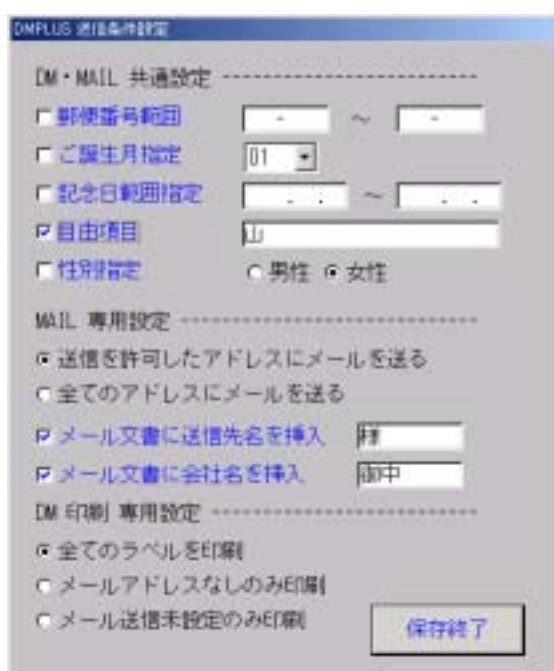


何が IT 革命なのか？この疑問に昨年から回答が出ないままに既に二月になってしまった。確かに子供の学校にはコンピュータが年々数を増やし、関連の授業も増えている。昨日英検の試験を受けてきた長男は、学校から戻るとインターネットのホームページで答えを確認。確かに情報の流れは早い。駅で別れ際に友人同士と思われる二人が後はメールで！何て会話をしている。そうか、二人は家に帰ってからメールで話をするのか？確かに電話よりもメールは相手の時間を束縛しないという優しさがある。幾つかのメーリングリストに参加してみると面白い傾向を見ることが出来る。例えばお店で販売を担当しているような会社に勤めている方々が多いメーリングリストの場合、昼休み～14 時頃に突然メールが増える。帰宅や出社、残業時間の問題もあって、17 時～20 時が盛況で、24 時ぐらいは止まるが、また午前 2 時頃は増える。曜日によって多少傾向は違うが、ほぼ同じ状態で推移しているようだ。確かに多くの人達が距離と時間の制約に縛られず、情報や意見、心を交換している。掲示板を覗くと様々な商品を購入した本人からの生の声を確かめる事が出来る。特に携帯電話については様々な意見が飛び交っているが、有益なものもあるが、誤解を招くような話も多い。人気のあるサイトを調べて見ると、社会現象に関するもの、タレントやスポーツ選手、ホームページ自体を作るための素材や技術（CGI や Perl）についてのもの、占い、携帯電話、グルメ、オークション、株式、何か当る、というように検索エンジンのトップページにあるカテゴリーは大抵が人気サイトである。インターネットの周辺で仕事をしている人々は自分が勝ち組みに入ることを目論んでいるのは当然だが、勝敗自体が実に曖昧とも言える。ユニークな技術や考え方でスタートしたサービスがある日一気にブレークし、突然大金を掴むが、それを維持する事は大変な努力が必要だ。成功に目を付けた資本化がそっくり買い取ってしまえば、創設者は資産家となるが、それを存続させてこそ成功と見るか？ただ資産家になれば成功か？見識が分かれる所だろう。IT 革命は過去のヒエラルキーを取り去り、人間社会を情報で埋め尽くそうとしているかに見える。ただ過去のヒエラルキーが無意味かと言えば決してそうではない。人間の行動原理は情報だけではなく、感情や情緒、哲学や倫理がそこに大きく貢献する。感情や情緒を適切にコントロールする能力を身に付け、哲学や倫理の学習にはどうしてもヒエラルキーが必要になる。そこには情報を超えた、愛情とか信頼関係が教師と生徒の間に必要になるからだ。最近になって多少はインターネット社会と人間の関わりを本質から見直そうという論文を新聞等で目にするようになったが、相変わらず技術面でのセキュリティの話が多く、人間そのものへのセキュリティについてはあまり語られていない。数日前にインターネットで洗脳されるという話が報道されていた。便利さと危険はいつもセットである。

「DMPLUS を拡張」

二つ前の号でちょっとお知らせしたカスタマーバーコードの付いたラベルを発行するシェアウェア「DMPLUS」を拡張し、メール機能を付けたのでお知らせしておこう。メールと言ってもただメールが指定した相手に送れるだけではなく、データベースに登録された相手先全てにメールを送ったり指定した条件で相手先を特定してメールを送ることが出来る

というものである。実は Arago for Windows には標準でメールの送受信を可能とする機能が付属している。データベースソフトでこの種類の機能を利用する目的は毎日の入力データを本社に送信するような機能への応用が主なる目的だとは思うが、本格的なデータベースを使えるメールソフトというのは実は非常に有り難いという結論に至り、DMPLUS をバージョンアップした。メールをまとめて送る機能は通常のメールソフトにもあるが、100件未満なら実用性はあるかもしれないが、それ以上の場合や、今回はこの条件の相手だけ送りたいとなると使えない。またメーリングリストという方法も、都度相手を選ばないとか、相手先の名前がメール内に入らないという問題があり、目的は限られてしまう。画面 1 は DMPLUS の条件設定画面である。



画面 1 送信条件の指定

これらの項目設定はメールを送る相手の住所範囲、誕生月、年齢、性別、そして自由項目では趣味等を複数指定できるようにした。ここでは「山」と指定しているが、これは画面 2 の顧客編集画面の自由項目に関連する。例えば自由設定項目に「山」という文字が何処かに含まれているデータは送信の対象になる。またメール専用の設定欄ではメール文書に送信先の名称や会社名の挿入をするしない?を選ぶ事が出来る。メールの内容で様ではなくて「田中さん」という呼びかけにしたいという要求にも応えられるようにした。DMPLUS からはパソコンへのメールの配信は勿論の事、i-mode 等の携帯電話へもメールを送信できる。実はこの数ヶ月の間、長男に携帯電話を持たせてその利用実態や友人達の利用状況を探っていたのだが、アルバイトをしていない学生の場合、利用料金が 5000 円までは親が負担、それ以上になった場合は 5000 円からオーバーした金額を自分の小遣いで負担というのが大半である事が分かった。彼らの多くが会話よりもメールを中心に利用していて、WEB のサイトは以外に利用しておらず、ニュースや天気予報のメール配信は利用しているとい

DMPLUS ADDRESS EDIT

顧客氏名 : TAMADA (必須項目)

郵便番号 : 730-0003

住所1 : 愛媛県 松山市

住所2 :

住所3 : 〒7300003

会社名称 :

電子メール :

誕生日 :

記念日 :

電話番号 :

自由項目 : スキーサッカー山カメラ音楽

性別 : 男性 女性

送信指定 : 送信する

編集終了

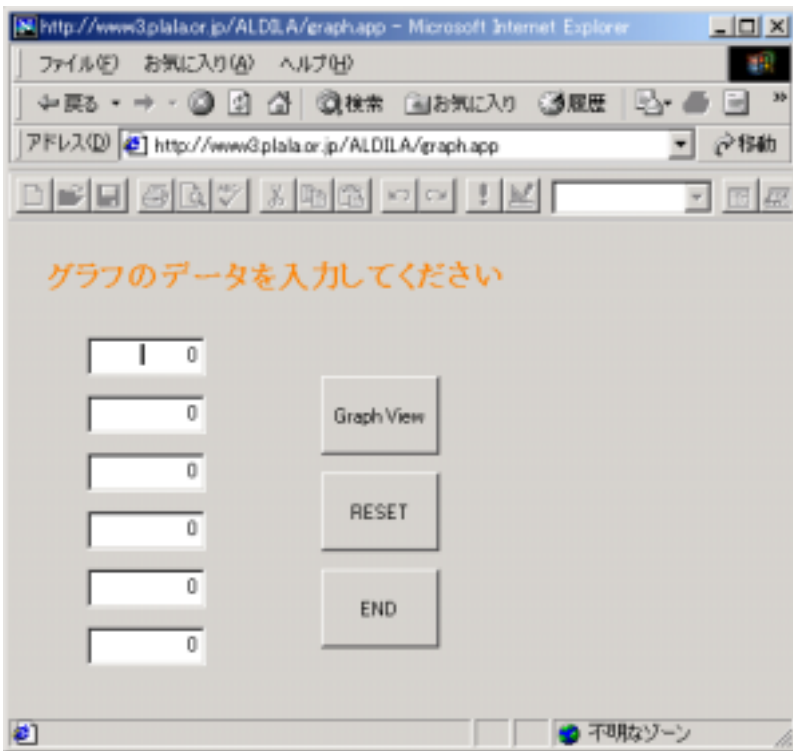
画面 2 顧客メンテナンス

うのが大半であった。例えば音声の電話は彼らにはかなり負担である。メールなら 1 回 3 円とかいう料金設定だったり、i-mode ならパケットでトータルの文字数だからあまり料金を気にしなくて済むわけだ。実は i-mode とか EZ-WEB に向けた WEB ページを作りながら、本当に見てくれるのだろうか？という疑問に襲われ、そう言えばメールだったら確実に受けて貰えて、それが送られた人にとって幸運を運ぶ情報なら（ここが難しいのだが）送り方を工夫できれば（誕生日半額とか）と考えて DMPLUS は生まれた。DMPLUS は Vector にシェアウェア（制限付きで一部フリーウェア）として登録されている。

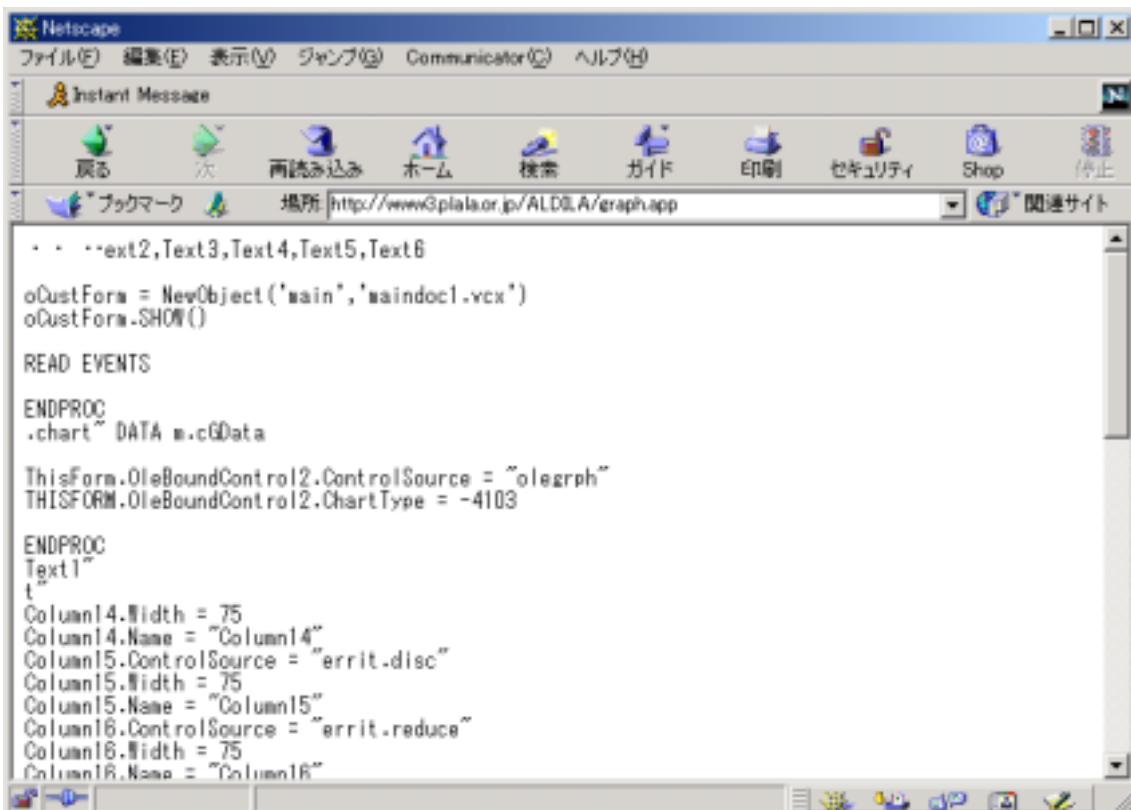
場所は (<http://www.vector.co.jp/vpack/filearea/win/writing/address/index.html>) である。

「VisualFoxPro でグラフを作る」

先月は VisualFoxPro アプリが Internet Explorer で動くまでの簡単な作成方法をお見せしたが、今回は実際に動くアプリケーションを WEB 上でお見せしながら（これは実際に見て頂きたい）簡単なグラフを作成してみよう。最初に画面 3 を見て頂きたい。ここではグラフに入力するデータをユーザが入力する画面が表示されているが、これは筆者のホームページにある VisualFoxPro のテストページに用意されている。このグラフを動かすためには別に FORBS をインストールしておく必要がある。FORBS は「[Object Oriented XBASE Forum](#)」から入手可能である。（<http://www2u.biglobe.ne.jp/~objxbase/forbport.htm>）FORBS はマイクロソフトが確認し公認されている。問題となったのは VisualFoxPro に含まれている DLL 等のドライバーで一般への配布を禁止しているものが結構あり、FORBS の作者である市岡氏もその理解に苦勞されたと言う。さて FORBS がインストールされると VisualFoxPro のアプリケーションである画面 3 はクライアントにインストールされたアプリケーション並みのスピードで動く。早い話が VisualFoxPro の WEB アプリケーションは FORBS を利用したインタープリターなのである。だからコードの一部は Netscape を使うと見えてしまう。だが全てが見える訳ではなく、部分的にコードが露出するようになる。その様子は画面 4 で確認出来る。

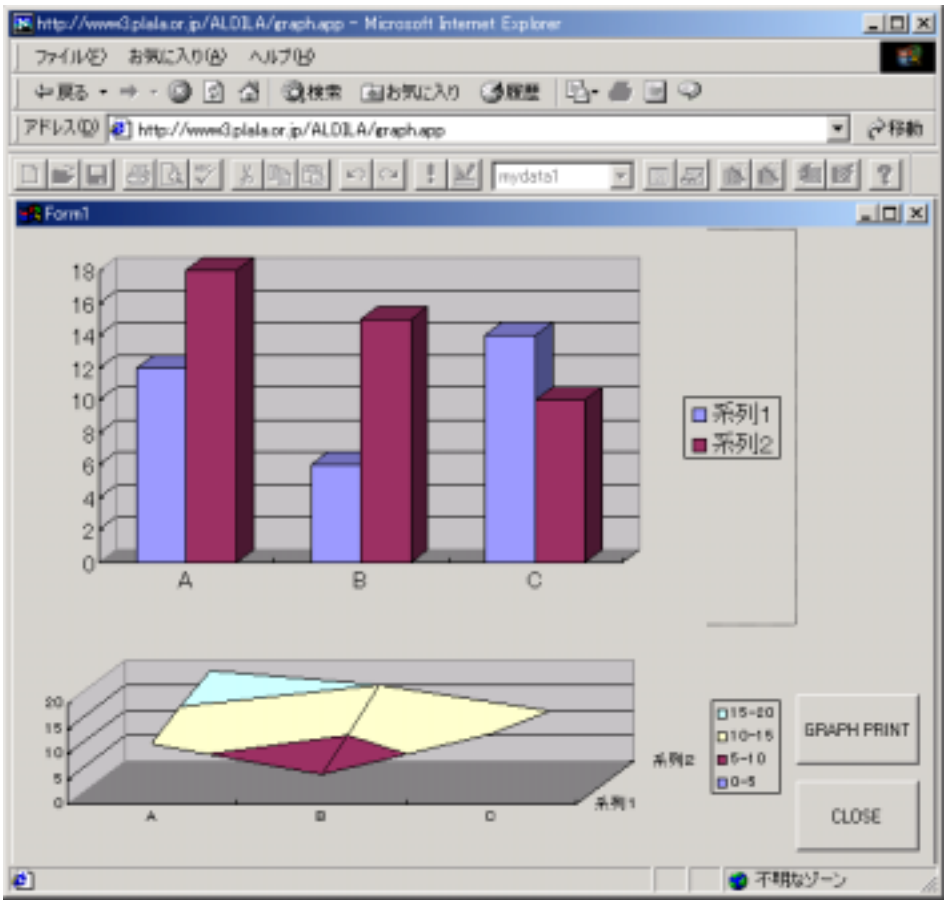


画面 3 入力画面

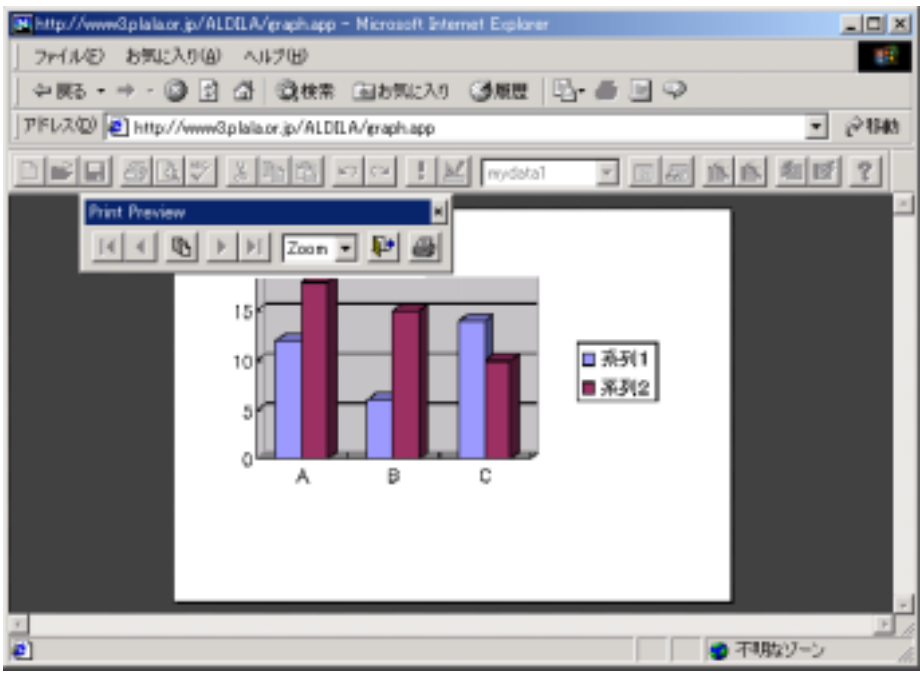


画面 4 Netscape ではこのようにコードが見えてしまう。(画面 3 と同じものだ)

では画面 3 にデータを入力してこのグラフプログラムがどのように動くか実際にお見せしよう。画面 3 に実際にデータを入力して「Graph View」を押すと画面 5 が表示される。



画面 5 表示されたグラフ



画面 6 グラフの印刷画面

そして更に画面 5 の「GRAPH PRINT」を押すと画面 6 が表示される。ここまでは実際に

筆者のサイトで動いている WEB アプリケーションである。グラフの印刷画面ではズーム機能が働くが、VisualFoxPro ではレポート画面には全て標準でこの機能が搭載される。何度か本誌でご説明してきた事なのでご記憶の方も多いとは思いますが、今回のこのグラフは画面でデータを入力し表示させているが、ユーザのマシンにあるデータを読み込んでグラフを表示させる事も可能である。むしろ VisualFoxPro にとってはその方が楽なので、例えば別の機器からユーザ側の PC にデータを取り込み、WEB 上で分析した結果を表示するような機能は非常に作り安い。分析を行う VisualFoxPro のアプリケーションを日々更新する事で、ユーザ側のデータから様々な判断を導き出すような事も可能になる訳だ。ちょっとその仕組みを以下に解説してみよう。

最初にプログラム 1 を参照して頂きたい。このコードは画面 3 の「Graph View」ボタンに定義されたコードである。「Graph View」ボタンを押すとこのコードが動くわけだが、プログラム 1 が行う作業はユーザの PC 上の C: ドライブに指定したディレクトリがあるか判断して、なければディレクトリを作成し、同様にデータベースとテーブルを作成している。そしてデータベースの作成後グラフを定義済みのチャイルドウィンドを表示する。

SET SAFE OFF

CLOSE DATABASES

CLEAR

IF FILE("C:¥VFP¥VFPGRPH.DBF")=.F. && (ディレクトリの有無を判断)

 RUN MD C:¥VFP

ENDIF

ThisForm.refresh

* Create mydata database in the current directory or folder

CREATE DATABASE C:¥VFP¥mydata1

CREATE TABLE C:¥VFP¥VFPGRPH ;

 (OLEGRPH g(4))

CREATE TABLE C:¥VFP¥VFPGRPH1 ;

 (OLEGRPH g(4))

CREATE TABLE C:¥VFP¥VFPGRPH2 ;

 (OLEGRPH g(4))

local oCCustForm

oCCustForm = NewObject('childwind','maindoc1.vcx')

oCCustForm.show() && (チャイルドウィンドの表示)

read events

プログラム 1

プログラム 1 から分かるようにその気になれば VisualFoxPro の WEB アプリからユーザのクライアントをコントロールする事すら可能である。だから VisualFoxPro の WEB アプリを使用する場合には必ず信頼関係のある提供者からのものである事を理解してからでなくてはならない。筆者のページでも具体的にユーザのマシン上に作成するデータベースの内容を記述している。これが困る場合には使用は避けていただきたい。とまあちょっと危険な事も出来てしまうのだが、サーバからユーザサイドの PC ヘデータを強制的に送りつける機能は VisualFoxPro にはない。あくまでもあるフォーマットのデータベースをユーザサイドの PC 上に作る機能を持っている訳だ。電子メールでデータベースを配信しておいて、VisualFoxPro のページに入ると、データベースの中身を加工して見せたり、プロモーションが流れたりという機能も考えられる。制限はクライアントの WEB ブラウザが Internet Explorer でなければならないという点以外は何もなく、Java には制限は見当たらないが、スピードは遥かに VisualFoxPro の方が速く、開発工数は Java の数分の一という所だろうか？その使い方の違いは、Java や PHP、Perl はサーバ側にデータベースを置いて不特定多数の相手に情報公開し、サービスを提供するものであるが、VisualFoxPro は特定の会員や社員または趣味とか興味を持つ相手に、特化した機能やサービスを提供するものである。クライアント側にデータベースが置かれるために、セキュリティには気を使う必用もあまりなく、同時に利用するユーザが幾ら増えてもレスポンスには全くと言って良いほどに影響しない。精々アプリケーションを自分のマシンにロードする時間ぐらいだろうが、ご紹介したグラフのアプリケーションのサイズはたったの 43K しかない。ちょっと自分のサイトの HTML ファイルサイズを調べた所、多くが 10K ~ 20K で画像を含めると 100K はざらである。もしもグラフを画像ファイルにして表示させたらあんな速度では表示できない。光ファイバならずと早くなると思っている方も多いだろうが、それは利用者が少なくサーバが空いているという条件が整っての話である。ローカルで 100M の LAN を用意して、これをイントラネットとして利用した場合、サーバに同時アクセスで 10 台ぐらいが同じ WEB ページ(画像が多数ある)にアクセスした場合、以外にその表示には時間が掛かる事が理解されるはずだ。

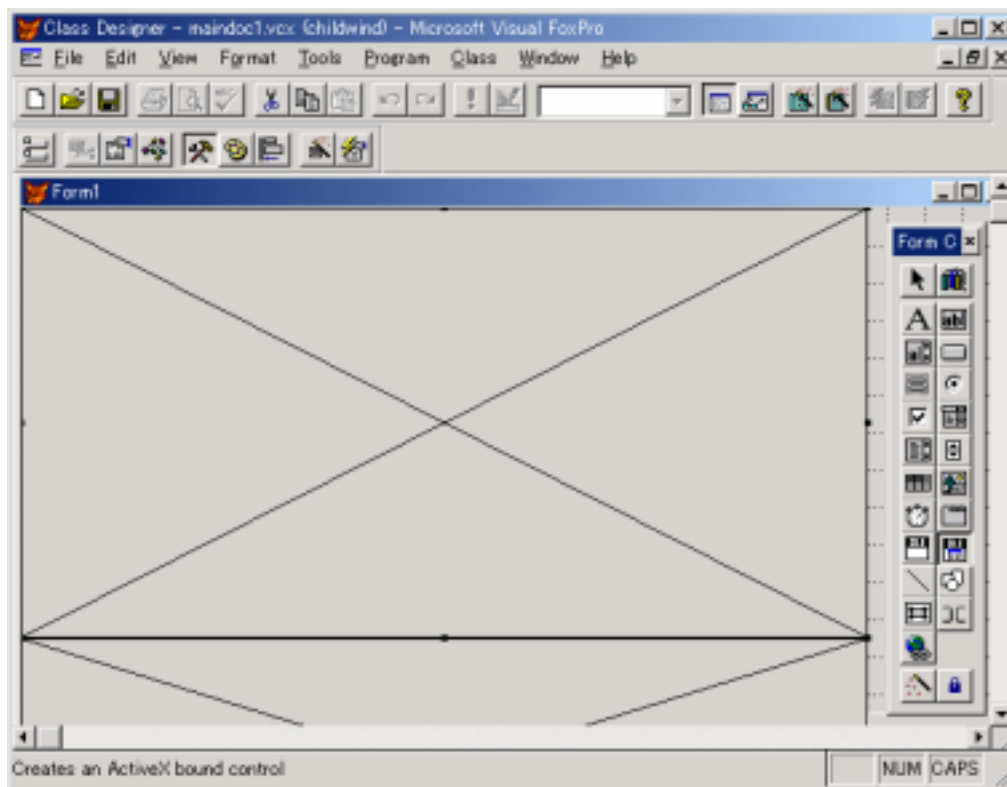
「WEB アプリケーションを考える」

この事についてもう少し私見を述べたいと思う。最近、企業内部の業務システムも WEB アプリケーションでという希望が徐々にではあるが増えているように感じる。理由は幾つか有るが、1. 使っているブラウザから直接業務へ移れる。2. そのシステムをインターネットに公開すれば自宅でも仕事が可能である。3. 同様に携帯電話からも確認や承認が可能となる。4. アプリケーションを更新した場合もメンテナンス作業が不要である。特にこの中で 4. は保守作業の簡略化という面で非常に大きな効果を持っている。フランチャイズで全国展開しているチェーン等にプログラムの変更の度にアプリケーションをダウンロードさせインストールさせるのはリスクが大きい。実は VisualFoxPro でも 1 ~ 4 が全て可能となる。ただし 2. を実現するためにはサーバは WindowsNT となる。また 3. は別

のツールとの組み合わせが必需なので、完全に OK とは言えないが、これは他のツールでも同様だから問題はない。そしてここが重要なのだが、VisualFoxPro にはセキュリティ面で他のツールには真似の出来ないメリットがある。先に VisualFoxPro はクライアントの PC にデータベースを作成する事が可能だという話をしたが、実は VisualFoxPro で作られたアプリケーションはログイン名やパスワードだけでは利用できない裏技を作る事が可能である。それはユーザ側の PC の設定を VisualFoxPro のプログラムから調べる事が出来るのである。簡単なのはユーザマシンの特定のディレクトリにあるはずのファイルがなかったらプログラムをハングさせる事や、更に攻撃してそのマシンにダメージを与える事も可能である。これはプログラム 1 の 4 行目に書かれた IF FILE("") ~ ENDIF を利用すれば簡単に作成する事が出来る。ログイン名とパスワードを特定して潜入した不正行為に対して厳しい仕打ちも出来るのである。例としてはそのマシンをフォーマットしてしまうなんていうのは如何だろうか？永遠にファイルを作りつづけるというのもあるし、特定ファイルやディレクトリ内を丸ごと消去というのもある。サーバそのものを攻撃する不正行為はこれでは防止できないが、インターネット上に用意された会員ページのアプリを無断で使用するような行為は防止出来るはずだ。

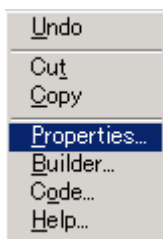
「フォーム上にグラフを表示」

フォームにグラフを表示する作業はあまり単純ではないが、理解できればそれ程難しくはなく、再利用も可能である。



画面 7 フォームに OleBoundControl を貼り付ける

最初に画面7のようにグラフを表示させる位置に OleBoundControl を貼り付ける。この作業はフォームコントロールのツールボックスの ActiveX Bound Control を押し、グラフのエリアを指定する。これが画面7の状態です。ここで指定したエリアをマウスでドラッグしてマウスの右ボタンを押すと画面8のような設定パネルが登場する。



画面8 OleBoundControl の設定

ここで Code を指定してプログラム2のように記述する。

```

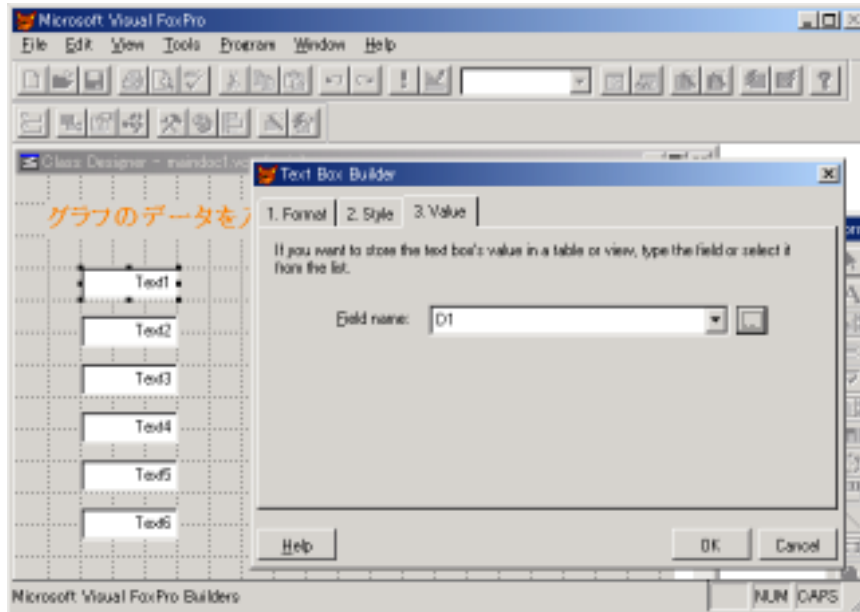
.....
#include "j:\avis\msggraph.h"
#define CRLF CHR(13)+CHR(10)
#define TAB CHR(9)
cGData = "" + TAB + "A" + TAB + "B" + TAB + "C" + CRLF
cGData = cGData + "系列 1" + TAB + alltrim(str(D1)) + TAB +;
          alltrim(str(D3)) + TAB + alltrim(str(D5)) + CRLF
cGData = cGData + "系列 2" + TAB + alltrim(str(D2)) + TAB +;
          alltrim(str(D4)) + TAB + alltrim(str(D6)) + CRLF
*use vfpgrph in 0 SHARED
*use vfpgrph1 in 0 SHARED
cd c:\vfp
sele 1
append blank
append general olegrph class "msggraph.chart" DATA m.cGData
ThisForm.OleBoundControl1.ControlSource = "olegrph"
THISFORM.OleBoundControl1.ChartType = 54
sele 2
append blank
append general olegrph class "msggraph.chart" DATA m.cGData
ThisForm.OleBoundControl2.ControlSource = "olegrph"
THISFORM.OleBoundControl2.ChartType = 60
With THISform.OleBoundControl2.Legend
.Border.Color = RGB(255, 0, 0)
.font.name = "M S ゴシック"

```

.Font.Size = 10

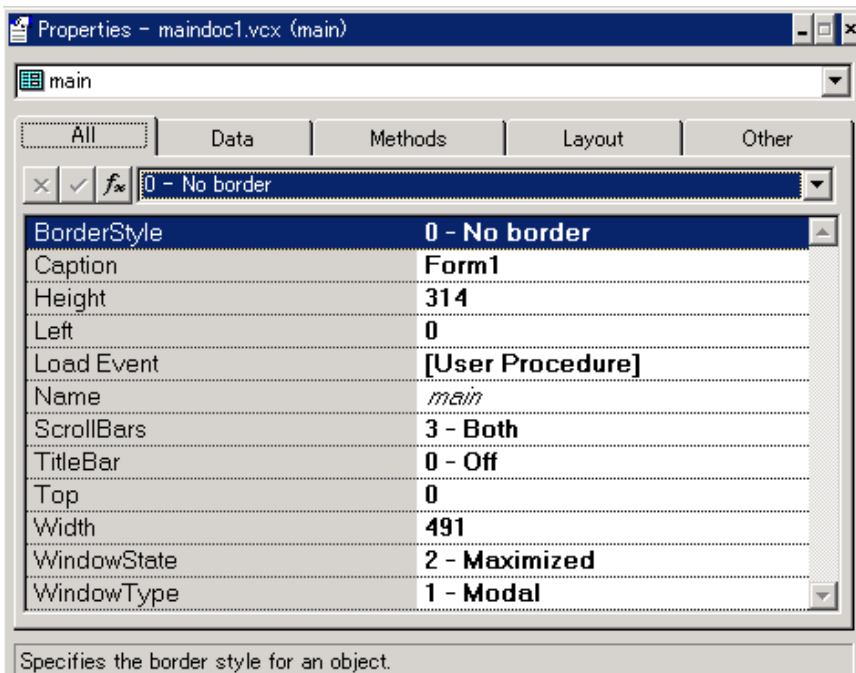
EndWith

プログラム 2 の 4 行目から 8 行目がグラフの数値データを変数「cGData」に入力している部分である。4 行目がグラフの X 軸の定義であり、5 ~ 8 行目が系列と数値データ (D1 ~ D6) までを定義している。ちょっと解説が前後するが、この数値データ (D1 ~ D6) は画面 3 の入力画面に対応する。

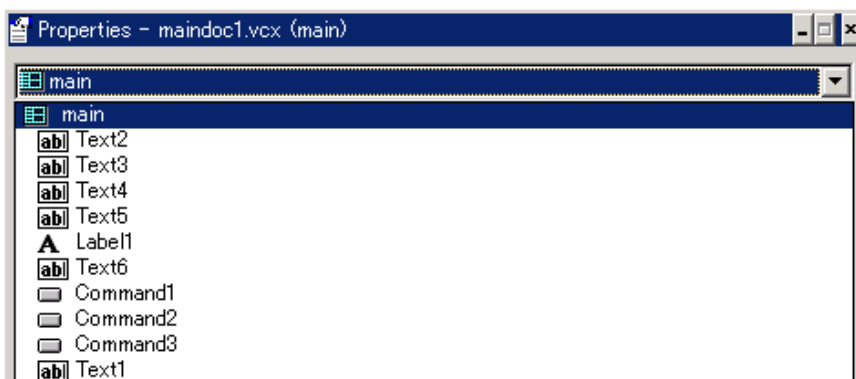


画面 9 入力定義

画面 9 は初期画面の入力の定義を行っている様子である。フォーム上にテキストボックスを最初に一つ作る。このボックスをドラッグすると Builder という項目があるので、ここを選ぶとテキストボックスの定義が出来る。現在は Value のタブを選択しているが、ここでデータベーステーブルのフィールドを選ぶ事も可能だから、簡単に表示画面の定義がここで出来る。グラフデータの入力用に先程のプログラム 2 の変数に相当する (D1 ~ D6) をこのテキストボックスに定義しておく。Format タブでこのボックスを数値型にしておけばここで受け取ったデータはそのままグラフに反映する。画面 10 はフォームのプロパティを表示しているパネルだが、実際にフォームのプロパティ項目はこれだけではない。VisualBasic 等と比較して VisualFoxPro のプロパティはかなりな種類があり、これを全部表示しても辛いものがある。最初の定義では全部表示されるのだが、作業を続けていて変更されている部分だけを見たい場合に変更されたものだけを表示しているのが画面 10 である。画面 11 では画面 10 のプロパティからフォーム上のパーツのプロパティを確認出来るし、設定できる。同じ箇所を連続して変更したい場合はこのパネルだけ見ながらプルダウンで切り替えていけば、テキストボックスなら同じ定義 (高さとかに) の位置にカーソルが固定されるので、効率的に作業が進められる。複数のボタンとかテキストボックスをまとめて移動したい場合の作業方法は VisualBasic と同様で、マウスでまとめてドラッグしてから選択して動かせば移動する。



画面 10 フォームのプロパティ



画面 11 フォームのボタン、テキストボックス等のプロパティ

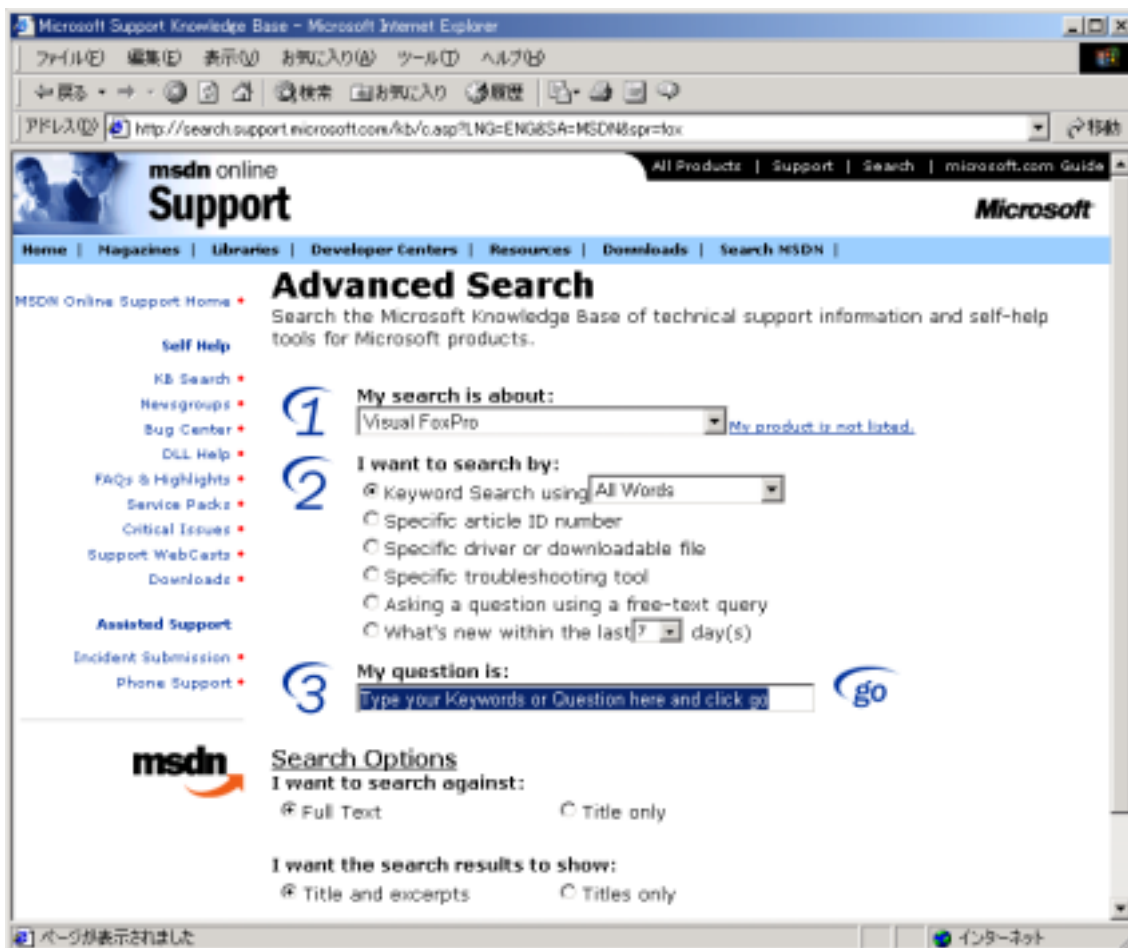
「VisualFoxPro をどうやって理解するか？」

話の順序が逆のようだが、ここまで苦労して簡単な VisualFoxPro のアプリを幾つか作ってきて、自分はどうやって VisualFoxPro を理解する（多少は）に至ったか？ VisualFoxPro の存在は知っていて、日本語は問題なく使える事までは理解でき、実際に稼動する素晴らしいアプリケーションの数々を見てもどうやって作るか？分からないというのが半年前の自分だったのだが、最初は VisualFoxPro について自分が理解できそうな本を AMAZON から取り寄せた。でも読んでいただけではさっぱり分からず、WIZARD で遊んでいた。ここで何が出来るかがかなり具体的になった。そして先月号に掲載した「Visual FoxPro Ver 6.0 アクティブドキュメント徹底解説：市岡信幸 Dec.2000」を参考とした内容でかなり核心に近づいた。そして Object Oriented XBASE Forum に公開されているサンプルプログラムは実に有益だ。（<http://www2u.biglobe.ne.jp/~objxbase/vfpsampl.htm>）だがこれでも無理な部分となると、マイクロソフトの VisualFoxPro ホームページから入れる Microsoft

Knowledge Base of technical support がやはり最終手段であった。

(<http://search.support.microsoft.com/kb/c.asp?LNG=ENG&SA=MSDN&spr=fox>)

次回は更に具体的に開発中の業務アプリケーションの開発過程をご紹介する予定である。



画面 12 Microsoft Knowledge Base of technical support