

はじめてのDelphi/400プログラミング

Delphi/400 を使用してアプリケーションを開発する場合
 まず何をすればいいのだろうか。
 Delphi/400 を使うからといって
 何も特別なことをするわけではない。
 要点は2つである。
 IBM iへの接続方法と既存資産の利用方法を押さえておけばいいのである。
 ここでは、照会機能を持った簡単なアプリケーション開発を事例に挙げ
 はじめての Delphi/400 アプリケーション開発を体感できるようにした。



略歴
 1983年7月6日生
 2006年京都産業大学 法学部卒
 2006年株式会社ミガロ入社
 2006年4月システム事業部配属

現在の仕事内容
 システム受託開発に携わって3年目。ミガロに入社し初めてプログラムを作成するも、今では担当顧客も持ち、小規模から中規模案件のリーダーや大規模案件のサブリーダーを務めるに至る。Delphi/400やRPGなどのプログラム開発経験を積みながら、スキルを磨きつつ、お客様のご要望に耳を傾け、一步一步提案力をつけるため修行中の日々。

- Delphi/400 を使おう
- アプリケーションの全体像をイメージする
- アプリケーションの基本部分を作る 1
 ~ IBM iとの接続 (画面の設定)
- アプリケーションの基本部分を作る 2
 ~ IBM iとの接続 (接続処理)
- メイン機能の実装~データ抽出処理
- さいごに

Delphi/400を使おう

新しくアプリケーションを開発するにあたって、まずは要件を整理する意味も含め必要な機能の洗い出し、それら機能ごとのつながりを画面遷移図としてまとめてみよう。

機能については IBM i のファイルに対して何を行うのかで、大きく3つに種類分けを行うことが可能である。

- ・照会機能：IBM iのファイルを、さまざまな抽出条件で参照する。
- ・更新機能：IBM iのファイルに新しいレコードを追加したり、すでにあるレコードを更新、または削除する。
- ・支援機能：上記の2機能のほか、検索や帳票出力、異なるアプリケーションの呼び出しや連携などを行う。

今回は「はじめての Delphi/400」ということで、IBM i 上にあるファイルを参照する照会機能を備えたアプリケーションの開発についてまとめることにする。

アプリケーションの全体像をイメージする

今回作るアプリケーションの構成は、全部で3画面となる。

まずは、アプリケーションの起動時に最初に表示させる画面として、サインオンダイアログを準備する。これは、どんなアプリケーションを作ろうと、IBM i のファイルを参照するのであれば、IBM i への接続が必要不可欠だからだ。今回のアプリケーションでは、ユーザーに ID とパスワードを入力してもらい、その入力値をもとに IBM i との接続を行うことにする。

無事に IBM i との接続ができれば、次画面として機能を統括するメニューを表示させ、このメニューからさまざまな機能へ展開できるようにしよう。単一の機能しかないアプリケーションであれば、メニューを作成しなくてもよいが、将来機能を増やすことも想定してメニュー画面を用意することにする。

そして、メニュー画面で「照会画面へ」

を押下することで、メインとなる IBM i 上のファイルを照会する3画面目を表示させる。

上記の内容を、遷移図にまとめた。全体像がイメージできたので、さっそく開発に着手していこう。【図1】

アプリケーションの基本部分を作る 1 ~IBM iとの接続 (画面の設定)

Delphi/400 アプリケーションを作成するうえで、IBM i への接続 (サインオン) が必要になる。データベースアプリケーションでは、データベースへサインオンするために、ユーザーとパスワードの情報が必要となる。もちろん Delphi/400 も例外ではない。

サインオンするためのユーザー / パスワードを設定する方法は、大きく分けて2つの手法がある。

- ①アプリケーション上で、接続情報を入力する (明示的サインオン)

図1 概要図

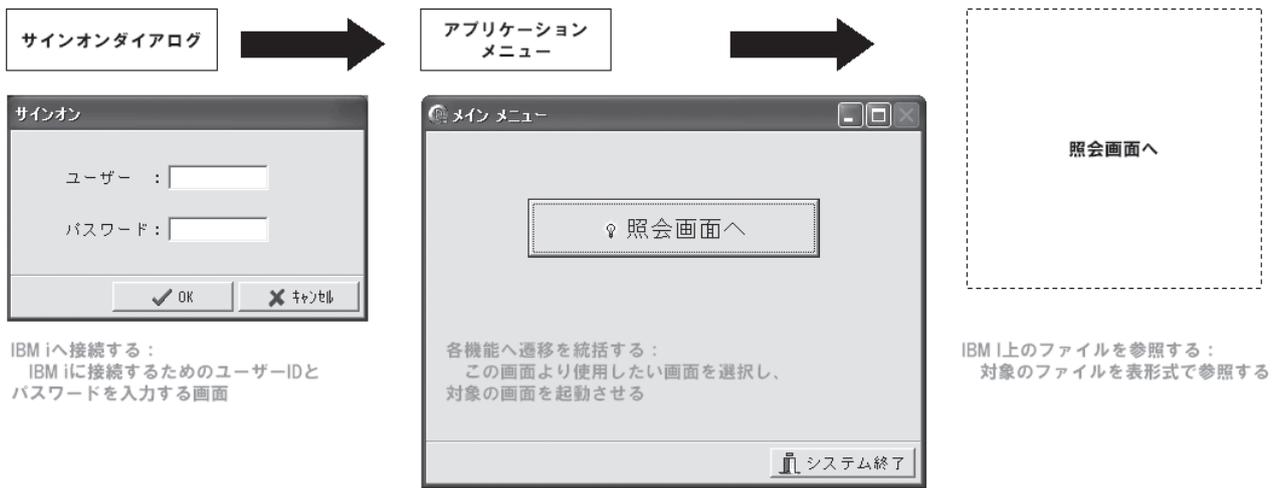


図2 プロジェクト追加

Borland Delphi 2005 for Microsoft Windows

ファイル(F) 編集(E) 検索(S) 表示(V) リファクタリング(Q) プロジェクト(P) 実行(R) コンポーネント(C)

新規作成(N) ▶ VCL フォームアプリケーション - Delphi for Win32(V)

開く(O)...

プロジェクトを開く(O)... Ctrl+F11

開き直す(R)

書き込み保存(S) Ctrl+S

名前を付けて保存(A)...

プロジェクトに名前を付けて保存(E)...

すべて保存(V) Shift+Ctrl+S

閉じる(C)

すべて閉じる(L)

ユニットを使う(U)... Alt+F11

印刷(P)...

終了(X)

パッケージ - Delphi for Win32(D)

フォーム - Delphi for Win32(E)

ユニット - Delphi for Win32(L)

Data Module - Delphi for Win32

その他(O)...

カスタマイズ(C)...

リリース情報

Readme

インストール

配布

マニュアル

ファーストステップ

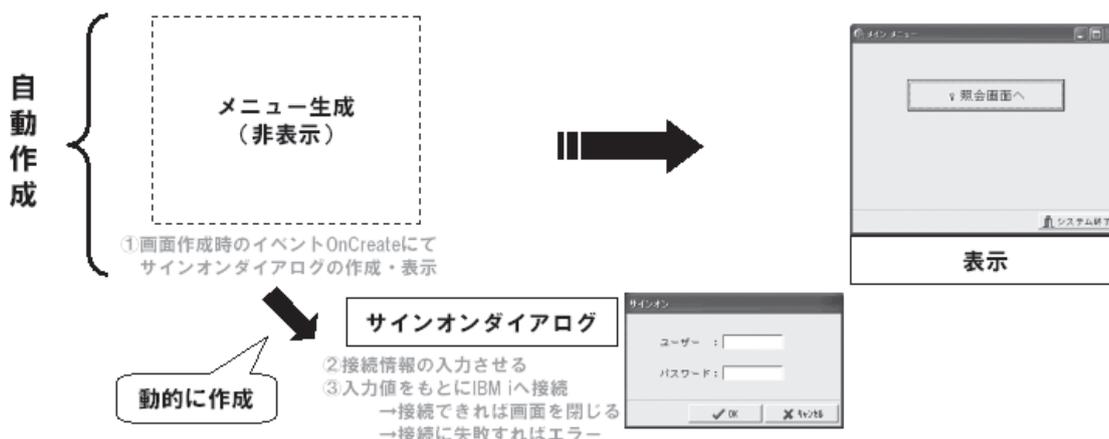
ハウツーガイド

新規プロジェクトの作成はこちらから

ここからプロジェクトに新しいフォームを追加できる

プロジェクト : SampleProject		
	Nameプロパティ	保存時のファイル名
ダイアログ	frmSignOn	SignOnFrm
メニュー	frmMenu	MenuFrm

図3 作成の流れ



サインオンダイアログ画面よりユーザー/パスワードを入力した内容で、接続を行う。

②プログラム上で、内部的に接続情報を設定する(暗黙のサインオン)

プログラム上で固定のユーザー/パスワードを設定しておくか、応用して外部ファイル(iniファイル等)からユーザー/パスワードを読み込んで、接続を行う。

今回は①の手法を使って開発を行う。それでは、接続に成功した後にメニューを表示させる部分までを含め、作ってみよう。

プロジェクト追加

Delphiのメニューから[ファイル][新規作成][VCLフォームアプリケーション]を選択する。すると、画面Unit1を持つ新規プロジェクトProject1が作成できる

まず、Unit1の画面にあたるForm1のNameプロパティを「frmSignOn」と変更し、続けてDelphiのメニューから[ファイル]の[名前を付けて保存]と、[プロジェクトに名前を付けて保存]より、Unit1を「SignOnFrm」、Project1を「SampleProject」と名付け、適当なフォルダに保存する。

保存ができたなら、同様に、使用するメニューを同じプロジェクトに追加しておく。【図2】

なお、FormのNameプロパティとUnitのファイル名には、同じ名前を指定できないので注意が必要である。

作成の流れ

これで、接続までの流れで使用する画面が用意できた。

というわけだが、プロジェクトに追加されたフォームは、初期設定では起動時に自動作成されるようになっている。起動時に自動作成されるということは、その先に遷移しない可能性のある画面をも起動時に作成してしまい、無駄が発生するということである。

そこでDelphiのメニューから[プロジェクト][オプション]を選択し、ツリーメニューのフォーム項目から自動作成の対象を必要最低限に設定しておくこと。

今回は、メニューを設定し、メニュー

の生成時にサインオンダイアログを表示し、入力された情報をもとにIBM iに接続、その後メニューを表示という流れにする。【図3】【図4】

また、メニューがサインオンダイアログを使用するには、参照関係を設定する必要がある。Delphiのメニューより[ファイル][ユニットを使う]を選択し、サインオンダイアログを参照できるように設定する。【図5】

コンポーネント

これで土台となる必要な画面が用意できた。ここからは、部品となるコンポーネントを、ツールパレットより各フォームに貼り付けていく。

【使用するコンポーネント】

● frmSignOn

- ・ TPanel: ツールパレット (Standard) コンテナ (土台) コンポーネント。
- ・ TEdit: ツールパレット (Standard) ユーザーに入力域を提供する、標準的なコンポーネント。
- ・ TBitBtn: ツールパレット (Additional) ビットマップを表示できるボタン。よく使用される動作がビットマップ付きで実装されている。

● frmMenu

- ・ TAS400: ツールパレット (SCD400 Data) IBM iとクライアントとの通信を管理するコンポーネント。コマンドを発行することも可能。
- ・ TDataBase: ツールパレット (BDE) BDEを使用したアプリケーションが、データベースに接続する。その管理を行うコンポーネント。
- ・ TButton: ツールパレット (Standard) 標準的なプッシュボタン。

コンポーネントの貼り付けが終われば、それぞれのプロパティ値を設定していく。

まず、IBM iへの接続の要となるAS400コンポーネントであるが、特に変更するところはない。次にデータベースコンポーネントでは、今回はサインオンダイアログを作成するので、Login PromptプロパティをFalseにしておく。

【図6-1】【図6-2】

あとはビットボタンのKindプロパティを変更する程度で、デフォルトの値を使う。(Kindプロパティを変更すると、それにあった外観と応答結果が自動で変更される点に留意してほしい。必要に応じてプロパティを再変更することを覚えておくといいだろう)。【図6-2】

アプリケーションの基本部分を作る 2 ～IBM iとの接続 (接続処理)

以上で一通りの準備が完了できた。いよいよIBM iとの接続処理をプログラミングしていく。

Delphi/400を使ったIBM iへの接続は、AS400コンポーネント、データベースコンポーネントのユーザーIDおよびパスワードのプロパティにそれぞれの値をセットし、各コンポーネントをOpenするだけのシンプルなプログラミングのみで可能となる。

また、AS400コンポーネントではコマンドの発行ができるので、接続後には「ADDLIB」コマンドからライブラリー環境の設定を行うことも可能である。今回は初期値を「*LIBL」とし、「ADDLIB」コマンドより、対象のライブラリーをセッションのライブラリーリストに追加する。

これを踏まえて図7のように、アプリケーション起動時に自動生成されるメニューのOnCreateイベントで、サインオンダイアログを表示させて接続を行うようにする。ここでは、受け取った引数をもとに接続処理を行い、ライブラリー環境を設定する一連の処理を、関数[OpenDatabase]としてメニューに作成している。【図7】【ソース1】【ソース2】

以上のプログラミングで、Delphi/400を使ってIBM iに接続を行い、アプリケーション終了時までその接続状態を保持する基本部分が完成した。接続後はもちろん、5250画面の「WRKACTJOB」コマンドで確認できる。(このプログラムでは接続先を指定していないので、Delphi/400 Configurationの「AS/400List」にある最上部のIBM iへ接続される)。

図4 自動作成

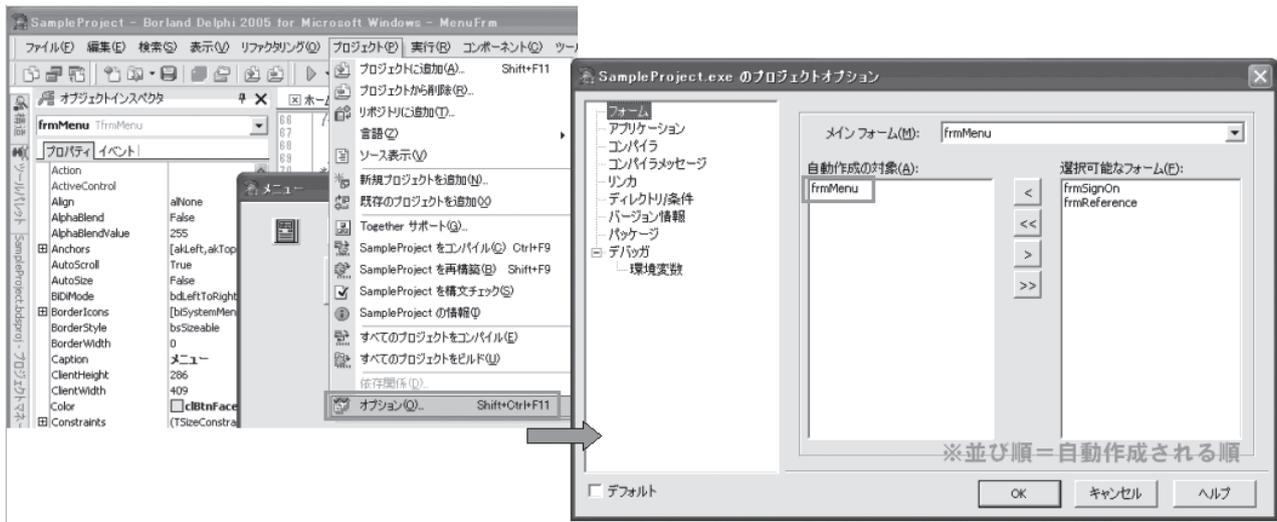
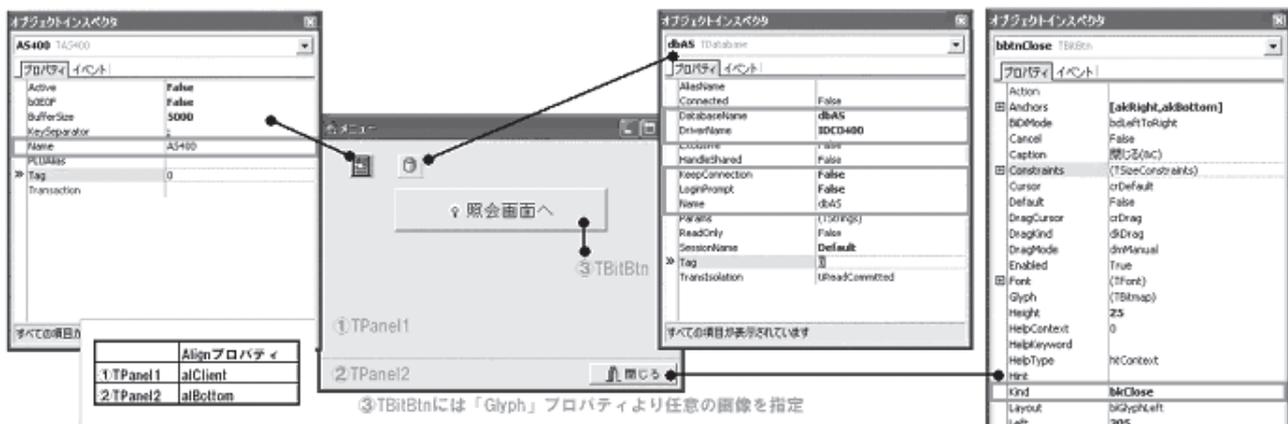


図5 ユニットの使用



図6-1 プロパティ



メイン機能の実装 ～データ抽出処理

アプリケーションの基本部分が完成したら、いよいよメインの照会機能のプログラミングに取りかかろう。

Delphi/400 を利用した IBM i のファイル参照は、大きく分けて 2 種類の方法がある。

- ① SQL でデータを取得する方法
- ② ファイル単位で取得する方法

今回は、すでに IBM i 側に、RPG プログラムで作成された照会システムがあることを前提とする。CL プログラムや RPG プログラムとの連携で実現する②の手法を使って、開発を行い、IBM i 資産を利用する方法について紹介しよう。

既存プログラムの利用

すでに IBM i にある照会機能の GUI 化を Delphi/400 で行う場合、特に工数の削減が見込めるのが、既存プログラムを利用するという方法である。

図 8 を見てほしい。既存プログラムについて抽出結果をサブファイルに出力する代わりに、ワークファイルに出力するようにメンテナンスするだけで、既存の RPG/CL プログラムを流用することができる。

その際、ワークファイルのフィールド名を画面ファイルの項目と同じにすることで、変更を最小限に抑えることが可能である。

さらに、Delphi/400 プログラムから IBM i のプログラムを呼び出すために、CL プログラムを経由するというポイントも押さえていただきたい。CL において「MONMSG」を使用することで、呼び出す RPG プログラムの MSGW を回避することができるのである。

また、CL プログラムに、抽出条件のパラメータ以外にエラー区分のパラメータを持たせることで、対象データがない場合などのエラー発生を、Delphi/400 側に返すことが可能となる。【図 8-1】

今回は、図 8-2 のように、画面起動時に QTEMP 上にワークファイルを作る CL プログラムを実行し、条件入力後に検索ボタンを押下することで、メインの抽出処理 RPG プログラムを呼び出す

CL プログラムを実行することとする。【図 8-2】【表】【ソース 3】【ソース 4】【ソース 5】

【使用するコンポーネント】

● frmReference (フォームを新規作成しメニューから呼び出す)

・ TMaskEdit : ツールパレット (Additional)

入力された文字を指定した表示形式で検証し、表示する。

・ TCall400 : ツールパレット (SCD400 System)

IBM i 上のプログラムを呼び出す。パラメータはプログラムからセットすることも、実行後に返された値を参照することも可能。

・ TTable : ツールパレット (BDE)

データベース上のファイルを保持・操作するデータセット。

・ TDataSource : ツールパレット (Data Access)

データセットコンポーネントと、画面上にあるデータベース対応コンポーネントをつなぐ役割。

・ TDBGrid : ツールパレット

(Data Controls)

対象のデータセットより、レコードを表形式で表示し、操作する。

なお、照会画面はメニューを参照できるようユニット参照を行い、また、プロジェクトオプションで、自動作成の対象より除外すること。その他の、既出のコンポーネントについては割愛した。

プロパティ

プロパティは、図 9-1 と図 9-2 を参照してほしい。【図 9-1】【図 9-2】

ここでのポイントは、Call400 コンポーネントで、IBM i の CL プログラムへパラメータを渡すので、同数のパラメータ定義および各値の桁数と属性を指定する必要があるということだ。(ただし、属性については、プログラミング上は、全て文字型 (String) でセットする点に注意)。【ソース 5】

ここで定義したパラメータに、プログラムから動的に値をセットし、CL プログラムを呼ぶのである。また、テーブルコンポーネントには、参照対象となるファイル名を設定する。そして、いずれ

のコンポーネントの AS400 プロパティにも、メニューに配置した AS400 コンポーネントを指定することをあわせて確認しておこう。

CL プログラム

さて、各コンポーネントの準備ができたので、プログラミングを行っていこう。

と、いっても、Delphi/400 から CL プログラムを実行し、対象データが更新されたワークファイルを参照するというのは、いたってシンプルなプログラミングで済む。

Call400 コンポーネントに設定した各パラメータに、ただ画面入力値をセットし、実行した後に、プロパティが正しくセットされたテーブルコンポーネントを開くだけで、画面に表示できるのである。

【図 10】【ソース 6】

これで全てのプログラムが完成した。Delphi のメニューより [プロジェクト] [SampleProject をコンパイル] を選択し、エラーがないことを確認した後に、実行ファイル SampleProject.exe を実行してみよう。そして、ぜひとも IBM i 上にあるファイルを参照できていることを、ご自身の目で確かめていただきたい。

さいごに

以上で、照会機能を想定し、Delphi/400 を使用する一連のアプリケーション開発についての説明は終わりである。冒頭で述べたように「はじめての Delphi/400」と題したが、IBM i への接続、既存資産の利用方法が把握できていると、GUI 化は思うがままなのである。

とはいえ、アプリケーションからの SQL 実行により、ファイルの操作を簡単に行うことも可能である。もちろん Delphi/400 を使用した接続なので、早いレスポンスが期待できる。

この「はじめての Delphi/400」が、Delphi/400 ユーザーである開発者の増加の一助になれば幸いである。

■

図6-2 プロパティ

すべての項目が表示されています

プロパティ	値
1.TPanel1	alClient
2.TPanel2	alBottom
3.TLabel	「Caption」プロパティのみ変更

図7 各フォームのイベント

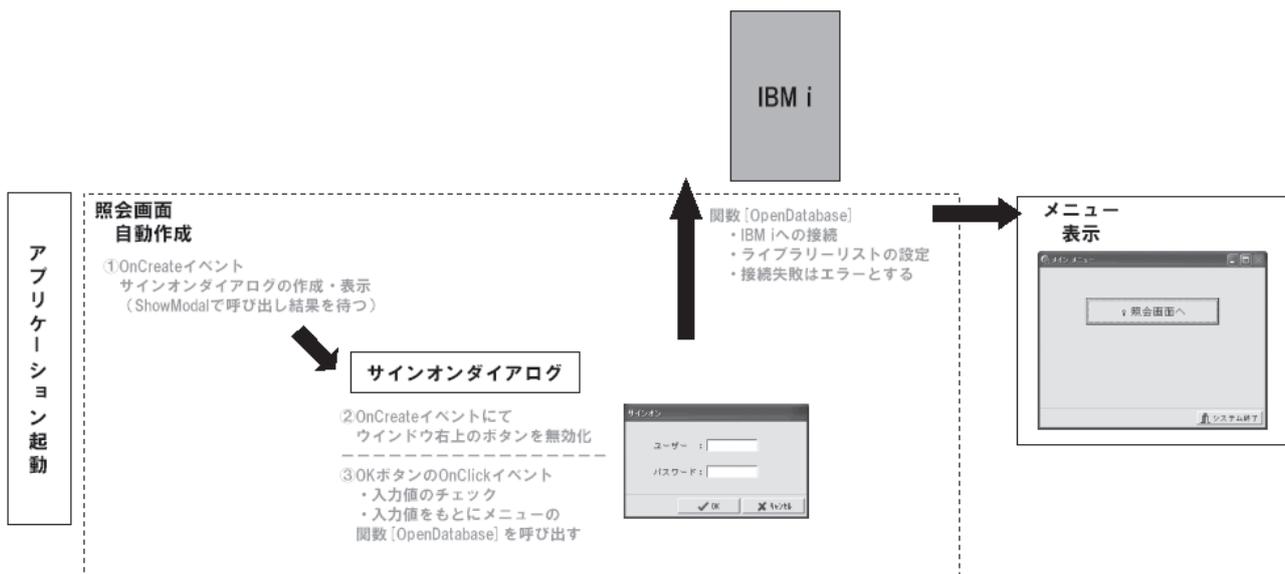


図9-2 プロパティ

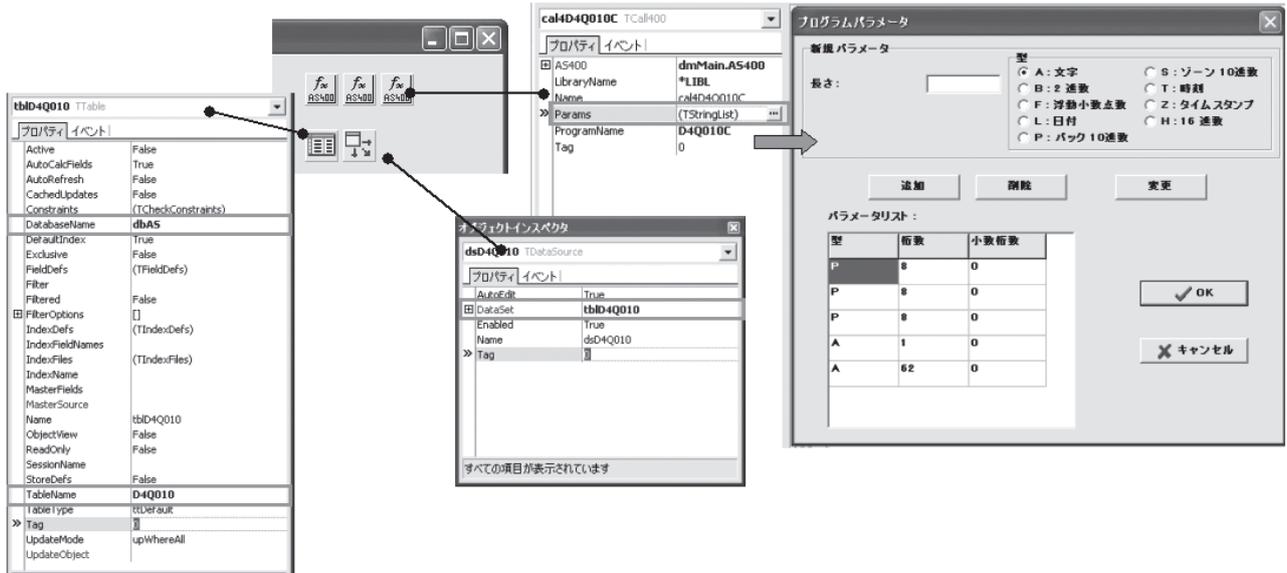


図10 照会の流れ

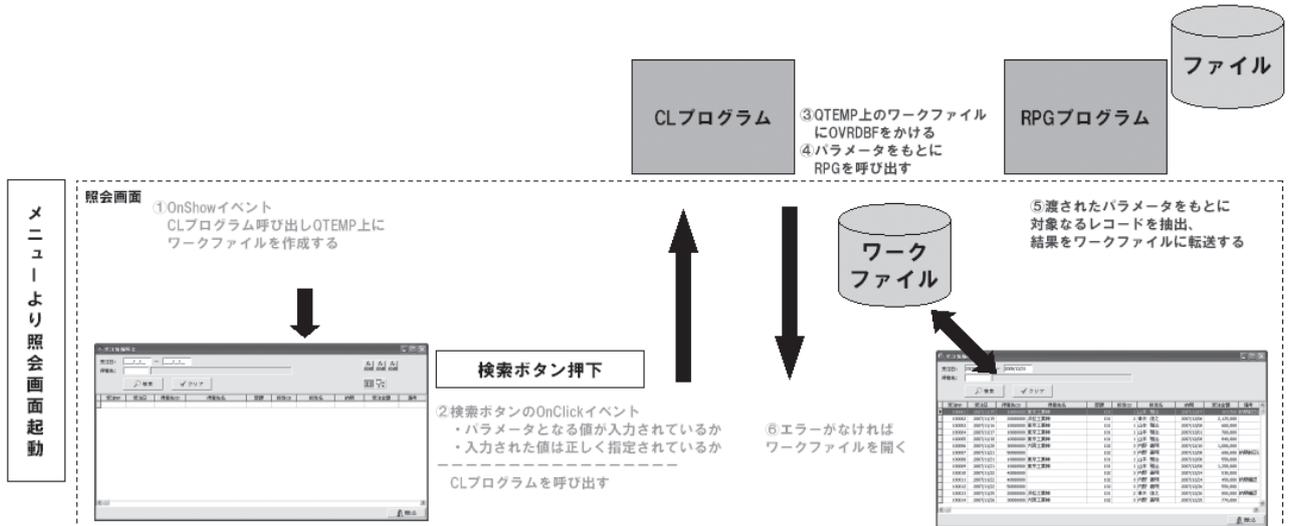


表 IBM iプログラムおよびファイルレイアウト

プログラム一覧		
	ID	名称
CL	D4Q010C	受注一覧照会（呼出）
	D4Q011C	受注一覧照会QTEMP作成
RPG	D4Q010	受注一覧照会ワーク作成

ファイルレイアウト			
ファイル名：受注一覧ワーク		ファイルID：D4Q010	
項目名	桁数	属性	テキスト記述／欄見出し
HWJGCD	4	0 S	部課
HWJUNO	6	0 S	受注No.
HWJUDT	8	0 S	受注日
HWTRRK	14	0	得意先名
HWTANM	16	0	営業担当者名
HWNOKI	8	0 S	納期日
HWJUGK	9	0 P	受注金額
HWTKCD	8	0 S	得意先 CD
HWTACD	5	0 S	担当者 CD
HWBIKO	32	0	備考

ソース1 サインオンダイアログ

```

1  #####
2
3  サインオンダイアログ
4
5  #####
6  unit SignOnFrm;
7
8  interface
9
10 uses
11   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
12   Dialogs, StdCtrls, Buttons, ExtCtrls;
13
14 type
15   TfrmSignOn = class(TForm)
16     pnlBottom: TPanel;
17     bbtnOK: TBitBtn;
18     bbtnCancel: TBitBtn;
19     pnlMain: TPanel;
20     Label1: TLabel;
21     Label2: TLabel;
22     meUserID: TEdit;
23     mePassWord: TEdit;
24     procedure bbtnOKClick(Sender: TObject);
25     procedure FormCreate(Sender: TObject);
26   private
27     { Private 宣言 }
28   public
29     { Public 宣言 }
30   end;
31
32 var
33   frmSignOn: TfrmSignOn;
34
35 implementation
36
37 uses MenuFrm;
38
39 {$R *.dfm}
40
41 {*****}
42 目的: 画面作成時
43 引数:
44 戻値:
45 {*****}
46 procedure TfrmSignOn.FormCreate(Sender: TObject);
47 var
48   SysMenu: HMENU;
49 * begin
50   //閉じるボタンを無効にする
51   SysMenu := GetSystemMenu(Handle, False);
52   EnableMenuItem(SysMenu, SC_CLOSE, MF_BYCOMMAND or MF_GRAYED);
53   SysMenu := GetSystemMenu(Application.Handle, False);
54   EnableMenuItem(SysMenu, SC_CLOSE, MF_BYCOMMAND or MF_GRAYED);
55 * end;
56
57 {*****}
58 目的: OK処理
59 引数:
60 戻値:
61 {*****}
62 procedure TfrmSignOn.bbtnOKClick(Sender: TObject);
63 * begin
64   inherited;
65   // ユーザーID未入力チェック
66   if meUserID.Text = '' then
67     begin
68       meUserID.SetFocus;
69       raise Exception.Create('ユーザーIDを入力して下さい。');
70     end;
71
72   // パスワード未入力チェック
73   if mePassWord.Text = '' then
74     begin
75       mePassWord.SetFocus;
76       raise Exception.Create('パスワードを入力して下さい。');
77     end;
78
79   // ユーザーIDへフォーカス移動
80   meUserID.SetFocus;
81
82   // AS/400への接続処理
83   frmMenu.OpenDatabase(meUserID.Text, mePassWord.Text);
84
85   // 接続できれば終了
86   ModalResult := mrOk;
87 * end;
88
89 * end.

```

【図VII】②
ウインドウ右上のボタンを無効化

【図VII】③
入力チェック

【図VII】③
メニュー【OpenDatabase】の呼び出しにてIBM iへ接続

ソース2 メニュー

```

1  {#####}
2
3  メニュー
4
5  {#####}
6  unit MenuFrm;
7
8  interface
9
10 uses
11   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
12   Dialogs, StdCtrls, Buttons, ExtCtrls, DB, DBTables, Scdconn;
13
14 type
15   TfrmMenu = class(TForm)
16     pnlBottom: TPanel;
17     bbtnClose: TBitBtn;
18     pnlMain: TPanel;
19     BitBtn1: TBitBtn;
20     AS400: TAS400;
21     dbAS: TDatabase;
22     procedure FormDestroy(Sender: TObject);
23     procedure FormCreate(Sender: TObject);
24     procedure BitBtn1Click(Sender: TObject);
25   private
26     { Private 宣言 }
27   public
28     { Public 宣言 }
29     // AS/400への接続処理
30     procedure OpenDatabase(AUserID, APassword :String);
31   end;
32
33 var
34   frmMenu: TfrmMenu;
35
36 implementation
37
38 uses ReferenceFrm, SignOnFrm;
39
40 {$R *.dfm}
41
42 {#####}
43 目的: メニューフォーム作成時処理
44 引数:
45 戻値:
46 {#####}
47 procedure TfrmMenu.FormCreate(Sender: TObject);
48 begin
49   // サインオンダイアログを表示
50   frmSignOn := TfrmSignOn.Create(Application);
51   try
52     // サインオンダイアログより接続結果を受け取り
53     // 接続出来なかった場合、アプリケーションを終了する
54     if frmSignOn.ShowModal <> mrOK then
55       begin
56         Application.ShowMainForm := False;
57         Application.Terminate;
58         Abort;
59       end;
60     finally
61       frmSignOn.Release;
62     end;
63   end;
64
65 {#####}
66 目的: メニューフォーム破棄時処理
67 引数:
68 戻値:
69 {#####}
70 procedure TfrmMenu.FormDestroy(Sender: TObject);
71 begin
72   // TDatabase切断処理
73   dbAS.Connected := False;
74
75   // TAS400切断処理
76   AS400.Active := False;
77 end;
78

```

【図VII】
関数【OpenDatabase】宣言部分

【図VII】①
サインオンダイアログの表示

●
●
●

```

78
79
80 {*****}
81 目的： 照会画面遷移ボタン押下時処理
82 引数：
83 戻値：
84 {*****}
85 procedure TfrmMenu.BitBtn1Click(Sender: TObject);
86 begin
87     inherited;
88     // 照会画面の呼び出し
89     frmReference := TfrmReference.Create(Self);
90     try
91         Self.Hide;
92         frmReference.ShowModal;
93     finally
94         frmReference.Release;
95         Self.Show;
96     end;
97 end;
98
99 {*****}
100 目的： IBM iへの接続処理
101 引数：
102 戻値：
103 {*****}
104 procedure TfrmMenu.OpenDatabase(AUserID, APassword :String);
105 var
106     sMsg :String;
107 begin
108     // 初期化処理
109     // TAS400の切断
110     if AS400.Active then AS400.Active := False;
111     // TDatabaseの切断
112     if dbAS.Connected then dbAS.Connected := False;
113
114     // TAS400の設定
115     with AS400 do
116     begin
117         // ユーザーID
118         Userid := AUserID;
119         // パスワード
120         PWD := APassword;
121     end;
122
123     // TDatabaseの設定
124     with dbAS do
125     begin
126         // ライブラリー
127         Params.Values['LIBRARY NAME'] := '*LIBL'; {※「*LIBL」とすることで接続されたセッション
128         // ユーザーID
129         Params.Values['USER NAME']:= AUserID;
130         // パスワード
131         Params.Values['PASSWORD'] := APassword;
132         // ログインプロンプトの表示
133         LoginPrompt := False;
134     end;
135
136     // データベース接続処理
137     try
138         AS400.Active := True;
139         dbAS.Connected := True;
140     except
141         // エラー発生時
142         sMsg := 'AS/400に接続することができませんでした。';
143         sMsg := sMsg + #13#10 + 'ユーザー名、パスワードを確認して下さい。';
144         raise Exception.Create(sMsg);
145     end;
146
147     //ライブラリーリストの追加 (ここでは"TESTLIB"とし、QTEMPのうしろに追加)
148     try
149         AS400.RemoteCmd('ADDLIB LIB(TESTLIB) POSITION(*AFTER QTEMP)');
150     except
151         // エラー発生時
152         on E: Exception do
153             begin
154                 raise Exception.Create(E.Message);
155             end;
156     end;
157 end;
158
159 end.

```

メニュー
「照会画面へ」ボタン押下時



図VII
関数【OpenDatabase】IBM iへの接続

ソース3 CL1

```

/*****
/* システム : 受注照会機能 *
/* NAME : 受注一覧照会 (呼出) *
/*****
PGM PARM(&PJUDF +
&PJUDT +
&PTKCD +
&PPERR +
&PPMSG )

/* CL変数定義 */
/* 受注日 FROM (数値8桁) */
DCL VAR(&PJUDF) TYPE(*DEC) LEN(8 0)
/* 受注日 TO (数値8桁) */
DCL VAR(&PJUDT) TYPE(*DEC) LEN(8 0)
/* 得意先コード (数値8桁) */
DCL VAR(&PTKCD) TYPE(*DEC) LEN(8 0)
/* エラー (文字1桁) */
DCL VAR(&PPERR) TYPE(*CHAR) LEN(1)
/* メッセージ (文字62桁) */
DCL VAR(&PPMSG) TYPE(*CHAR) LEN(62)

/*-----*/
/* QTEMP */
/*-----*/
CALL D4Q011C
/*-----*/
/* 更新処理 */
/*-----*/
OVRDBF FILE(D4Q010) TOFILE(QTEMP/D4Q010) LVLCHK(*NO)
MONMSG MSGID(CPF0000)
CALL PGM(D4Q010) PARM(&PJUDF +
&PJUDT +
&PTKCD +
&PPERR +
&PPMSG )

DLTOVR FILE(D4Q010)
MONMSG MSGID(CPF0000)
/*-----*/
/* 初期モード設定 */
/*-----*/
RETURN
ENDPGM

```

ソース4 CL2

```

/*****
/* システム : 受注照会機能 *
/* NAME : 受注一覧照会 QTEMP作成 *
/*****
#START: PGM
/* 変数宣言 */
DCL VAR(&SRCLIB) TYPE(*CHAR) LEN(10)
/* パラメーター */
RTVDTAARA DTAARA(D4DTAARA (1 10)) RTNVAR(&SRCLIB)
/* オブジェクト確認 */
CHKOBJ OBJ(QTEMP/D4Q010) OBJTYPE(*FILE)
MONMSG MSGID(CPF9801) EXEC(GOTO CMDLBL(#CRTTBLA))
GOTO #END
/* テーブル作成 */
#CRTTBLA:
CRTPF FILE(QTEMP/D4Q010) SRCFILE(&SRCLIB/QDDSSRC) +
OPTION(*NOSRC *NOLIST) SIZE(*NOMAX) +
LVLCHK(*NO)
/* プログラムメッセージ・モニター */
MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(#PGMERR))
/* 正常終了 */
#END:
CLRPFM FILE(QTEMP/D4Q010)
MONMSG MSGID(CPF0000)
RETURN
/* エラー終了 */
#PGMERR: ENDPGM

```

ソース5 RGB

```

***** データの始め *****
H*-----*
H* PROGRAM 名      : D4Q010
H* PROGRAM 見出   : 受注一覧照会ワーク作成
H*          作成日 :
H*          更新日 :
H*-----*
H          Y/                      1
F*-----*
F*          F I L E
F*-----*
F*<< 受注DB (得意先指定) >>
FDTHJL01IF E      K      DISK
F*<< 受注DB (受注日指定) >>
FDTHJL02IF E      K      DISK
F          HJR00                      KRENAMEHJR02
F*<< 取引先M >>
FDTMMCP IF E      K      DISK
F*<< 社員M >>
FDTMMBP IF E      K      DISK
F*<< メッセージM >>
FDTMMJP IF E      K      DISK
F* 受注一覧照会ワーク
FD4Q010  O  E      K      DISK
E*-----*
E*          配列
E*-----*
I*-----*
I*          D S
I*-----*
I* プログラムID
I          SDS
I                      244 253 WSID
I                      254 263 WSUSR
I                      *PROGRAM D#PGID
C*-----*
C*          P L I S T
C*-----*
C          *ENTRY  PLIST
C          PARM      PPJUDF 80      * 受注日 FROM
C          PARM      PPJUDT 80      * 受注日 TO
C          PARM      PPTKCD 80      * 得意先コード
C          PARM      PPERR  1      * エラー
C          PARM      PPMMSG 62      * メッセージ
C*-----*
C*          K L I S T
C*-----*
C* 受注DB
C          KEYHJ1  KLIST
C          KFLD      HJTKCD      得意先コード
C          KFLD      HJJUDT      受注日
C          KEYHJ2  KLIST
C          KFLD      HJTKCD      得意先コード
C* 受注DB (受注日)
C          KEYHJ3  KLIST
C          KFLD      HJJUDT      受注日
C* 取引先M
C          KEYMC1  KLIST
C          KFLD      MCTRCD
C* 社員M
C          KEYMB1  KLIST
C          KFLD      MBTACD      社員コード
C*-----*
C*          M A I N  R O U T I N E
C*-----*
C*<< 初期設定ルーチン(とぶ。)>>
C          EXSR SBINZ
C*
C          EXSR SB0000
C*
C          EXSR SBEND
C*-----*
C*          SBINZ  初期設定サブルーチン
C*-----*
C          SBINZ  BEGSR

```

•
•
•
•
•

```

C*-----*
C* SBINZ 初期設定サブルーチン
C*-----*
C SBINZ BEGSR
C<<< 受注 DB>>
C *LIKE DEFN HJJUDT WKJUDT * 受注日
C *LIKE DEFN HJTKCD WKTKCD * 得意先
C MOVEL*BLANK WKFST 1 * エラー
C MOVEL'0' WKOK 1 * 処理 OK
C MOVEL*BLANK PPMG * メッセージ
C<<<KEY 判断 >>
C MOVEL'0' *IN70
C PPTKCD IFNE 0
C MOVEL'1' *IN70
C ENDIF
C ENDSR
C*-----*
C* SB0000 メイン処理
C*-----*
C SB0000 BEGSR
C Z-ADDPJUDF HJJUDT 受注日
C<<< 得意先処理 >>
C *IN70 IFEQ '1'
C Z-ADDPPTKCD HJTKCD 得意先コード
C KEYHJ1 SETLLHJR00
C ELSE
C<<< 受注日処理 >>
C KEYHJ3 SETLLHJR02
C ENDIF
C MOVEL'0' *IN90
C<<< メインREAD >>
C *IN90 DOWEQ'0'
C *IN70 IFEQ '1'
C KEYHJ2 READEHJR00 90
C ELSE
C READ HJR02 90
C ENDIF
C<<< ファイルEOF >>
C *IN90 IFEQ '1'
C LEAVE
C ENDIF
C<<< 受注日範囲を超えた場合終了 >>
C HJJUDT IFGT PPJUDT
C LEAVE
C ENDIF
C<<< ワークファイル更新 >>
C EXSR SB1000
C ENDDO
C<<< エラー処理 >>
C WKFST IFEQ *BLANK
C MOVEL'1' PPERR
C MOVEL'E0010' MJMSGC
C EXSR SBMSG
C ENDIF
C ENDSR
C*-----*
C* SB1000 ワークファイル更新
C*-----*
C SB1000 BEGSR
C
C MOVEL'1' WKFST
C
C CLEARHWR00
C Z-ADDHJGCD HWJGCD 部課 CD
C Z-ADDHJUNO HWJUNO 受注No.
C Z-ADDHJJUDT HWJUDT 受注日
C Z-ADDHJTKCD HWTKCD 得意先 CD
C Z-ADDHJTKCD MCTRCD 得意先 CD
C N91 KEYMC1 CHAINMCR00 91
C MOVELMCTRRC HWTRRK 取引先略称
C Z-ADDHJTACD HWTACD 営業担当者
C<<< 営業担当者名取得 >>
C Z-ADDHJTACD MBTACD 営業担当者
C N91 KEYMB1 CHAINMBR00 91
C MOVELMBTANM HWTANM 社員名称
C Z-ADDHJNOKI HWNOKI 納期日
C Z-ADDHJUGK HWJUGK 受注合計額
C MOVELHJBTKO HWBTKO 備考
C
C WRITEHWR00
C ENDSR
C*-----*
C* SBMSG メッセージ取得 *
C*-----*
C SBMSG BEGSR
C
C MJMSGC MOVEL*BLANK PPMG 92
C CHAINMJR00
C MOVELMJMSGM PPMG
C
C ENDSR
C*-----*
C* SBEND 終了処理
C*-----*
C SBEND BEGSR
C<<< 終了処理を行なう。 >>
C SETON LR
C RETRN
C
C ENDSR
C***** データの終わり *****

```

ソース6 照会画面

```

1  /#####
2
3  照会画面
4
5  #####
6  unit ReferenceFrm;
7
8  interface
9
10 uses
11   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
12   Dialogs, Scdcall, Grids, DBGrids, StdCtrls, Mask, Buttons, ExtCtrls, DB,
13   DBTables;
14
15 type
16   TfrmReference = class(TForm)
17     Panel1: TPanel;
18     Label2: TLabel;
19     Label3: TLabel;
20     Label1: TLabel;
21     btnSelect: TBitBtn;
22     meJUDT_F: TMaskEdit;
23     meJUDT_T: TMaskEdit;
24     edTRCD: TEdit;
25     dbgList: TDBGrid;
26     pnlBottom: TPanel;
27     bbtnClose: TBitBtn;
28     cal4D4Q010C: TCall400;
29     cal4D4Q011C: TCall400;
30     tbID4Q010: TTable;
31     tbID4Q010HWJGCD: TIntegerField;
32     tbID4Q010HWJUNO: TIntegerField;
33     tbID4Q010HWJUDT: TIntegerField;
34     tbID4Q010HWTRRK: TStringField;
35     tbID4Q010HWTANM: TStringField;
36     tbID4Q010HWNOKI: TIntegerField;
37     tbID4Q010HWJUGK: TIntegerField;
38     tbID4Q010HWTKCD: TIntegerField;
39     tbID4Q010HWTACD: TIntegerField;
40     tbID4Q010HWBIKO: TStringField;
41     dsD4Q010: TDataSource;
42     btnClear: TButton;
43     procedure FormShow(Sender: TObject);
44     procedure bbtnCloseClick(Sender: TObject);
45     procedure btnClearClick(Sender: TObject);
46     procedure btnSelectClick(Sender: TObject);
47   private
48     { Private 宣言 }
49   public
50     { Public 宣言 }
51   end;
52
53 var
54   frmReference: TfrmReference;
55
56 implementation
57
58 uses MenuFrm;
59
60 {$R *.dfm}
61
62 /#####
63 目的: 画面表示時
64 引数:
65 戻値:
66 /#####
67 procedure TfrmReference.FormShow(Sender: TObject);
68 begin
69   inherited;
70   // QTEMPヘワークファイルを作成
71   cal4D4Q011C.Execute;
72
73   // 初期フォーカスを受注日
74   meJUDT_F.SetFocus;
75 end;
76

```

【図X】①
OnShowイベントで初期設定

-
-
-

```

76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

【図X】②
入力値のチェック

【図X】②
CLプログラムの呼び出し

【図X】⑥
画面に表示する

照会画面
「条件クリア」ボタン押下時

