでインデックス項目を指定する。

フィールドの区切りはセミコロン「;」 なので、例えば「日付 (HIZUKE)」「売 上金額 (URIK)」の順であれば、Fields プロパティで次のように記述する。

・Fields プロパティ値:HIZUKE;URIK

設定後、TClientDataSetのIndex Na meプロパティで指定することで、この インデックスが反映される。【図6】【図7】

また、特定のフィールドを降順にする こともできる。前述の Fields プロパティ のうち、降順にするフィールドを DescFi elds プロパティに記述するだけである。

例えば「売上金額(URIK)」だけ降 順にするのであれば、DescFields プロ パティに次のように記述するだけで、「日 付 (昇順)」「売上金額 (降順)」の順にデー タが並び変わる。【図 8】

・DescFields プロパティ値:URIK

上記のように、コード内で Fields プ ロパティと DescFields プロパティを書 き換えることにより、降順/昇順を切り 替えることができる。複数のフィールド を記述する時には、Fields プロパティ と同様にセミコロン「;」を使用する。

最後に、並び替えの項目を変更するに は、以下のようにする。

ソース例では、2つのインデックスを 用意し、切り替えを行っている。【ソー ス 6】

 IndexFieldName プロパティで指定 プロパティ値を変更する。

IndexDefs プロパティで指定

IndexDefs プロパティで詳細なイン デックス情報を複数設定し、IndexName プロパティで IndexDefs プロパティの設 定名を変更する。

集計項目

TClientDataSet には集計項目と呼ば れるものがあり、合計値や平均値などが 取得できる(※)。また図表に、集計演 算子をまとめた。【図 9】

利用するには、データセットに項目を 追加する方法を用いる。「項目の新規作 成」で、「型」を Aggregate、「項目の種 類」は集合体にすると項目が追加される。

例えば、合計値を求めたい時には、Ex pression プロパティに SUM 演算子を使 用した式を記述する。また、Active プロ パティを True にしなければならない。

表示は TDBEdit を使用すると、TDB Grid 上の明細データ値を変更した時にも、 即座に反映される。その際、項目の表示 形式を Alignment や DisplayFormat プ ロパティで設定できる。【図 10】【図 11】

※注意

集計対象となるのはキャッシュされているレコー ドだけなので、後述の PacketRecords プロパティ を指定している場合、全レコードでなくキャッシュ されているレコード内の集計や平均となる。

計算項目/内部計算項目

通常のデータセットでも計算項目は存

在するが、TClientDataSet には計算項 目とは別に、内部計算項目がある。計算 項目はレコードが変更されたり項目編集 が行われる度に発生するのに対し、内部 計算項目はデータとして保存され、項目 として扱うことが可能である。

分かりやすい例として、計算項目で ソートする場合を見てみよう。

まず、計算項目として粗利を作成する。 【図 12】 【ソース 7】

実行してみると、TDBGridへの表示 は問題なく行われる。【図 13】

次に、計算項目である「粗利」でソー トを行うために、ボタンで記述し実行し てみると、エラーが発生してしまう。 【ソース8】【図14】

これを内部計算項目に変更してみよう。 変更する場合には、計算項目フィールド の FieldKind プロパティを fkCalcula ted から fkInternalCalc に変えることで 行える。この状態で再びソートを行うと、 正常に並び替えが行われている。【図 15】

計算項目では内部計算項目と異なり、 データが保存されずその度に計算される ので、ソートを行う場合などには利用で きない。その際には、内部的に保存され る内部計算項目を使用する。

4. 更新処理

キャッシュ上での更新

更新処理を行うメソッドは通常のデー タセットと同じで、Insert、Edit、Dele te、Postを使用する。また、TDBGrid 上で直接操作したり、TDBNavigator から行うことも同じである。さらに、 CancelUpdates メソッドを呼び出すと、 すべての変更を取り消すことができる。

サーバーへの更新適用

キャッシュ上のデータ変更をサーバー 側へ反映させる処理が必要であるが、 ApplyUpdates メソッドを呼び出すだ けである。

ただし、キャッシュ上のデータ変更を サーバー側へ一括で反映させるという機 能上、トランザクション処理が必要であ る。BDE 接続の場合、以下の2点を行う。


 ● IDCO400ドライバのパラメータの TRAN_ISOLATIONが、*NONE 以外の値が設定されていること。 (値は*CHG、*CS、*ALLで、IBM i上のコミットメント制御に同じ)。

● IBM i(AS/400)上で、対象ファイル のジャーナルが開始されていること。

エラーの取得

ApplyUpdates メソッドの引数は、そ の数のエラーが更新時に発生した場合に 処理を停止する。

ApplyUpdates (-1) と引数を「-1」に すると、エラー数に制限がないことにな るが、「0」にするとエラーが発生した時 点で更新処理が停止する。

なお、戻り値は発生したエラー数を返 すので、エラー発生時にメッセージを表 示したり、トランザクション時にロール バックしたりといった制御を行うことも できる。【ソース 9】

また、TDataSetProviderのOnUpda teErrorで、ApplyUpdates時のエラー を捕らえることができる。【ソース10】

5. 注意点

データ量

TClientDataSet はとても便利な機能 であるが、一旦すべてのデータをキャッ シュ上に取り込むため、データ量が膨大 だと時間が非常にかかることがある。そ の場合には、PacketRecords プロパティ を利用する方法が考えられる。

例えば、TDBGrid にデータを表示す る場合、データ量があまりにも多いと全 件を読み込むには時間がかかるため、な かなかデータが表示されない。しかし、 PacketRecords プロパティの値を1画 面の表示データ件数分とすると、データ の表示が早くなる。

スクロールダウンする時には次のデー タの読み込みが行われるが、反対にスク ロールアップする時には、以前のデータ がキャッシュ上に残されているので、メ モリにアクセスし表示は速い。

しかし、スクロールの途中で読み込み が完了していない状態では、キャッシュ 上にはすべてのデータが存在しておらず、 集合体やソート処理などが行えないため、 せっかくの機能が十分活用できない。

このため、膨大なデータの時には全件 を取り込むのではなく、ある程度サー バー上でデータを絞り適切な件数になる ようにした方がよい。それによって本来 のTClientDataSetの機能を十分活用し つつ、速度も得られるようになる。

レコードの順序

文字コードの関係で、AS/400 では並 び順が「アルファベット→数字順」にな るが、PCでは「数字→アルファベット順」 となる。

このためローカルに取り込んだ時、イ ンデックスが英数混在の項目の場合、 TClientDataSet では PC 上の並び順に なる。もし IBM i (AS/400) での並び 順とするのであれば、TDataSetProvider の Options プロパティにある poRetain ServerOrder を True に設定する。これ だけでレコードの順序が IBM i (AS/ 400) と TClientDataSet で一致する。

なお、別端末からサーバー上のデータ が更新される場合、一旦キャッシュ上に 取り込まれたデータと差異が生じること がある。このため変更されやすいデータ については、ローカルキャッシュの活用 は適していない。

midas.dllの配布

TClientDataSet を使用したアプリ ケーションの配布時についてだが、mid as.dll も同時に配布する必要がある。し かし、Delphi7 以降であればアプリケー ションの uses 節に midaslib を追加する 方法で、この dll の配布が不要になる。

6. まとめ

よく利用されるであろう機能を簡単に説 明したが、容易に利用できることやそのメ リットを実感していただけただろうか。

すべてを紹介しきれなかったが、興味 を持たれたのであれば、Delphiのヘル プにも記載されているので、ぜひ一度目 を通していただきたい。

なお、パフォーマンスについてはサー バー上での処理速度も当然ながら大きく 影響するため、その点の考慮も必要であ ることはいうまでもない。とはいえ、通 信回数を減らすという面では、クライア ントデータセットは非常に有益である。 RPGやCLの実行をはじめとするサー バー側の機能のみならず、ローカル キャッシュも十分活用していただき、 Delphi/400の性能を十分に引き出して もらいたい。

図9

沙世7	18:44
演算士	(族配
Sum	式または数値型の項目の値の合計
Ave	式,数値型または日付時刻型の項目の平均値
Count	空白ではない値を持つ式または項目の数を返す
Min	式あるいは文字列型,数値型,または日付時刻型の項目の 最小値を示す
Max	式あるいは文字列型, 数値型, または日付時刻型の項目の 最大値を示す

項目の新規作成				
項目のプロパティー 名前(1):	ARARI		コンボーネント(の):	ClientDataSet1ARAR
型(I):	Integer	*	サイズ(5):	0
項目の種類 〇データ(<u>D</u>)	() 参照(1)	C)集合体(<u>A</u>)
◎計算項目(⊆)	(○内部計算項	(B())	
参照の定義	-		-	-
The last of the lot of the lot		~	チータビット(品);	

•

図10



図11

			コード位置移動	現在のレ		金レコード件数	
			J 1 (2208-1946)	000000		EVD TITRA	
L-28	a≥		の以来行				
			in the second se				
1	云票番号	日付	得意先コード	売上金額	仕入金額		
	123013	2010/07/01	1221	125,000	100,000		
	123014	2010/07/01	1513	168,000	120,000		
1	123015	2010/07/02	2156	99,200	72,000		
1	123016	2010/07/02	3041	223,000	160,000		
	123017	2010/07/02	2156	456,000	368,000		
1	123018	2010/07/02	5384	12,000	10,000		
	123019	2010/07/02	1221	865,000	800,000		
	123020	2010/07/03	3984	1,215,000	1,055,000		
1	123021	2010/07/03	6516	368,700	330,000		
	123022	2010/07/03	1380	546,000	435,000		
1	123023	2010/07/04	1513	420,000	400,000		
1	123024	2010/07/04	2156	1,098,000	989,000		
1	123025	2010/07/04	2975	86,000	100,000		
1	123026	2010/07/04	2315	455,000	400,000		

				_	-	-
	V	コード位置移動	現在のレ	->ド位置	全レコード件数)
売上金額≧		7ィルタ実行				
伝票番号	日付	得意先コード	売上金額	仕入金額	粗利	
123013	2010/07/01	1221	125,000	100,000	25,000	
123014	2010/07/01	1513	168,000	120,000	48,000	
123015	2010/07/02	2156	99,200	72,000	27,200	
123016	2010/07/02	3041	223,000	160,000	63,000	
123017	2010/07/02	2156	456,000	368,000	88,000	
123018	2010/07/02	\$ 5384	12,000	10,000	2,000	
123019	2010/07/02	1221	865,000	800,000	65,000	
123020	2010/07/03	3984	1,215,000	1,055,000	160,000	
123021	2010/07/03	6516	368,700	330,000	38,700	
123022	2010/07/03	3 1380	546,000	435,000	111,000	
123023	2010/07/04	1513	420,000	400,000	20,000	
and the second se	2010/07/04	2156	1,098,000	989,000	109,000	
123024		2075	86,000	100,000	-14,000	
123024	2010/07/04	67/0				

図14



図15

プロパティ イベント Alignment tal AutoGenerate an ConstraintErro CustomConstr DefaultExpres	kightJustify None		志卜会相			コード位置移動	現在のレン	コード位置 (1	をレコード件数	
DisplayLabel AF	RARI		Planks M. C.	-		1///weit	(Har bere	TOBY		
DisplayWidth 10)		fz	原香号	日付	得意先コード	売上金額	仕入金額(*粗ギリ	2
EditFormat			E	123025	2010/07/04	2975	86,000	100,000	-14,000	1
Fieldkind 31	nternalLaic	-		123018	2010/07/02	5384	12,000	10,000	2,000	
HasConstraint	Falce		1	123023	2010/07/04	1513	420,000	400,000	20,000	
ImportedCons	11 000		11	123013	2010/07/01	1221	125,000	100,000	25,000	
Index 6				123015	2010/07/02	2156	99,200	72,000	27,200	
KeyFields		\sim		123021	2010/07/03	6516	368,700	330,000	38,700	
eldKind				123014	2010/07/01	1513	168,000	120,000	48,000	
			1	123026	2010/07/04	2315	455,000	400,000	55,000	
	and the second second	_		123016	2010/07/02	3041	223,000	160,000	63,000	
へての項目が表示	स्ताराज्य		1	123019	2010/07/02	1221	865,000	800,000	65,000	
				123027	2010/07/04	3151	780,000	700,000	80,000	
				123017	2010/07/02	2156	456,000	368,000	88,000	
				123024	2010/07/04	2156	1,098,000	989,000	109,000	
				123022	2010/07/03	1380	\$46,000	435,000	111,000	
								,		

```
ソース1
  1
  procedure TForm1.Button1Click(Sender: TObject);
   begin
    ClientDataSet1.RecNo := StrToInt(Edit1.Text);
   end;
ソース2
   procedure TForm1.Button2Click(Sender: TObject);
   begin
     ShowMessage('現在のレコード番号は '
+ IntToStr(ClientDataSet1.RecNo)
+ 'です');
   end:
ソース3
  procedure TForm1.Button3Click(Sender: TObject);
   begin
     ShowMessage('レコードは '
+ IntToStr(ClientDataSet1.RecordCount)
+ '件です');
   end;
ソース4
  procedure TForm1.Button4Click(Sender: TObject);
   begin
     with ClientDataSet1 do
     begin
       Filter := 'URIK >= ' + Edit2.Text:
       Filtered := True;
     end:
   end;
ソース5
   procedure TForm1.DBGrid1TitleClick(Column: TColumn);
   begin
     ClientDataSet1.IndexFieldNames := Column.FieldName:
   end;
```

```
ソース6
   procedure TForm1.Button5Click(Sender: TObject);
   begin
   if ClientDataSet1.IndexName = 'Index1' then
       ClientDataSet1.IndexName := 'Index2'
     else
       ClientDataSet1.IndexName := 'Index1';
   end:
ソース7
   procedure TForm1.ClientDataSet1CalcFields(DataSet: TDataSet);
   begin
     with
           DataSet
                     do
     begin
       FieldByName('ARARI').AsInteger := FieldByName('URIK').AsInteger
-FieldByName('SHIK').AsInteger;
     end;
   end:
ソース8
   procedure TForm1.Button5Click(Sender: TObject):
   begin
     ClientDataSet1.IndexFieldNames := 'ARARI';
   end:
ソース9
   procedure TForm1.Button7Click(Sender: TObject);
   begin
       f ClientDataSet1.ApplyUpdates(-1) > 0 then
ShowMessage('エラーが発生しました');
     if
   end:
ソース10
   procedure TForm1.DataSetProvider1UpdateError(Sender: TObject;
     DataSet: TCustomClientDataSet; E: EUpdateError; UpdateKind: TUpdateKin
     var Response: TResolverResponse);
   begin
     raise Exception.Create(E.Message);
   end:
```

尾崎 浩司

株式会社ミガロ.

システム事業部 システム3課

Delphi/400: 帳票開発ノウハウ公開

業務に必須の帳票開発。Delphi/400 で使用できる帳票ツールや PDF ファイル出力連携など、帳票開発手法の具体例を紹介する。

●はじめに ●帳票レイアウト上での「計算項目」の活用 ●画像ファイルの取り扱い ●出力先プリンタの指定 ●XMLファイルをソースとした帳票出力 ●PDFファイルの作成 ●最後に



1973 年 8 月 16 日生まれ 1996 年三重大学工学部卒 1999 年 10 月株式会社ミガロ . 入社 1999 年 10 月システム事業部配属

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

1. はじめに

"帳票"を出力するには、どのような方 法があるだろうか?

IBM i ユーザーであれば、「OUTQ」 (スプール)を使用した帳票が一般的で あろう。また、以前からの Delphi/400 ユーザーであれば、「QuickReport」を 使用した帳票出力を行っている場合もあ るだろう。では、現在の Delphi/400 に バンドルされた標準レポートツールは何 であろうか? それは「Rave Reports」 である。

本稿では、この Rave Reports を中心 に、帳票開発に関するノウハウをトピッ クス形式で紹介したいと思う。

なお、Rave Reportsの基本的な使用 方法については、ミガロ.のホームペー ジに詳しく紹介しているのでそちらを参 照していただきたい。

Rave Reports 講座 http://www.migaro.co.jp/contents/ products/delphi400/tips/report/3_01 /index.html

2. 帳票レイアウト上での 「計算項目」の活用

Rave Reports の特徴は、QuickRepo rt 同様、Delphi/400の DataSet を直接 使用できることである。

DataSet には、データベースのフィー ルド情報を保持しているため、各フィー ルドをそのまま帳票出力項目として、レ イアウト設計に使用できる。さらに、レ ポート側に「計算項目」を定義すること で、実体のフィールドと同様に、レイア ウト設計で使用可能になる。例えば、デー タベース上に「数量」と「単価」しかフィー ルドを持たない場合でも、「金額」フィー ルドを「数量×単価」という計算項目と して定義できる。

ここでは、この帳票レイアウトにおけ る計算項目を取り上げ、いかに活用でき るかを見ていこう。

最初は、作成手順である。Rave Repo

rts の帳票レイアウト設計は、専用の「レ ポートデザイナ (Rave Reports Design er)」を使用して行う。【図 1】

まず、作成するレポートにある Para meters プロパティに、新規パラメータ Amount を定義しよう。この Amount が帳票で出力する計算項目となる。

次に、計算項目に使用する計算式を定 義する。ツールパレットから、CalcOp コンポーネントをレポートに貼り付けよ う。【図 2-①】

CalcOp コンポーネントには計算式が 指定できるので、下記のように定義する。 【図 2-②】

DestParam:出力用計算項目
 (ここでは「Amount」を指定)

- ・Operator:算術演算子を指定 (ここでは「乗算」を指定)
- Src1DataField、Src2DataField: 演算を行うフィールドを指定

 (ここでは「数量」「単価」の各フィールドを指定)

41

これで定義は完了だ。あとは、出力用 フィールドとして DataText コンポーネ ントを貼り付け、Amount フィールド を指定すれば、帳票出力時に「数量×単 価」から「金額」が計算され、結果が帳 票出力されるというわけである。【図 3】 このように、あらかじめ帳票レイアウ トに計算項目を使用し、計算式を埋め込

むことで、有効な帳票が実現する。その 際、帳票出力プログラムは、Rave Repo rtsにDataSet を渡すだけでよいのである。

3. 画像ファイルの 取り扱い

画像ファイルの帳票出力は、IBM i のOUTQからは難しいと思われるが、 Rave Reportsを使用すると容易に実現 できる。ここでは、帳票出力において画 像をセットする手順を紹介しよう。

Rave Reports での画像出力には、 Bitmap コンポーネントを使用する。こ のコンポーネントには、画像ファイルを 指定する FileLink プロパティが用意さ れているので、ここに出力する Bitmap ファイルを指定すればよい。

では、プログラムから動的にファイル を指定するにはどうすればよいだろう か?

Rave Reportsで作成するレポートは、 Delphi と同様にイベントを使用するこ とができる。つまり、画像ファイルが出 力される 直前に発生するイベント OnBeforePrintで、出力したい画像ファ イルを指定すればよいのである。【図 4】

ここでは、PictureFileStr というパラ メータを使用しているので、レポートの Parameters プロパティに PictureFile Str というパラメータを追加しておこ う。これで Delphi ソースから値を指定 することが可能になる。

Delphi ソースの記述例を示す。【ソース1】

これは、Edit1 コンポーネントに指定 された Bitmap ファイル名を、レポート のパラメータ PictureFileStr に設定し、 印刷する手順である。本プログラムを実 行すると、画面に示すように選択した ファイルが帳票出力される。【図5】 この仕組みを応用することにより、画 像を埋め込んだ帳票が容易に作成できる ので、グラフィカルで利用価値の高い帳 票開発にぜひ挑戦していただきたい。

4. 出力先プリンタの 指定

Rave Reports で作成した帳票をその まま印刷しようとすると、デフォルトで 「印刷指示」画面が表示される。この画 面では、出力方法や印刷枚数、出力先プ リンタ等の指定が行えるため大変便利で あるが、残念ながらすべて英語表記と なっている。【図 6】

では、この印刷指示面を使用せずに帳 票出力するにはどうすればよいだろう か?

帳票出力プログラム

帳票出力に関する設定情報は、TRvS ystem コンポーネントで管理できる。 つまり、このコンポーネントを使用する ことで、印刷指示画面の設定自体をプロ グラムで行うことが可能になる。 設定箇所は、次の通りである。【図7】

・DefaultDest:出力先の指定 (プレビュー / 印刷)

・SystemSetup/ssAllowSetup: 「印刷指示」画面の表示有無

これで印刷指示画面を経由せずに、帳 票出力が可能になる。

ソース例を示す。【ソース 2】 このソースは、例えば、Delphiの フォーム上に TragioGroup(コンポー ネント名:rgSelect として定義)を配 置し、選択値に応じて「プレビュー」あ るいは「印刷」を切り替える場合の記述 である。

出力先プリンタの指定

これで、印刷指示画面を経由せずに、 Rave Reports が使用可能となるわけだ が、このままでは印刷時には、常に「通 常使うプリンタ」へ出力されてしまう。 では「出力先プリンタ」を切り替えるに はどうすればよいだろうか?

「出力先プリンタ」を切り替えるロジッ クを記述するには、デバイス定義ユニッ トである RPDevice を Uses 節に追加し、 RPDev オブジェクトを使用すればよい。 RPDev オブジェクトには、SelectPrint er メソッドが用意されているので、こ こに出力したいプリンタ名を指定すれば よいわけである。

ソース例を示す。【ソース 3】 Windows標準の「印刷用ダイアログ」(T PrintDia log)を表示し、選択されたプリ ンタに帳票を出力するソース例である。

5. XMLファイルを ソースとした帳票出力

データ連携といえば、以前はCSVファ イルを使用するのが一般的であったと思 う。現在でも利用する機会は多いと思う が、近年は、より柔軟な定義が可能な XMLファイルを使用することも多いだ ろう。【図 8】

ここでは、XML ファイルを使用して 帳票出力する手法を紹介したい。

XML を Delphi で扱う方法はいくつ かあるが、今回は TClientDataSet を使 用する方法を紹介しよう。

TClientDataSet はこれまでも何度か 紹介しているが、クライアントPCのメ モリ上に展開可能な DataSet である。 実は、TClientDataSet には、XMLを 読み込んで DataSet として利用する方 法がある。DataSet となれば、Rave Reports でそのまま使用可能になるとい うわけである。

では、どのようにすれば XML ファイ ルを DataSet として使用できるのだろ うか?

変換ファイル(拡張子.xtr)

これを実現するには、TXMLTransfor mProvider というコンポーネントを使 用すればよい。このコンポーネントは、 XML 文書から DataSet へ変換するため のものである。その変換には、専用の変 換ファイル(拡張子.xtr)を使用する。

実は Delphi には、XML ファイルか ら変換ファイルを作成するためのマッピ ングツールが付属している。Delphi 開 発画面上で[ツール]→[XML マッパー] を選択すると、画面のようなツールが起 動する。【図 9】

このツールより、対象となる XML ファイルを開くと、ドキュメントの構成 が左側のペインに表示されるので、

Rave Reports 7 ProjectIrav Elle Zoom Edit Icols Help	Drawing Bar Code Standard Report Zoom Colors Lines Fills Fonts Alignment
DataText1: DataText component Anchor (Top / Left)	ge Designer Event Editor
Color Black Pro	-20 0 , 20 , 40 , 60 , 80 , 100 , 120 , 140 , 60 , 80 , 100 , 120 , 140 , 50 , 50 , 50 , 50 , 50 , 50 , 50 ,
DisplayOn doParent Font MSゴシック,14 FontJustify pjLeft	
FontMirror Left 62.4 Locked False	
LookupDisplay LookupDisplay LookupTireld	
Mirror DataText1	タイトル 《DataText1
2	
PRAVE Reports 7 - C-*Docu	ments and Settinge¥XPMIIserXデフカトゥナXRaveSampleXA Rave#トナルク(計算
<u>File Zoom Edit Tools Help</u>	B Crawing Bar Code Standard Report Zoom Colors Lines Ella Coste Aliza
CalcOp1: CalcOp component DestParam Amount	①「CalcOp」コンポーネント
DestPIVar DevLocked False DisplayForm #,0	
DisplayType dtNumericFormat Locked False Name CalcOp1	
Operator coMul ResultFunctior cfNone Sec1CaleVae	DestParam(結果) → Amount(計算項目)
Src1DataFiel JUPRSU Src1DataVie DataView1	Operator (演算子) → coMul (乗算) Src 1DataField → 数量フィールド
Src1Function cfNone Src1Value 0 Src2CalcVar	Src2DataField → 単価フィールド
Src2DataFie JUPRTN Src2DataVie DataView1 SrC2Function cfivone	B GRDrgb 1PC)
Src2Value 0 Tag 0 Wishle True	8 <u>数量 単価</u> (Master 1PC)
VISIDIE	- 1 Ifiprsii (.uprtni
3	
age 1 of 1 🔍 🔍 🖦 🛛	2 Zoom 1000 % E
	数量×単価が金額欄として出力
平	注商品明細
又	
間品名 DELPHI/400 DEVELOP	女X里 単1回 玉額 1 1,000,000 1,000,000
マウス	20 2,000 40,000
ゴフクレッゴの	

DataSet に変換したい項目をダブルク リックしていく。すべての項目を選択し たら、[作成]→[XML からデータパケッ ト]を選択しよう。すると、XML の各 項目がデータパケット欄に表示されるは ずだ。これでマッピングが完了である。

最後に「変換の作成とテスト」をクリッ クすると、図に示すように、データセッ トが表示されるのが分かる。(画面は、 XML ファイルを、実際に DataSet に変 換した結果である)。【図 10】

変換が完了したら、「保存」より変換ファ イル(拡張子.xtr)の保存を行う。これ で変換ファイルの作成は終了である。

あとは、TXMLTransformProvider のXMLDataFile プロパティに処理対象 となるXMLファイルを指定し、Trans formRead プロパティにある Transform ationFile に変換ファイルを指定の上、 TClientDataSet の ProviderName プロ パティに対象のTXMLTransformProvi der を指定すれば完成である。

このように、XML ファイルは簡単に DataSet に変換できるため、Rave Repo rts において、XML ファイルを使用し た帳票出力が容易であるということもご 理解いただけるだろう。

6. PDFファイルの作成

Rave Reports には、「PDF ファイル」 出力機能が用意されている。しかし残念 ながら、日本語環境化での動作はサポー トされていないのが実情だ。では、 Delphi/400 から PDF ファイルを作成 するには、どうすればよいだろうか?

最も簡単な方法は、PDF プリンタド ライバを使用する方法であろう。本家の Adobe Acrobat のみならず、近年は PDF 出力可能なフリーソフトも多数リ リースされているので、これらを出力先 プリンタに指定すれば手軽に PDF 出力 が可能である (※)。

※前述の「4. 出力先プリンタの指定」で解説した技 法を使用すれば、PDF プリンタの出力先指定も行 える。

ただし、この手法の欠点は、クライア ント PC 上に PDF プリンタドライバの インストールが必須となることである。 企業内の GUI アプリケーション等であ れば統一環境の構築も検討できるであろ うが、そうでない場合、クライアント PC の環境を固定化するのは困難であろ う。そういった場合は別の手法を検討す る必要が出てくる。

そこで今回ご紹介する方法は、Delphi で使用可能な、PDF 出力コンポーネン トである「PowerPDF」を使用する方法 だ。PowerPDF は、LGPL ライブラリと してソースコード付で公開されている。

PowerPDF0.91 for Delphi 2009 (Delphi/400 Ver2009 以降) http://mam.dnsalias.net/download/ delphi_powerpdf.html

PowerPDF0.9

(Delphi/400 Ver2007 以前) http://www.est.hi-ho.ne.jp/takeshi_ kanno/powerpdf/

PowerPDFを使用すると、クライア ントPCのプリンタドライバ導入状況に かかわらず、PDFファイルの出力が可 能になる。(作成されたPDFファイル の確認には、専用の閲覧ソフトが必要で あることはいうまでもないだろう)。

本ツールは、QuickReportのように、 Delphiフォーム上に帳票レイアウト設 計が可能なコンポーネント群となってい る。【図 11】

この PowerPDF の基本的な使用方法 は、次の通りである。【図 12】

 プロジェクトに帳票用フォームを新 規作成する。

②下記コンポーネントを貼り付ける。

- TPReport: PDF 出力を管理するコン ポーネント
- TPRPage: PDF ファイルの1ページ を表わすコンポーネント
- TPRLayoutPanel:単票形式のパネル コンポーネント

③ TPRLayoutPanel 内に出力用コンポー ネントを貼り付ける。

- ・TPRLabel:タイトル (表題)を表わす コンポーネント
- TPRText:複数行表示可能なコンポー ネント

④ソース4のようなプログラムを記述 する。【ソース4】

- FileName:出力先 PDF ファイル名 を指定
- ・ BeginDoc: PDF 作成の開始
- Print:指定したページの PDF を出力
- ・ EndDoc: PDF 作成の終了

なお、PDF1 ページを表わす TPRPa ge は、コンポーネントサイズが大きい ため、TScrollBox に置いて使用すると 便利であろう。

このように、PowerPDF を使用すれば 帳票の PDF 化も容易である。ただし、 Rave Reports とは互換性はないため、 従来のプリンタ出力と PDF 出力と両方 が必要となる場合は、2 種類の作り込み が必要となることにご注意いただきたい。

7. 最後に

今回は、Delphi/400から作成する帳 票として、Rave Reports および Power PDF という2つの帳票ツールのトピッ クスを紹介した。

特に、これまで IBM i の OUTQ (ス プール)を使用していた方は、画面の GUI 化同様、帳票の表現力が向上する ので、ぜひ一度これらのツールを使用し てみてほしい。Delphi/400 からの帳票 出力が、容易でかつ効果的あることを実 感いただけるであろう。

また、ミガロ.ホームページには帳票 に関するトピックスも多数掲載している ので、参考になれば幸いである。



	図8		
	xml version="1.0" er</td <td>coding="ISO-2022-JP" ?></td> <td>_</td>	coding="ISO-2022-JP" ?>	_
	- <住所録> - <個人情報 ID=" 1 "> <名前> 升村 承 名</td <td>新 、</td> <td>_</td>	新 、	_
	<郵便番号>920-09 <住所> 金沢市菊川 1	67 郵便番号 -9-6 住所	_
	<電話番号>076-26 個人情報	1-4921 電話番号	
	<名前>北陸太郎 <名前>北陸太郎 <多卵便番号>920-00	名前> 00 郵便番号	
	<住所>金沢市大手■ <電話番号>076-11	【1-1-1 住所 1-3333 /電話番号	
	/ □/ □ (〒₩2)<br - <個人情報 ID="3"> <名前>金沢 花子 </td <td>名前></td> <td></td>	名前>	
[図9		
	3. XML マッピングツール ファイル(E) 作成(C) ヘルプ(H)		
	☞ ■ × ドキュメント : addr00.xml	変換: <未指定> データパケット: <未指定>	
	□-ⓒ(住所続 □-□ 個人情報[[*] □-▼[] @ID	道択されたノード 道択されたフィールド	
	- <u>国</u> 名前 - <u>国</u> 郵便番号 - <u>国</u> 住所 - 国 住所		
		○ クライアントデータセット(C) ○ 前「除余デルタ(D) ○ 問(い合わせパラメータ(Q)	
		 ○ 1回加テルタバケット(1) ● 変換の方向 ○ データパケットから XML(A) ○ XML からデータパケット(M) 	
	「データ表示(y)	変換の作成とテスト	
	ドキュメント表示 【スキーマ表示】 ドキュメント: C:¥Documents and Settings¥XPf	<u>ノードプロパティ_マッピング</u> User¥デスクトップ¥XML_Sample¥addr00.xml	
		F	
	□ ×mL マチビリクラール ファイル① 作成② ヘルプ(!!) □ □ ■ ×		
	ドキュメント: addr00.xml	変換: <自動生成> データパケット: <自動生成>	
	□ [1] [1] [1] [1] [1] [1] [1] [1] [1] [1]		
	- [1] 郵便番号 - [1] 住所 - [1] 電話番号	#1上別が転半回し入情報紙「1=950度世番 #KOWDATA#KOW[1]@150度世 ¥1主所録¥1個人情報紙*191主所 ¥1主所録¥1個人情報紙*194重話番 ¥ROWDATA¥ROW[*]@1電話番 ¥1主所録¥1個人情報紙*194電話番 ¥ROWDATA¥ROW[*]@1電話番	
	馬 テスト変換	XML ドキュメント -> データパケット	
	□□名前	1. 郵便番号│住所	
	▶1 升村丞 2 北陸太郎 3 金沢花子	920-0967 金沢市荀川1-9-6 076-261-4921 920-0000 金沢市大手町1-1-1 076-111-3333 920-012 金沢市片町1-2-3 076-123-4567	
	- ドキュメント : C	FDocuments and Settings¥XPMUser¥デスカトップ¥XML_Sample社 「 データパケットから XML(A) で XML からデータパケット(M)	
	□ データ表示(V)	変換の作成とテスト	
	ドキュメント表示	ノードプロパティ マッピング フィールド表示 データパケット表示	



	ソース1	
	40 begin	
	//レポート選択 SelectReport('Report1', True);	
	· SetParam('PictureFileStr', Edit1.Text); - // 印刷実行	
	Execute; end;	
	· end;	
	ソース2	
	 	
	// 振票プレビュー/印刷 選択 · if_rgSelect.ItemIndex = 0 then	
	RvSysteml.DefaultDest := rdPreview // フレビュー else 40 RvSysteml DefaultDest := rdPrinter: // 印刷	
	with RyProjecti do begin	
	ソース3	
	儲ホームページ 局 SampleFm · · □	
	30 uses <u>RPDevice</u> . Printers;	
	[\$R *.dfm]	
	- =procedure TfrmSample.Button1Click(Sender: TObject); - begin - //印刷ダイアログを表示する	
	if not PrintDialog.Execute then Exit;	
	RPDev.SelectPrinter(Printer.Printers[Printer.PrinterIndex], False);	
	with RyProject1 do	
	ソース4	
	 □ procedure trmsample.Button Ulick(Sender: Ubject); · begin 28 // 埋方ガイアロガム美テ 	
	if not SaveDialog1.Execute then Exit;	
	<pre>frmPDFPrpt := TfrmPDFPrpt.Create(Self); try</pre>	
	· // レポート作成 · with frmPDFPrpt do	
	$\begin{array}{c} - & \text{Degin} \\ \cdot & // \vec{\tau} - \varphi \overline{\varepsilon} \overline{\varepsilon} \nu \epsilon \\ \end{array}$	
	PRText2.Text := Memol.Text;	
	50 // <i>PDF出力</i> PReport1.FileName:=SaveDialog1.FileName;	
	PReport1.BeginDoc; PReport1.Print(PRPage1);	
	- PReport I.EndDoc; end; fipally	
	frmPDFPrpt.Release;	
	end;	
48		

辻林 涼子

株式会社ミガロ.

システム事業部 システム2課

Delphi/400で ドラッグ&ドロップを制御

ドロップ&ドラッグの操作により Excelファイルをシステムに取り込む方法を紹介する。

●はじめに ● Drag&Dropとは何か ● Delphiで実現するためには ● Windows Messageとは ● 実装方法 ● 応用 ● 最後に



1987 年 3 月 3 日生まれ 2009 年近畿大学農学部卒 2009 年 04 月株式会社ミガロ . 入社 2009 年 04 月システム事業部配属

現在の仕事内容 入社2年目で、主に Delphi/400 を使用したシステムの受託開発を担 当している。

1. はじめに

アプリケーションを開発する際に、設 計者としては考えなければいけないこと は多数あるが、必須条件として主に2つ あると考える。

1つは内部の設計をシンプルにし、保 守性や拡張性を見越した設計にすること である。またもう1つは、ユーザーが満 足する操作性を実現することだ。特に ユーザーには、直感的な操作が分かりや すく使いやすい。

本レポートでは、直感的な操作として 普段何気なく使われる Drag&Drop に ついて、Windows での仕組みと Delphi に実装する方法を紹介したいと思う。

2. Drag&Dropとは 何か

Drag&Dropとは、マウス操作の1つ で、マウスでファイルやデータをクリッ クし選択したまま、移動(ドラッグ)さ せ別の場所でクリックしたボタンを離す (ドロップ)ことである。これは、Win dowsが提供している機能の1つである。 【図 1】

この機能について大きく分類すると、 以下の3つに分けられる。

- ●アイコン、フォルダへの Drag&Drop ファイルを、別フォルダに移動やコ ピーできる。
- ●実行中のアプリケーションへの Drag &Drop

メモ帳にテキストファイルをDrag&D rop すると、ファイルを開く。

●アプリケーション内での Drag&Drop Excel では、セルの移動ができる。

これらは Windows が、指定した Win dow、今回はドラッグ元のファイルとド ラッグ先のファイルに対して、Drag&D rop の機能を呼び出しており、それぞれ の処理が行われている。

また、受信側のアプリケーションに よって、Drag&Drop を許可するものと しないものがある。これは、Delphiで実 装する上でも当然必要な制御となる。 Word、Excel、メモ帳などのファイルを 開く機能のあるアプリケーションの多く は、Drag&Dropを制御する機能が実装 されている。が、その他のアプリケーショ ンでは、受け付けないものも見られる。

さらに、Drag&Drop を受け付けた場 合でも、ファイルの種類により制限され ることも多い。これは、PowerPointの ファイルをメモ帳、Excel に Drag&Dr op してみると、そのことが理解できる。

メモ帳では、文字化けして読めない文 字列が表示されるものの、メモ帳のテキ ストファイルを開くという処理としては 正しい対応がなされる。【図 2-1】

これに対し Excel では、「ファイルが 開けません」というエラーが出る。こち らは、Excel では PowerPoint のファイ ルを受け付けないよう指定されているか らである。【図 2-2】

さて、今回は上記から、アイコンへの Drag&Drop、実行中のアプリケーション への Drag&Drop、これらによる外部ファ イルの取り込みについて解説していく。

3. Delphiで 実現するためには

実際に Delphi で実現するためには、 いくつか情報が必要である。必要となる 情報とその取得方法について、アプリ ケーションのパターン別に説明しよう。

アイコンにドロップインして アプリケーションを起動する場合

実際のところ、特に処理をしなくても Delphi で作成したプログラムアイコン は、ドロップインを受け付けるように なっている。しかし、受け入れ後の処理 ロジックを作成していないため、何も起 こらない。

何かを起こす処理ロジックを作成する には、受け入れたファイル名が必要であ る。コマンドライン引数を利用すれば、 簡単にそのファイル名が分かる。コマン ドライン引数とは、プログラム実行時に 指定される引数のことで、Delphiでは ParamCount と ParamStr を用いて取 得できる。

コマンドプロンプトから実行する例を 見ていただけると分かりやすい。【図 3】 コマンドプロンプトでは、以下の形式 になっている。

アプリケーションのパス(半角スペース) 開くファイルのパス

ParamCountでは、このパラメータ(開 くファイルのパス)の数を取得する。 ParamStrでは、指定した番号のパラ メータを取得する。ただし、ParamStr の0番目はアプリケーションのパスが 入っているため、番号を指定する場合に は注意が必要だ。

実行中のアプリケーションに ドロップインする場合

実行中アプリケーションでファイルを 受け取る場合、Windows に、アプリケー ションが Drag&Drop を受け入れるこ とを伝えなければならない。この手続き を行うのが、Windows APIの DragAc ceptFiles 関数である。

API に WindowHandle と True (ド

ロップされたファイルの受け入れ許可指 定)を渡すと、フォーム上で Drag&Drop の受け入れが可能となる。

WindowHandle とは、Message を送 信する際に画面上のオブジェクトを識別 するために、個々の要素に割り当てられ る一意の番号のことである。フォームや アイコンなどの Drag&Drop の受け入 れを許可するオブジェクトは、この Win dowHandle で指定する。

DragAcceptFiles を許可すると、ファ イルがドロップされた時、フォームに WM_DROPFILES という Windows Me ssage が送られる。ただし、この Windo ws Message は、Drag&Drop が行われ たことを伝えるだけで、実際のファイル 名やパスは取得できない。

従って、DragQueryFile 関数を用い て、ファイルのパスを取得する必要があ る。この API は、複数のファイルが渡 された場合でも、1つのファイルしか取 得できないが、何番目のファイルを取得 するかの指定は可能である。

また、パラメータのファイル番号部分 に「\$FFFFFFFF(= -1)」を指定した 場合には、いくつのファイルがドロップ されたのかを返す仕組みとなっている。

なお、必要なファイル名が取得できた ら、最後に DragFinish 関数でメモリを 開放する。これを行わなかった場合、使 用可能なメモリが減っていき、Windows がフリーズする場合があるので必ず行う ようにする。

4. Windows Messageとは

前章で出てきた「Windows Message」 について、少し詳しく説明する。

Windows Message と は、Windows (OS) がそれぞれの Window を制御す るために使っているもので、これらを Delphi で意図的に記述することにより、 Windows が行っている動作を取得した り、逆に動作するようにしたりできる。

例えば、ユーザーがマウスやキーボー ドで入力した時、それらはまず Windows によって取得される。その後、Windows は、それぞれのアプリケーションに Win dows Message として送信する。アプリ ケーションは、Windows Message を受 け取ることによって、ユーザーがどのよう な入力を行ったのかを知ることができる。 【図 4】

●" Hello!"の例

Delphi でイベントを記述する場合、 例えば、Create イベントに"Hello!"と いうダイアログを表示する場合について 考えてみよう。

このアプリケーションを実行した場 合、まず Windows がフォームを作成し、 Windows から Delphiへ WM_CREATE の Windows Message が送信される。

Delphi ではその Windows Message を受け取り、Delphi の中で Create イベ ントに書かれた内容を実行するよう、 Windows 側に Windows Message を送 信する。

すると、"Hello!"と書かれたダイアロ グが表示され、ダイアログ内のボタンを 押すと、それがまた Windows Message で Delphi に送信され、「ダイアログを出 す」という1文が終了する。

このように、Create イベントや Dest roy イベント等の多くの Windows Messa ge 取得は Delphi が自動で行うため、特 に意識せず使用することができる。【図 5】

他方、Drag&Drop の Windows Mes sage (WM_DROPFILES) は、対応し ている標準のイベントがないため、意図 的に DragAcceptFiles を True に設定 する必要がある。これによって、Drag &Drop の Windows Message(WM_DR OPFILES) を取得することができるよ うになる。

5. 実装方法

実際に簡単なサンプルを作成する。

アイコンにドロップインして アプリケーションを起動する場合

例として、ドロップしたファイルパス を表示するだけの簡単なプログラムを作 成する。これは Form の OnCreate また は OnShow のイベント時に、ParamStr を使用すればよい。単純に取得したパス を表示するだけのプログラム例である。

まずは、コンポーネントをフォームに 追加する。



【使用するコンポーネント】

・TPanel:ツールパレット (Standard)
・TMemo:ツールパレット (Standard)
・TBitBtn:ツールパレット (Additional)

次に、コンポーネントのプロパティを 設定する。今回は TBitBtn の Kind を 変更する程度で、他はデフォルトの設定 でかまわない。

あとは、OnCreate または OnShow の イベントに、ソースを記述すれば完成だ。 【ソース 1】

実行すると、画面のようになる。【図6】

実行中のアプリケーションに ドロップインする場合

例として、csv ファイルを Drag&Dr op すると、先ほどのプログラムと同様 に、ファイルパスを画面上の Memo に 表示するプログラムを作成する。

こちらでも、まずはコンポーネントを フォームに追加する。

【使用するコンポーネント】

・TPanel:ツールパレット(Standard)

- ・TMemo:ツールパレット(Standard)
- ・TBitBtn:ツールパレット(Additional)
- ① uses 節に ShellAPI ユニットを追加 する。

今回必要な DragAcceptFiles 関数、 DragQueryFile 関数、DragFinish 関数は、すべて ShellAPI ユニットに 格納されている。【ソース 2】

- Create 時に DragAcceptFiles 関数 を呼び出す。【ソース 3】 Handle は自身のフォームの Handle、 True は許可を意味している。
- ③ private 部に使用する手続きと Wind ows Message を追加する。【ソース4】
 ④手続きの処理を記述する。【ソース5】

処理は以上である。実際に動かすと、 画面のようになる。【図 7】

アイコンとアプリケーションのどちら の場合も、一度に取得できるファイル名 は1つだけなので、複数ある場合には繰 り返し文を利用する。

また、アイコンヘドロップインした場 合、ParamStr で取得される0番目はア プリケーション自身のファイルである。 一方、アプリケーションにドロップイン した場合、DragQueryFiles で取得され る0番目は、アプリケーション自身では なくドロップインされたファイルであ る。これらにも注意が必要だ。

6. 応用

これまでの内容を用いて、少し実践的 なプログラムを紹介する。今回は、実行 中のアプリケーションにドロップインす る場合を例とし、追加する形で紹介する が、アイコンにドロップインする場合に も、ほぼ同じ記述で使用できる。

次の処理では、取得するファイル数を 1つに限定する。また、拡張子を csv 形 式に限定する。取得できたファイルパス から csv ファイルのテキストを、TStri ngList を用いて StringGrid にセットす るまでのロジックを追加していく。

【追加コンポーネント】 ・TStringGrid:ツールパレット(Additional)

 ⑤コンポーネントを追加する。今回は sgListと命名している。
 前章の手続きを、ソース例のように 変更する。【ソース 6】

ソースについて説明していこう。

ExtractFileExt は、文字列から末尾 に最も近いドット「.」を探し、それよ り後の部分を返す関数で、主に拡張子の 取得に使用される。今回は csv に限定 するために使用している。

StringGrid への入力は、まず変数 slCSV1 (TStringList) にファイルのテ キスト内容をすべて保持する。TString List に入力された文字列は Count プロ パティ で 個数 が取得 できる ため、 slCSV1 の Count プロパティから csv ファイルの明細行数を取得できる。

次に、取得した行数を StringGrid に 設定し、テキストファイルを1行ずつ変 数 slCSV2 (TStringList)の Commma Text プロパティにセットする。

もともと1行であった文字列を Com mmaText プロパティにセットすると、 カンマ「,」部分で区切った文字列のリ ストに変換される。

行数と同様に Count プロパティから

列数を取得し、取得した行数・列数から StringGrid のセルに入力していく。 実行すると、以下のようになる。【図8】

- 図 8-1:正常な場合 図 8-2:複数ファイルをドロップインした 場合
- 図 8-3:csv 以外のファイルをドロップ インした場合

これらをさらに応用すれば、簡単に csvからAS/400へ、内容を確認・修正 しながらファイル転送するプログラムな どの作成が可能である。同様に、OLE 等を用いてExcelファイルの読み込み といったことも可能となる。

7. 最後に

外部ファイルの取り込みや Windows Message の処理と聞くと難しい印象を 受けるかもしれないが、プログラミング してみると実に 100 行にも満たない記 述で実行できることが分かるだろう。

ユーザーの満足度が高いアプリケー ションを開発するために、Drag&Drop など、こういった直感的な操作をアプリ ケーションに組み込むことを、今後も設 計者として考慮していただければ幸いで ある。



図8-1

🍃 新しいフォルダ	 ℤ Drag&Drop S	ample3					
77イル(E) 編集(E) 表示(V) お気に入り(A) ツール(I	[CSV77	マイルをドラ	ラッグしてく	ださい。】			
アドレス(D) C*Documents and Settings¥新しいフォルダ	郵便番号	都道府県	市区	町村	市区(カナ)	町村(カナ)	
名前 A サイズ 種類 愛引test xis 146 KB Microso	5390000	大阪府	大阪市中	以下に掲	オオサカシチュウ	<u> </u>	
「テスト用CSVデータ2.csv 1 KB Microso 「テスト用CSVデータ3.csv 2 KB Microso	5410052	大阪府	大阪市中	安土町	オオサカシチュウ	79` F7F	
予ラスト用CSVデータcsv 1 KB Microso 1 KB Micro	5410047	大阪府	大阪市中	淡路町	オオサカシチュウ	アワシ゛マチ	
	5420061	大阪府	大阪市中	安堂寺町	オオサカシチュウ	アント・ウシ・マ	
	5400019	大阪府	大阪市中	和泉町	オオサカシチュウ	イズ・ミマチ	
	5400022	大阪府	大阪市中	糸屋町	オオサカシチュウ	イトヤマチ	
	5410042	大阪府	大阪市中	今橋	オオサカシチュウ	イマバシ	
	5420064	大阪府	大阪市中	上汐	オオサカシチュウ	ウエシオ	~
						<u> </u> 開じる(0

図8-2



図8-3



```
ソース1
  procedure TForm1.FormCreate(Sender: TObject);
   var
i : Integer;
   begin
     //1つ以上のファイルがDrag&Dropされた場合のみ行う
if ParamCount > 0 then
     begin
       //パラメータの数だけ繰り返す
       for i := 0 to ParamCount do
       begin
         //メモにパラメータを表示する
         Memo1.Lines.Add(ParamStr(i));
       end;
     end:
   end;
ソース2
   interface
  uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ComObj, StdCtrls, Grids, Buttons, ExtCtrls, ComCtrls, AppEvnts,
ShellAPI; // 追加する
  type
    TForm1 = class(TForm)
ソース3
  procedure TForm1.FormCreate(Sender: TObject);
  begin
//ドラッグ&ドロップを受け付け許可
    DragAcceptFiles(Self.Handle, True);
  end;
ソース4
  private
    //Windowsからのメッセージ WM_DROPFILES を WMDropFiles で受け取るようにする
    procedure WMDropFiles(var msg: TWMDropFiles);
    message WM_DROPFILES;
```

```
ソース5
   try
      //ドロップされたファイル数を取得
     FileNum := DragQueryFile(msg.Drop, $FFFFFFFF, nil, 0);
     for i := 0 to FileNum - 1 do
     begin
        //ファイル名の取得
        DragQueryFile(msg.Drop, i, FileName, SizeOf(FileName));
       Memo1.Lines.Add(FileName);
     end;
   finally
      //ドラッグ完了時、メモリの解放
     DragFinish(Msg.Drop);
   end:
  nd;
ソース6
   procedure TForm1.WMDropFiles(var msg: TWMDropFiles);
   var
     FileName : array[0..255] of Char;
    sFileName, sFileType : String;
FileNum : Integer;
                : TStringList;
: TStringList;
     sICSV1
     sICSV2
     i,j : Integer;
   begin
     try
       FileNum := DragQueryFile(msg.Drop, $FFFFFFF, nil, 0); //ファイル数の取得
       if FileNum <> 1 then // ファイル数を限定する
       begin
         MessageDlg('受け付けるファイルは1つのみです。', mtError, [mbOK], 0);
         Abort;
       end:
       End,
DragQueryFile(msg.Drop, 0, FileName, SizeOf(FileName)); //ファイル名取得
sFileName := FileName; //文字列型にする
     finally
       //ドラッグ完了時、ハンドルの解放
       DragFinish(Msg.Drop);
     end:
     sFileType := ExtractFileExt(sFileName); //拡張子取得
if sFiletype <> '.csv' then//エクセル以外の場合
     hegin
       ShowMessage('対応はcsvファイルのみです');
       Abort;
     end;
     11 Gridt= 入力
     sICSV1 := TStringList.Create;
     sICSV2 := TStringList.Create;
     //ファイルのオープン
     sICSV1.LoadFromFile(sFileName);
     sgList.RowCount := slCSV1.Count;
                                        //行數取得
     for i := 0 to sgList.RowCount - 1 do
     begin
       sICSV2.CommaText := sICSV1[i]; //一行分情報
sgList.ColCount := sICSV2.Count; //列数取得
       for j := 0 to sICSV2.Count -1 do
         sgList.Cells[j,i] := slCSV2[j];
     end:
     sICSV1.Free;
     sICSV2.Free;
     sgList.Col := 1;
     sgList.Row := 1;
   end;
```

株式会社ミガロ.

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

バージョンチェック用アプリケーションで バージョンの比較と最新モジュールの取得方法を紹介する。

●なぜExeの置き換えが必要? ●Exe置き換えの手法 ●自動置き換えの実現方法 ●実行環境の変化への対策 ●まとめ ●参考ソース



1983 年 11 月 21 日生まれ 2006 年大阪工業大学情報科学部卒 2006 年 04 月株式会社ミガロ. 入社 2006 年 04 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・保守対応などを担当して いる。

1. なぜExeの 置き換えが必要?

通常、システムを運用 / 稼働している と、アップデート (バージョンアップ) が必要不可欠である。機能拡張、あるい は不具合の修正など理由は多岐に渡る が、運用当初のままの状態でシステムを 稼働し続けることは難しい。

例えば、WindowsのOSなどに代表 されるパッケージソフトは、Service Packのように常に新しいパッチプログ ラムでアップデートが適用される。この パッチプログラムはインターネット経由 でユーザーに配布されたり、個別のイン ストーラーの形で提供されたりすること が多い。

他方、各企業で独自に開発している業 務アプリケーションにおいてはどうなる だろうか。

同様のアップデートの仕組みを実現す る場合には、もちろん別途に配布を行わ なければならず、その仕組みそのものを 構築する必要がある。

Exe(モジュール)

アップデートを実施する場合に気を付 けなければならないのが、最新のExe(モ ジュール)をいかに各クライアントPC に適用するかである。

アプリケーションを機能拡張した場 合、新しい Exe を、各クライアント PC の古い Exe と置き換える必要がある。

その時、プログラムの変更部分が Exe だけであれば、クライアント PC の Exe を順次置き換えるという対応をがとれ、 問題が生じる可能性も小さい。

しかし、IBM i (AS/400) など、デー タベース・サーバー側のプログラム (例 えば RPG) と連携したパラメータ部分 などが変更になった場合には、問題があ る。サーバー側の変更と同時に、クライ アント側の Exe の置き換えを行わなけ れば、アップデートが逆にエラーの原因 となってしまう。そういった場合には、 サーバー側の変更と同期をとって、クラ イアント PC 側の Exe も置き換えるこ とが必須になってくる。

もちろん Delphi/400 では、SQL など で、データベースの処理もすべて Exe 側 で行うことができる。そういった Exe に 処理を集約したプログラムであれば、ク ライアント/サーバー間におけるインター フェースの懸念は少なくて済むだろう。

2. Exe置き換えの手法

Exe の置き換えの必要性は前述の通り だが、ここでは Exe を置き換える手法 について整理してみる。大きくは2つの 運用手法が考えられるだろう。また、こ の2つの手法の特徴について列挙する。

(1)配布ツールを利用して、Exe置き換え を運用

【実現コスト】

- ・ツールの導入費用が必要である。
- ・ツールで置き換えの仕組みが提供されるため、開発が不要である。

【汎用性】

- ・複数のExeを管理する場合も、ツー ルですべて管理できる。
- ・Exe の置き換えだけでなく、新規 Exe の配布も行える。
- ・ツールとして提供されるため、機能の カスタマイズは行えない。

【Exe 置き換えのタイミング】

 ・Exeの置き換えはPC 起動時やスケ ジュールされたタイミングで行われる ため、それにあわせて新しいExeを配 布する必要がある。

(2)独自に仕組み、プログラムを構築して Exe置き換えを運用

【実現コスト】

- ・ツールの導入費用が不要である。
- ・置き換えの仕組み、プログラムの作成
 (開発工数とスキル)が必要である。

【汎用性】

- ・個々の Exe ごとに、置き換えのプロ グラムを用意する必要がある。
- ・新規に Exe を作成した場合、初回配 布は手動で行う必要がある。
- ・自動置き換えのプログラムに機能拡張 や修正が必要となった場合、自動置き 換えのプログラム自体の置き換えが必 要となる。
- ・必要に応じてカスタマイズした仕組み の開発が可能である。

【Exe 置き換えのタイミング】

・Exeの実行時に置き換えができるため、
 確実に新しい Exe を利用することができる。

このように2つの手法は、それぞれメ リット/デメリットとなる点が異なる。 開発の必要性や汎用性を考慮すると、 (1)配布ツールを使って運用する場合も 多いが、本稿では、Delphi/400の開発 テクニックとして(2)Exeの自動置き 換えの仕組みをプログラムで実現する例 を紹介したい。

3. 自動置き換えの 実現方法

ここからは、Exeのバージョンチェッ ク〜置き換えを行うにあたり、技術的な 実現方法について説明していきたい。 まず、自動置き換えを行うには、Exe を3つ準備する必要がある。

- AExe: チェック/ 置き換えを行うExe
- BExe:置き換えが行われる実行 Exe
- CExe:最新版の実行 Exe

AExe はバージョン情報をチェックし て、BExe を最新の CExe に置き換える。 その後、BExe を起動する。つまり、 AExe を起動することで、自動的に最新 に置き換えられた BExe が起動する仕組 みである。【図1】

AExe と BExe はクライアント PC の 同一フォルダに保持し、CExe はファイ ルサーバーなど、システム利用者がアク セス可能な領域で一元管理を行うものと する。なお、システム利用者の権限によ りアクセス制限が設けられている場合、 自動置き換えが実施できなくなるため注 意が必要である。

次に、ここで3つの Exe 構成をとる 理由について説明しておく。

最新版としての CExe はいうまでもな いが、AExe と BExe をわざわざ分ける のには理由がある。

本来、バージョン情報のチェックだけ であれば、BExe 単体でも可能である。 しかし、同時にファイルの置き換えを行 う必要があるため、BExe 単体では実行 中の自身を最新版の CExe と置き換える ことができない。そのため、バージョン 情報のチェックと置き換え処理を兼ねた AExe が必要となるのである。

バージョン情報

BExe と CExe の置き換え有無の判断 については、各 Exe ファイルが保持し ているバージョン情報を利用する。

そこで、Exe ファイルバージョン情 報の設定方法、バージョン情報を取得す る GetProductVersion 関数についてそ れぞれ紹介する。

バージョン情報の設定方法は以下の通り。【図 2】

①Delphiの [メニュー] → [プロジェ クト] → [オプション]

 ②バージョン情報タブの「バージョン 番号を含める」にチェックを付ける。
 ③「モジュールのバージョン番号」に任 意の番号を設定する。

また、バージョン情報の取得は、次の ようになる。 ソース例のように、関数を利用するこ とにより、BExe と CExe のバージョン 情報を取得 / 比較し、ファイル置き換え の有無をチェックする。【図 3】

引数 PGNM に指定したファイルの バージョン情報を、文字列として返却す る。なお、例外が発生した場合は、N/ A を返却する。【図 4】

自動置き換え

準備についての説明を終えたところ で、本題である実現方法の説明に話を進 めたい。BExe と CExe はバージョン チェック、置き換えに利用するだけであ り、ここからは AExe の動作について 詳細に述べる。

まず、AExe の動作は大きく以下の5 ステップに分かれる。【図 5】

ステップ①: BExe の絶対パスを取得 ステップ②: CExe の絶対パスを取得 ステップ③: BExe と CExe のバージョ ン情報を比較 ステップ④: バージョン情報が異なる場 合、最新版への置き換え

ステップ⑤:BExe を実行

AExe の動作(5つのステップ)について、少し掘り下げて説明を続けたい。

以下、Delphiソースを交えて説明を 行うにあたり、BExe、CExeの絶対パ スとして変数宣言が行われているものと する。【ステップ0】

BExe の絶対パスを取得

AExe と BExe が同一フォルダに保持 されていることを前提として、Extract FilePath 関数を利用して、BExe の絶対 パスを取得する。

また、ソース上の cExeName には、 実際に実行する BExe のファイル名(拡 張子含む)を定数として定義しておく。 【ステップ1】

CExe の絶対パスを取得

CExe の絶対パスは、別途準備した iniファイル(設定ファイル)より取得 する。

ini ファイルの利用については、次章 「4. 実行環境の変化への対策」で補足説 明を行う。【ステップ 2】



③ BExe と CExe のバージョン情報を 比較

 ②で取得した BExe、CExe の絶対 パスをもとに、それぞれのファイルの バージョン情報を取得する。CExe が常 に最新バージョンであるため、バージョ ン情報が異なる場合、BExe のバージョ ンを古いものとして扱う。【ステップ 3】

④バージョン情報が異なる場合、最新版 への置き換え

DeleteFile 関数で、クライアント PC の BExe の削除を行う。CopyFile 関数 で、サーバーの最新版 CExe を、クライ アント PC に BExe としてコピーする。 【ステップ 4】

⑤ BExe を実行

最後に、CreateProcess 関数で BExe を起動する。ステップ①~⑤の手順を踏 むことで、最新版への置き換え~ Exe 実行が実現可能となる。ユーザーからは 特に意識されることもない。【ステップ5】

実際に、図5のような構成でAExe を実行すると、BExeが「Ver.1.0.0.」 から、最新版のCExeへ置き換えられ、 それにより「Ver.1.0.0.1」にアップデー トされたことを確認できる。【図6】【図7】 【図8】

4. 実行環境の 変化への対策

システムのアップデート(バージョン アップ)が行われるのと同様に、システ ムを運用する実行環境についても変化が 訪れる可能性はきわめて高い。

例えば、接続先サーバーの変更、帳票 レイアウトの変更、接続ユーザーの変更 などである。当然、ファイルサーバーの 変更(フォルダの変更)も例外ではない。

仮に、チェック / 置き換えを行う Exe (AExe) に、ファイルサーバーの接続 先情報を保持しているとしよう。ファイ ルサーバーの変更があった場合、AExe 自体の置き換えが必要となってしまう。 それでは運用にたえない。

そこで、ファイルサーバーの接続先情 報を、iniファイルにより外部で管理す る手法を紹介したい。ここでは、前章の ステップ②に登場した、iniファイルを 例に説明を行う。

はじめに、iniファイルについて説明 する。iniファイル(設定ファイル)と は一般に、拡張子「.ini」を持つテキス トファイルを指し、iniファイルの記述 は以下の構成をとる。【図 9】

セクション:キーのグループ、[]で囲 んで記述する キー :名前=値のセット

次に、iniファイル情報を取得する GetIniFile 関数について紹介する。なお、 ソースの記述は、iniファイルが AExe と 同じフォルダに保持されていることを前 提としている。【ソース1】【ソース2】 ソースの設定の GetIniFile 関数を実

行すると、図9の AExe.ini に示したように、セクション: Common、キー: CExePath として、返却値には以下が設 定される。

C:\SampleSource\CExe\CExe.exe

つまり、iniファイルを利用すると、 ファイルサーバーの変更があった場合、 iniファイルの設定値を変更するだけで よく、容易な対応が可能となる。

5. まとめ

モジュールのバージョンチェック / 置 き換え手法というと一見難しい処理を想 像するが、参考ソースを見ていただくと 分かる通り、比較的少ないソース記述と iniファイルの設定で実装することがで きる。

初期導入として、チェック用 Exe と ini ファイルの配布は必須となるが、そ れ以降のアップデートに関してはサー バー上の最新版 Exe を置き換えること で運用できる。

たしかに配布ツールを利用した場合 は、専用のツールとなるので汎用性や導 入してすぐ使えるなどの優れた点も多々 あるだろう。

しかし、オーダーメイドのシステムで あれば、システム運用にあわせて Exe 置き換えの仕組みをカスタマイズしてい くなど、管理しやすくなる面がある。 また、本稿で説明したバージョンの取 得や ini ファイルの利用、ファイルの置 き換えといった開発テクニックは、さま ざまな応用ができる。Exe の置き換え 以外にも、おおいに利用していただきた い。これらの技術が、Delphi/400 のシ ステム開発 / 運用の助けとなれば幸いで ある。

図4	Ver.1.0.0.1の返却	
	評価/変更 🛛	
	評価する式(E):	
	GetProductVersion(sCExePath) 結果(R):	
	1.0.01'	
	新しい(値(N))	
図5	AExeの動作	
	START	
	◆ ステップ1)日 E × eの絶対パスを取得	
	BExeとCExeの バージョン情報を比較	
	異なる 同じ	
	ステップ4)最新版への置換え	
	ステップ5) B E × e を実行	
 図6	ノアイル構成	
	Mr SampleSource Mr CExe ファイルビロ 編集(ロ) 表示(ひ) お気に入り(山) ツール(ロ) ヘルブ(山) ア	
	・ ・	
 Ī	フォルタ ヘ 名画) ワイス 種類 フォルタ × 名前 サイス 種類 (2) デスクトック (3) 日本 1 KB 構成設定 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (2) デスクトック (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本 (3) 日本	
	Documents and Settings Documents and Se	
	 B → Inprise B → Lotus B → Program Files B → Program Files 	
	w □ rrojects □ SampleSource □ CExe	

図7-1 AExe実行前のBExeのバージョン情報	図7-2 AExe実行前のCExeのバージョン情報	
 BExe.exeのプロパティ ?X 全般 パージョン情報 互換性 セキュリティ 検索結果 概要 ファイル パージョン: 1000	CExe.exeのプロパティ 全般 バージョン情報 互換性 セキュリティ 検索結果 概要 ファイル バージョン: 1001	
図8 AExe実行後のBExeのバージョン情報	図9 AExeのIniファイル	
BExe.exeのプロパティ	● AExe.ini - メモ帳 ファイル(生) 編集(生) 書式(型) 表示(型) ヘルブ(生) [Common] CExePath='C:¥SampleSource¥CExe¥CExe.exe'	
 ソース1 iniファイル取	1	
<pre>{************************************</pre>	**************************************	
 ソース2 iniファイル関連の設定		
 // 全 <i>殿</i> cINIFile = 'AExe.ini'; // /N/. cExeName = 'BExe.exe'; // 置換	ファイル名 えExe (BExe)	
 // /N/ファイル cSection_Common = 'Common'; // セク	「ション 共通情報 - 長午町Exa(CExa)のパス	
 GNEy_GEXEFAIN - GEXEFAIN; // +-	ADD ANY LOC LABY GUYIA	

ステップ0	
var aBEvaPath · String· // BEvan 编付パフ	
 sDExerath : String; // CExeの絶対バス	
ステップ1	
 // BFxeの絶対パスを取得(ローカル)	
sBExePath := ExtractFilePath(Application.ExeName) + cExeName;	
ステップ2	
 // CEXeの超対ハスを取得(ゲーハー) sCExePath := GetIniFile(cSection_Common, cKey_CExePath, '');	
 // ハージョン情報チェック if GetProductVersion(sBExePath) <> GetProductVersion(sCExePath) then	
 ステップ4	
 // 毎新版への要換え	
 DeleteFile(PChar(sBExePath)); CopyFile(PChar(sCExePath), PChar(sBExePath), True);	
 ステップ5	
 // BExeの起動	
 <pre>if not CreateProcess(nil, PChar(sBExePath), nil, nil, False, CREATE_DEFAULT_ERROR_MODE, nil, nil, SI, PI) then</pre>	

```
参考ソース1 AExe.dpr
   program AExe;
   uses
     Forms,
     Dialogs,
     SysUtils,
     Windows,
     Controls,
     pasConst in 'pasConst.pas',
pasCommon in 'pasCommon.pas';
   {$R *.res}
   var
                            // BExeの絶対パス
// CExeの絶対パス
     sBExePath : String;
sCExePath : String;
                : TstartupInfo;
     SI
                : TProcessInformation;
: Integer;
     ΡI
     ÷
     sPrmStr
                : String;
   begin
     Application.Initialize;
     Application.Run;
     // BExeの絶対パスを取得(ローカル)
     sBExePath := ExtractFilePath(Application.ExeName) + cExeName;
     // CExeの絶対バスを取得(サーバー)
     sCExePath := GetIniFile(cSection_Common, cKey_CExePath, '');
     //サーバーに最新版Exeが存在する場合、チェック実行
     if FileExists(sCExePath) then
     begin
       // バージョン情報チェック
if GetProductVersion(sBExePath) <> GetProductVersion(sCExePath) then
       begin
         try
           // 最新版への置換え
           DeleteFile(PChar(sBExePath));
           CopyFile(PChar(sCExePath), PChar(sBExePath), True);
         except
         end:
       end:
     end;
     // 実行時引数の取得
     sPrmStr :=
     for i := 1 to ParamCount do
     begin
      sPrmStr := sPrmStr + ' ' + ParamStr(i);
     end;
     sBExePath := sBExePath + Trim(sPrmStr);
     // BExeの起動
     if not CreateProcess(nil, PChar(sBExePath), nil, nil, False,
                           CREATE_DEFAULT_ERROR_MODE, nil, nil, SI, PI) then
     begin
      raise Exception.Create('Exec Error' + IntToStr(GetLastError));
     end:
   end.
参考ソース2 pasConst.pas
   unit pasConst:
   interface
```

// INIファイル名 // 置換えExe (BExe)

// セクション 共通情報 // キー 最新版Exe (CExe) のパス

```
resourcestring
```

```
// 全級
cINIFile = 'AExe.ini';
cExeName = 'BExe.exe';
```

```
// INIファイル
cSection_Common = 'Common';
cKey_CExePath = 'CExePath';
```

```
implementation
```

```
end.
```

```
参考ソース3 pasCommon.pas
  unit pasCommon;
  interface
  uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    IniFiles, Registry;
  function GetIniFile(sSection, sKey, sDefault :string): string;
  function GetProductVersion(PGNM: String): String;
  implementation
  uses pasConst;
  目的: Iniファイル取得処理
引数: sSection - セクション
sKey - キー
        sDefault - デフォルト値
   戻値: 取得文字列
  function GetIniFile(sSection, sKey, sDefault :string): string;
  yar
    FileIni: TIniFile;
    sPath: String;
  begin
    // フォルダの指定
    sPath := ExtractFilePath(Application.ExeName);
    // INIファイルの読み込み
FileIni := TIniFile.Create(sPath + cINIFile);
    try
     Result := FileIni.ReadString(sSection, sKey, sDefault);
    finally
     FileIni.Free;
    end:
  end;
  目的: バージョン情報の取得処理
引数:
戻値: 取得したバージョン情報
  function GetProductVersion(PGNM: String): String;
  var
   Size, SizeFileInfo, Ret: DWORD; pData, pInfo: Pointer;
  begin
    try
     Size := GetFileVersionInfoSize(PCHAR(PGNM), Ret);
     GetMem(pData, Size);
     try
       Assert(GetFileVersionInfo(PCHAR(PGNM), 0, Size, pData));
       Assert(VerQueryValue(pData,
        PCHAR('¥StringFileInfo¥041103A4¥FileVersion'), pInfo,
        SizeFileInfo));
       Result := PCHAR(pInfo);
     finally
       FreeMem(pData);
     end;
    except
     Result := 'N/A'; //---- データ取得できない場合
    end;
  end;
  end.
```

福井 和彦 / 清水 孝将

株式会社ミガロ.

システム事業部システム3課 システム事業部システム2課

Delphi/400:Webからの PDF出力

Web アプリケーションで需要の高い PDF でアウトプットを実現する方法を紹介する。

●はじめに ●Webアプリケーションからの帳票出力 ●処理の流れ ●今回のポイント ●開発手順 ●セキリュティ管理 ●最後に



略歴 福井 和彦 1972 年 3 月 20 日生まれ 1994 年大阪電気通信大学工学部卒 2001 年 04 月株式会社ミガロ.入社 2001 年 04 月システム事業部配属

現在の仕事内容

主に Delphi/400 を使用した受託 開発で、要件確認から納品・フォロー に至るまでのシステム開発全般に携 わる。また、Delphi/400 の導入支 援やセミナーの講師なども担当。



略歴 清水 孝将 1983年10月4日生まれ 2008年甲南大学文学部卒 2008年株式会社ミガロ.入社 2008年04月システム事業部配属

現在の仕事内容

入社3年目でDelphi/400やJC/ 400の開発業務を担当。Webに関 する知識や技術を身につけ、Web アプリケーションのスペシャリスト を目指している。

はじめに

Delphi/400 の「VCL for the Web」 を使用することで、クライアント / サー バー型の開発と同様の手法で Web アプ リケーションの開発ができるため、特に 照会系の画面開発については敷居が低く なったように思う。

本稿では、さらに Delphi/400 を使用 した Web アプリケーション開発のバリ エーションを広げていただけるよう、 PDF 出力機能の実装について紹介して いこう。

Webアプリケーション からの帳票出力

Web アプリケーションは Web サー バー上で実行されるため、印刷処理を行 おうとしても、クライアントに接続され たプリンタから出力することができな い。

また、HTML 文書はクライアントの

OS やブラウザ、出力用紙サイズといっ たさまざまな要因によって、レイアウト が変わることがある。

このような要因から、Web アプリケー ションは帳票出力が得意ではなく、実装方 法に悩まれている方も少なくないだろう。

そこで今回、Web アプリケーションで PDF 形式の帳票を作成し、その結果を ブラウザ上で確認できる仕組みを紹介す る。この手法を見ていただくことで、 Web アプリケーションへ帳票出力機能を 実装する際の参考になれば幸いである。

PDF出力機能の構成

Web アプリケーションから PDF 出 力を行うための構成は、図の通りである。 【図 1】

●ISAPI アプリケーション

Delphi/400 の「VCL for the Web」 で作成する、一般的なWebアプリケー ション。

(ISAPI: Internet Server Application Program Interface) ● CGI アプリケーション

Delphi/400 の「WebBroker」で作成 する、一般的なWebアプリケーション。 (CGI: Common Gateway Interface)

処理の流れ

図を参照しながら、PDF 出力構成の 流れを順番に見ていこう。【図 1】

- ① ISAPI アプリケーションは、ブラウザ から PDF 出力のリクエストを受ける。
- リクエストを受け付けるための指示 画面は、Delphi/400の VCL for the Web で開発する。
- ②リクエストを受けた ISAPI アプリ ケーションは、さらに CGI アプリケー ションに対して PDF 出力のリクエス トを行う。

③リクエストを受けた CGI アプリケー ションは、Delphi/400 の機能を使用



して IBM i より必要なデータを取得 し、PDF 形式の帳票を出力する。

- CGI アプリケーションは、Delphi/400
 の WebBroker で開発する。
- PDF 出力を行うために「PowerPDF」
 を使用している (※)。

④CGI アプリケーションは、出力した
 PDF 形式の帳票をブラウザ上で参照
 できるようブラウザ側へ結果を返す。

※ PowerPDF については、本誌「Migaro.Techni cal Report 2010」に、帳票出力に関する論文が記 載されている。そちらも参考にしていただきたい。

今回のポイント

今回の仕組みのポイントは、次の2点 である。

- VCL for the Web での開発は画面周 りに特化させ、PDF 出力の部分に関 しては CGI アプリケーション側で実 装し、その CGI アプリケーションを 呼び出すようにしたこと。
- PDF 出力の部分を、Delphi/400 を 使用して、CGI アプリケーションと して開発したこと。

このような構成にすることで、VCL for the Webの開発では、PDF出力の 仕組みを気にすることなく、画面の開発 を行うことができる。

そして、CGI アプリケーションを Delphi/400 で開発することで、CGI ア プリケーションとして Web アプリケー ションの機能を持つことができ、さらに クライアント / サーバー型の開発手法に よる機能を実装した CGI アプリケー ションを構築することができる。

従って、Delphi/400 で IBM i へ接続 し、IBM i より取得したデータをもと に PowerPDF を使用して PDF 出力を 行うといった一連の流れには、クライア ント / サーバー型の開発手法をそのまま 利用でき、これまで培ってきたノウハウ を活用することができる。

開発手順

具体的な開発手順について紹介してい

く。開発手順の流れは、以下の通りである。

- (1) CGI 実行形式アプリケーションの作成
- (2) CGI アプリケーションのActionsの実装
- CGI アプリケーションと PowerPDF との連携
- (4) ISAPI アプリケーションからの実行

(1) CGI 実行形式アプリケーションの 作成

まずは、実際にどのようにして CGI アプリケーションを作成するのかを紹介 したい。

Delphi では、WebBroker というフ レームワークを利用して、CGI アプリ ケーションを容易に作成することが可能 である。

手順としては $[ファイル] \rightarrow [その他]$ → [Delphi プロジェクト / WebBroker]→ [Web サーバーアプリケーション]を選択する。[Web アプリケーションの新規作成」の Window が表示されるの で、[CGI 実行形式]を選択すると、開 発画面に遷移する。

開発画面には、Unit.pas(TWebModu le)が自動で生成されている。このユニッ トに、CGI アプリケーションの処理を 記述していくこととなる。【図 2】

(2) CGI アプリケーションのActions の実装

CGI アプリケーションの URL
 図の URL は、CGI アプリケーション

の URL である。

URL には、Pathinfo 部と Query 部と いうものが存在する。Pathinfo 部は CGI アプリケーションの処理の分岐に使 われ、Query 部は CGI アプリケーショ ンへ渡すパラメータの役割を持ってい る。【図 3】

WebBroker で は、TWebModule が 持つ Actions を設定することによって、 Pathinfo ごとの処理を容易に分岐させ ることができる。

TWebModule が持つ Actions は、ク ライアント / サーバー型アプリケーショ ンの Action と非常によく似た仕組みに なっている。Actions は、オブジェクト インスペクタから、Actions のプロパ ティエディターで追加することができ る。この Actions の OnAction イベン トにコーディングすることで、CGIア プリケーションの処理を実装することが できる。【図 4】

ここからは、実際のコーディングの説 明を行う。

前述した通り、Web アプリケーショ ンとは、ブラウザ(ユーザー)がWebサー バーに対してリクエスト(要求)を送り、 Web サーバーがレスポンス(結果)を ブラウザ(ユーザー)に返すということ が大きな仕組みとなっている。

CGI アプリケーションがユーザー(ブ ラウザ)からのリクエストを受け取る時 や、それとは逆にユーザー(ブラウザ) にレスポンスを返す場合のどちらも、 TWebModule.Request プロパティで行 う。アプリケーションに渡された、Pat hinfo や Query の 値 は すべて Request プロパティ内で保持されている。

(3)CGIアプリケーションとPowerPDF との連携

では、帳票出力処理の実装の説明に入りたい。

最初に、PowerPDF で作成したテンプ レート Form を、CGI アプリケーション のプロジェクトに追加する。続いて、こ の追加した Form を、CGI アプリケーショ ン内部で生成し、帳票出力に利用する。

なお、PowerPDFのForm自体は、 クライアント/サーバー形式で作成する 場合と同じものを使用することができる ため、すでに作成済みのテンプレートが ある場合などはそのまま流用することも 可能である。

【コーディング処理の流れ】

 ①テンプレートファイルを生成
 ②リクエストからの情報 Query を取得
 ③パラメータをもとに SQL をデータ ベースに発行
 ④該当レコードをテンプレートファイ ルに転送
 ⑤PDF ファイルに出力(ファイルに保存)
 ⑥保存されたファイルの読込
 ⑦レスポンスに対して、読み込んだファ イルと表示アプリケーションの指定を 返却

この流れで連携処理を実装していく。 流れの中で、WebBroker 特有のクラス やメソッドを利用する部分は①と⑦の部

ect1.exe/getpdf?code=0 図4-2
図4-2 WebModule1.Actions の編集
WebModule1.Actions の編集
🛞 WebModule1. Actions の編集
22 4 4
Name PathI Enabled Default Producer
 * * ※ ※

分のみで、他の部分はクライアント / サーバー形式のコーディングと全く同じ である。

詳しい記述は、ソースを見ていただき たい。【ソース1】 このように作成された CGI アプリ ケーションを実行すると、画面のように、

アDF が表示される。【図 5】

● CGI アプリケーションへのアクセス 作成した CGI アプリケーションにブ ラウザからアクセスする際は、以下のよ うな URL でアクセス可能である。

URL の構成については、前述の図を 参照してほしい。【ソース 2】

http:// サーバー名 / スクリプト名 /Path info?Query= * * * * *

(4) ISAPI アプリケーションからの 実行

ISAPI アプリケーションから呼び出 す場合は、NewWindow メソッドを使 用して CGI アプリケーションを起動す る。【図7】

Pathinfo の値や Query の値を呼出時 に変えることによって、CGI アプリケー ションの処理を分岐させることができる。 ソース例に示すように、CGI アプリ ケーションの URL で「ESTCODE= 0011」の「0011」の部分を見積番号と した場合、CGI アプリケーションを呼 び出す際には、「0011」の部分に ISAPI アプリケーションで指定された見積番号 を指定することで、対象の見積書を出力 することができる。

セキュリティ管理

PDFやExcelファイルなどをそのま まブラウザ上で表示するためには、 Webサーバーの公開された範囲にファ イルを配置しなくてはいけない。だが、 実際に機密情報や個人情報などが記載さ れているようなファイルを、Webサー バー上に置くことは不可能である。

しかし、CGI アプリケーションを介 した場合は、Web サーバー上で公開さ れていないファイルに対してアクセスす ることが可能となる。CGI 内部に、ユー ザー認証の処理を組み込めば、限定され たユーザーに対して PDF 表示を実現す ることができる。

最後に

Web アプリケーションを構築する上 で悩みの種であった帳票出力を実装する 方法について、その解決策の1つとして、 今回「ISAPI アプリケーション」+「CGI アプリケーション」の構成で PDF 出力 結果をブラウザへ表示するという仕組み を紹介してきた。

ここまでご覧いただいて、Delphi/400 で CGI ア プリケーションを開発し、 VCL for the Web で開発した ISAPI ア プリケーションと連携させることで、 PDF の出力が簡単に行えることがお分 かりいただけたであろう。

今回紹介した例は PDF 出力であった が、CGI アプリケーションの作り方次 第では、Excel 形式への出力や CSV 形 式への出力を行うことも可能であるの で、PDF 出力以外の方法についても、 ぜひともチャレンジしていただきたい。 今回の内容が、Web アプリケーション からの帳票出力方法の参考になれば幸い である。


吉原 泰介

株式会社ミガロ.

RAD事業部 技術支援課 顧客サポート

Delphi/400でFlash動画の実装 —YouTubeプレイヤーの作成

Flash の機能を、ActiveX を利用して Delphi/400 アプリケーションへ組み込む方法を紹介する。

●FlashとActiveXコントロール ●ActiveXコントロールの取り込み ●YouTubeプレイヤーの実装 ●まとめ



1978 年 3 月 26 日生まれ 2001 年 龍谷大学法学部卒 2005 年 07 月株式会社ミガロ . 入社 2005 年 07 月システム事業部配属 2007 年 04 月 RAD 事業部配属

現在の仕事内容 Delphi/400 と JC/400 の 製品 試 験、および月 100 件に及ぶ問い合 わせやサポート、セミナー講師など を担当している。

1. Flashと ActiveXコントロール

近年の Web 上では、画像や動画、音 楽がコンテンツとして組み込まれること が一般的になっている。【図 1】

こういった画像、動画、音楽などのコ ンテンツをWebで扱うための技術とし て、広く使われるようになったのが 「Flash」である。私達は普段Webブラ ウザで何気なくWebページを閲覧する 際に、動画などを表示/再生するのに、 実はこのFlashを利用している。実際、 ネットワークに接続しているほとんどの PCには、Flashを利用するための環境 が導入されている。

ActiveXコントロール

この Flash はどのような形で私達の PC に提供されているかというと、実は 「ActiveX コントロール」という技術を 利用している。

ActiveX は Microsoft 社が提供する インターネット関連技術の総称である が、最も代表的なのが、Web ブラウザ 上で動的な Web ページを実現する ActiveX コントロールである。ActiveX コントロールを ActiveX と呼ぶ場合も 多い。

ActiveX コントロールはネットワー クを通じて Web サーバーからダウン ロードされ、Web ブラウザに機能を追 加する形で使用される。つまり、Web ブラウザ上で本来提供されていない機能 を連携して使用しているわけである。

そして、この ActiveX コントロール で Flash を利用することによって、昔 は Web で扱うことができなかった動画 や音楽を、Web ブラウザ上で見たり聞 いたりすることができるようになった。 他にも身近なものとして、Windows アッ プデート機能の更新チェックやインス トールなども、ActiveX コントロール によって Web ブラウザから提供されて いる。

ActiveX コントロールの機能をもう 少し分かりやすく説明していく。 例えば、Delphi/400 のアプリケーショ ンで Excel の出力を扱う場合、COM (コ ンポーネント・オブジェクト・モデル) コンロールという技術が使用されてい る。COM コントロールは、Delphi/400 のアプリケーションの中で、Excel とい う別アプリケーションの機能を利用する ための連携を提供している。

では、ActiveX コントロールと COM コントロールと何が違うかというと、実 はほとんど同じものだと考えてよい。違 いとしては、ActiveX コントロールは 「Web で利用する技術」という点だけで ある。

Delphi/400からFlashを扱う

以上のように考えてくると、ActiveX コントロールを利用して、Delphi/400 から Flash を扱えそうだというイメー ジができるのではないだろうか。

本稿では、インターネット動画共有 サービスで有名な YouTube の動画再生 アプリケーションを題材として、 Delphi/400 から Flash を利用する方法



を紹介したい。

YouTube http://www.youtube.com/

次章では、最初のステップとして、 Flashを扱う機能を Delphi/400 に取り 込む方法から説明していく。

2. ActiveXコントロール の取り込み

Delphi/400 で Flash 動画を扱うため には、Flash を再生するためのインター フェース情報が必要となる。このイン ターフェース情報は「TLB(タイプラ イブラリ)」と呼ばれる。

Delphi/400には、ActiveX コントロー ルからインターフェースを抽出して、D elphi/400のプログラムソースを自動で 作成してくれる機能が提供されている。 この機能を利用する。

TLBの自動作成

開発画面のメニューから [コンポーネ ント¦コンポーネントのインポート]を 実行すると、ウィザード画面が表示され る。【図 2】

ここで「ActiveX コントロールの取 り込み」を選択し、「次へ」を実行する。 次の画面では、その PC にインストー ルされている ActiveX コントロールが 一覧で表示されるので、Flash を再生す るための「Shockwave Flash」という ActiveX コントロールを選択して「次 へ」を実行する。【図 3】

次の画面では、この ActiveX コント ロールの取り込み先を指定する。「パレッ トページ名」は任意の名前、「ユニット ディレクトリ名」には自動生成されるプ ログラムソースの保存先ディレクトリを 設定して、「次へ」を実行する。【図4】

最後の画面では「ユニットの作成」を 選択して「完了」を実行する。【図 5】

完了すると、保存先に指定したディレ クトリに、ShockwaveFlashObjects_T LB.pas というソースファイルが自動で 作成されている。【図 6】

これは、ShockwaveFlashのTLB(タ イプライブラリ)情報のプログラムソー スである。このインターフェース情報の ソースは1000ステップ以上のプログラ ムからなるが、これらをコーディングす ることなくウィザードで自動作成できる ので、TLBの自動作成再生機能は非常 に便利な機能といえる。

もちろんこの ShockwaveFlash に限 らず、ActiveX で提供される機能は同 様に Delphi/400 に取り込むことが可能 である。

なお、ソースを見ると分かる通り、ソー スの前半にはコメントで警告が書かれて いる。これは自動生成で作成されたプロ グラムソースなので、独自にソースを変 更しても、次に自動生成した時には上書 きで消えてしまうという注意書きである。

Flash再生のためのクラス

ここで自動生成されたプログラムソー スの詳細を完全に理解するのは難しいか もしれないが、中身を見てみると、ソー スの①にあるように TShockwaveFlash というクラスが定義されている。【図7-①】

またソースの②にあるように、このク ラスには、Play や Stop といった Flash 再生を操作するためのメソッドも用意さ れていることが分かる。【図 7- ②】

つまり、インターフェースだけでなく Flash を再生するためのクラスも自動で 作成されているので、このクラスを利用 するだけで、Flash を Delphi/400 から 操作することができるのである。

次章では、実際にこのクラスを使用し て、YouTube の動画再生プレイヤーを 作成する。

3. YouTubeプレイヤーの 実装

実際に、YouTubeの動画を再生する アプリケーションを作成してみよう。

(1)はじめに新規プロジェクトを作成し、
 先に ActiveX コントロールから作成した自動生成ユニットの ShockwaveFlas
 hObjects_TLB.pas を組み込む。

これは[プロジェクト¦プロジェクト に追加]から保存している ShockwaveF lashObjects_TLB.pasを指定する。【図8】

(2) これでプロジェクトに組み込まれたので、実際に Flash を使いたいユニッ

トで、interface 部の uses 節に Shock waveFlashObjects_TLBを追記する。 【ソース1】

(3) ShockwaveFlashObjects_TLB で
 定義されたクラスを利用するために、変
 数を定義する。【ソース 2】

(4) 次に、アプリケーション画面を作成 する。

画面には、YouTube で再生する動画 の ID を入力する TEdit (①) と、再生 用の TBitbtn (②) と停止の TBitbtn (③)、そして Flash を再生するための TPanel (④) を配置する。【図 9】

(5) Flash の再生部分には TPanel を使 用している。

ShockwaveFlashObjects_TLB で 用 意されているのはクラスであって、コン ポーネントではないからである。今回は 画面で描画する領域として、TPanel を 利用する仕組みにしている。

この TPanel に Flash のクラスの描画 を割り当てるために、画面起動時に フォームの Create イベントで、ソース に示すような処理を行う。【ソース 3】

(6) 画面終了時にはこれを破棄するため に、フォームのDestroyイベントで、ソー スに示すような処理を行う。【ソース 4】

(7) ここから、Flash を再生 / 停止する ボタンの処理をプログラミングすること になる。

その前に、自動生成された TShock waveFlash のクラスで使用できる、主 な機能を記載しておく。

【TShockwaveFlashクラスで用意される 主な機能】

- ・Playメソッド:Flash Movieを再生する メソッド
- ・Stopメソッド:Flash Movieを停止する メソッド
- Movie プロパティ:再生する Flash
 Movie を指定するプロパティ(※)

※ Flash のファイルをここで指定する。URL でも 指定できる。今回の YouTube 動画の場合は、URL を指定する。 図6 ージ 🐻 Unit3 😨 ShockwaveFlashObjects_TLB 🐻 Project3 unit ShockwaveFlashObjects_TLB; 警告 ------このファイルで宣言されている型はタイプライブラリから読み取られたデータから 生成されています。このタイプライブラリが明示的あるいは(このタイプライブラ リを参照しているほかのタイプライブラリ経由で)間接的に再インボートされた り、タイプライブライブラリの編集中にタイプライブラリエディタの[更新]コマ とするですたもます。 レナママートもます。 はすべて失われます ********** \$Rev: 17244 \$ 下に説明されたタイプ ライブラリから 2010/08/21 17:07:36 に生成されたファイル。 XT: 図7 🕴 🐻 Unit3 🖉 ShockwaveFlashObjects_TLB 🐻 Project3 (I) TShockwaveFlash = **class**(TOleControl) NITAGES FOnReadyStateChange: TShockwaveFlashOnReadyStateChange; FOnProgress: TShockwaveFlashOnProgress; FOnFSCommand: TShockwaveFlashFSCommand; FOnFlashCall: TShockwaveFlashFlashCall; FIntf: IShockwaveFlash; function GetControlInterface: IShockwaveFlash; protected procedure CreateControl; procedure InitControlData; override; function Get_InlineData: IUnknown; procedure Set_InlineData(const ppIUnknown: IUnknown); public procedure SetZoomRect(left: Integer; top: Integer; right: Integer; bottom: Intege procedure Zoom(factor: SYSINT); procedure Fan(A. Integer, y. procedure Play; Integer, mode. orona procedure Stop; Procedure Duck procedure Forward; procedure Rewind; procedure StopPlay; procedure GotoFrame(FrameNum: Integer); function CurrentFrame: Integer; function IsPlaying: WordBool; function PercentLoaded: Integer; function PrameLoaded(FrameNum: Integer): WordBool; function FlashVersion: Integer; procedure LoadMovie(layer: SYSINT; const url: WideString); procedure TGotoFrame(const target: WideString; FrameNum: Integer); procedure TGotoLabel(const target: WideString): Integer; function TCurrentFrame(const target: WideString): Integer; function TCurrentLabel(const target: WideString): WideString; procedure TPlay(const target: WideString); procedure TPlay(const target: WideString); procedure TStopPlay(const target: WideString); procedure SetVariable(const name: WideString): WideString; function GetVariable(const name: WideString): WideString; procedure TSetProperty(const target: WideString): WideString; procedure TSetProperty(const target: WideString; property_: SYSINT; const value: function TGetProperty(const target: WideString; property_: SYSINT): WideString; function PercentLoaded: Integer;

76

(8) 再生ボタンが押された処理では、 Movie プロパティに YouTube の URL とあわせて画面で入力された動画の ID をセットして、Play メソッドで Flash の再生を行う。【ソース 5】

YouTubeのURL http://www.youtube.com/v/

指定する ID は、YouTube で公開さ れている動画に設定されている ID であ る。例えば、次の URL は Google ストリー トビューの YouTube 動画の URL であ り、「v/」以降の文字列が ID である。

http://www.youtube.com/v/XD_4WS wkWEU

なお、YouTube 動画上を右クリック してポップアップされるメニューから 「Copy embed html」を選択し、エディ タ等に貼り付けるなどして、ID を含む 情報を確認することもできる。

(9) 最後に停止ボタンを押された時の処 理では、単純に Stop メソッドを実行す るだけである。【ソース 6】

(10) これだけで、簡単な YouTube プ レイヤーの完成である。早速実行して Flash の再生を試してみよう。【図 10】

なお(9)に関連して、ShockwaveFlash 自体に「Stop メソッドを実行しても停 止しない場合がある」という動作問題が ある。

Stop メソッドの動作を確実に解決す る場合には、本レポートの最後に補足掲 載している、ソースのクラス修正を加え ることで、Delphi/400 側で対応するこ とができるので参考にしてほしい。【ソー ス7】【ソース8】

4. まとめ

この YouTube 動画再生アプリケー ションの開発は非常にシンプルで、簡単 に Flash の再生操作機能を実現するこ とができた。

もちろん Flash の再生プログラム自 体を自身ですべて開発しようと思えば、 複雑で高度なプログラムが必要だが、 Delphi/400のActiveXコントロールの 自動取り込み機能によって、それらの開 発はほとんど必要としない。

本稿のアプリケーションは YouTube の動画を再生する機能だけであるが、 Movie プロパティに Flash ファイルを 設定すれば、通常の Flash も再生する ことができる。つまり、Webページ同様、 Delphi/400 のアプリケーションも、画 面に Flash 動画などを組み込んだ見た 目のよい画面を作ることが可能なのであ る。【図 11】

また、最近では Flash を利用したプロ グラムで、「Flex」と呼ばれるプログラ ム(ActionScript)を組み込んだ Flash もある。これらと連携したアプリケー ションの開発も面白いだろう。【図 12】

この Flash の連携だけでも、Delphi/ 400 アプリケーションには、さまざまな 開発の可能性が含まれているのが分か る。他の ActiveX コントロールの利用 や今後の開発において、ヒントや参考に していただければ幸いである。

Μ



ジプロジェクトに追加	×
	▼ 4 Testの検索 P
整理 ▼ 新しいフォルダー	
 ☆ お気に入り ダウンロード デスクトップ 図 最近表示した場所 コイブコレ 	更新日時 cts_TLB.pas 2010/08/21 17:07 プレビュー を利用でき ません。
■ コンピューター	•
ファイル名(N): ShockwaveFlashObjects_TLB.pa	as Delphi ユニット (*.pas) 開く(Q) #
)	図10
I YouTubeHaver① ② ③ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	▲ YouTubeReyer VID.D_4WSwkWEU 年生 ● 存止 日本 日本 日本 日本 日本 日本 日本 日本 日本 日本
11	X12
€;taret	Formi ✓ Flash再生 ◎ 停止 Sales Marketing Yearly Revenue by Region To Do To Do To Do
RPG/GOBOLで簡単に開発 I IBM I WEB 化ツールの決定版 JC /400	Image Chart Image Chart

	_
 ソース1 Uses節	
 unit Unit1;	
 interface	
uses	
 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, ExtCtrls, Buttons, ShockwaveFlashObjects_TLB;	
 ソース2 宣言部	
 private { Private 宣言 } //表示用Flash fFlashplayer : TShockwaveFlash;	
 ソース3 初期処理(表示Flash生成)	
 procedure TEorm1.EormCreate(Sender: TObject):	
 begin //Flashをクラスから生成	
fFlashplayer := TShockwaveFlash.Create(Self); //最大化	
//取べる fFlashplayer.Align := alClient; //志元田FlashplayerのPanalの上に認定(組工関係)	
fFlashplayer.Parent := Panel1;	
 ena;	
ソース4 終了処理(表示Flash破棄)	
 procedure TForm1.FormDestroy(Sender: TObject);	
 begin	
 fFlashplayer.Free; end;	
 ソース5 再生ボタンクリック	
 procedure TForm1.btnPlayClick(Sender: TObject); begin	
- //入力のVIDを指定してMovieに設定 fElashplayer Movie := 'http://www.youtube.com/y/' + Edit1 tevt:	
//Flashを実行 fElashplayer Play:	
 end;	

```
ソース6 停止ボタンクリック
  procedure TForm1.btnStopClick(Sender: TObject);
  begin
    //Flashを停止
    fFlashplayer.Stop;
  end;
ソース7 修正クラス宣言部
  //宣言部
  type
    //Stop、Playメソッドの動作を補正するためのクラス
    TFlash = class(TshockwaveFlash)
    private
      FPlaying : Boolean;
    protected
      procedure WndProc(var vMsg: TMessage); override;
    public
      procedure Play; reintroduce;
procedure Stop; reintroduce;
      property Playing: Boolean read FPlaying;
    end;
ソース8 修正クラス実装部
     //実装部
      { TFlashPlayer }
     //Playメソッド
     procedure TFlash.Play;
     begin
       FPlaying := True;
       inherited play;
     end;
     //Stopメソッド
     procedure TFlash.Stop;
     begin
       FPlaying := False:
       inherited stop;
     end;
     //メッセージ procedure TFlash.WndProc(var vMsg: TMessage);
     begin
       case vMsg.Msg of
         WM_TIMER:
if (FPlaying) then
           begin
             inherited;
           end
           else
           begin
             vMsg.Result := 0;
           end;
         else
           inherited;
       end;
     end;
```



MIGARO. TECHNICAL REPORT

Migaro.Technical Report No.3 2010年秋 ミガロ.テクニカルレポート

ミガロ・テクニカルレポート
2010 年 11 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 Report2010
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。

MIGARO	
TECHNICAL REPORT ミガロ.テクニカルレポート	
株式会社 ミガロ . http://www.migaro.co.jp/	
〒556-0017 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 東京営業所 〒106-0041	
東京御港区麻布官1-4-3 エグゼクティブタワー麻布台 11F TEL:03(5573)8601 FAX:03(5573)8602	