Montando o Código:

Neste artigo vou abordar um pequeno problema que temos normalmente com nossos usuários, a criação de usuários e senhas para Interbase e Firebird, a demanda de trabalho é muito grande pois para criarmos usuários e senhas, temos que entrar no IB Console, logar-se criar, sair e tudo mais.

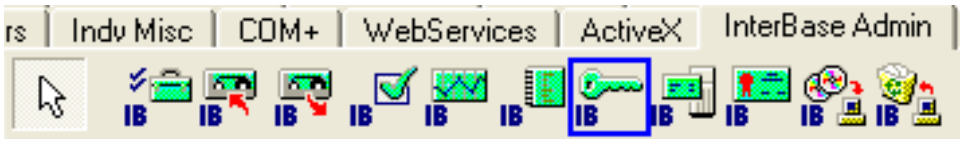
O programa abaixo descrito vai nos servir de base também para criar usuários automaticamente, no momento da instalação do seu sistema.Acima temos a tela de desenvolvimento do sistema, os componentes a serem utilizados pelo User Manager 1.0, serão:

Componente	Nome	Função
RadioGroup	Rg_servidor	Conecta e Desconecta o servidor
IBsecurityServer	ibssSenhas	Pegar os usuários do FB/IB, para manipulá-los, além de criar, apaga os usuários.
BitBtn	btnCriar	Pega o valor dos campos EDITs, e cria os usuários no firebird.
	btnExcluir	Apaga o usuário selecionado na CheckListBox (clbUsers).
CheckListBox	clbUsers	Demonstra os usuários existentes no Servidor.
StatusBar	stbStatus	Exibe os Hints, do sistema, e ainda se o servidor está conectado ou desconectado.
Edit	edtUsuario	Nome do novo Usuário a ser criado
	edtSenha	Senha do novo usuário
Label	lblUsuario	Label do edtUsuario
	lblSenha	Label da Senha
GroupBox	gpbUsers	Agrupa a lista de usuários clbUsers, e o botão excluir usuários
Image	imgAbout	Chama o form_about

Acima temos a lista dos componentes utilizados pelo sistema, com uma prévia das funções que efetuarão, abaixo começarei a descrever como montar o sistema, o primeiro passo é monta um form de acordo com a figura abaixo.

Depois de Posicionados os componentes vamos passar as funcionalidades de acordo com a ordem que foram colocados no Form.

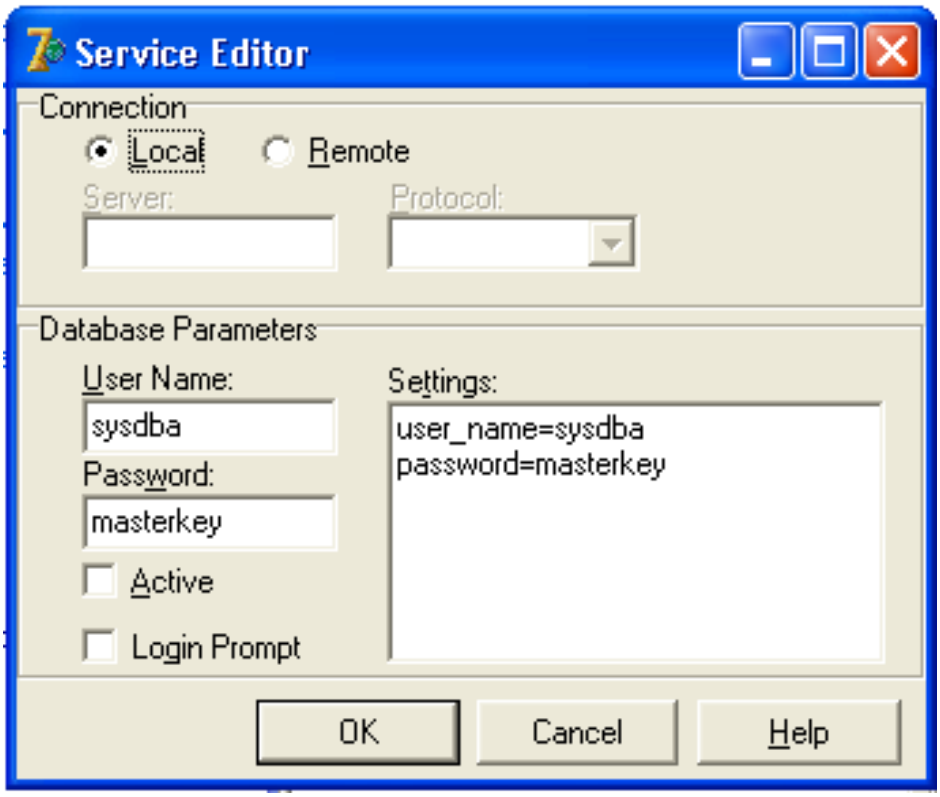
O Primeiro passo para começarmos o nosso sistema é configurar o Componente ibssSenhas (IBsecurityServer), podemos encontrar este componente na paleta Interbase Admin.



Depois de Inserir o componente, vamos completar suas propriedades:

Propriedade	Valor
Name	ibssSenhas
Params	*
Login Prompt	False

* Para preencher a propriedade params do componente, vamos dar um duplo clique no componente ibssSenhas, quando fizermos isso ele vai apresentar uma janela que deverá ser preenchida como abaixo:



Depois de preencher, vamos testar a conexão com o Servidor, para isso basta colocar a propriedade encontrada no Object Inspector, **Active** para **True**, depois de testado, coloque novamente para False.

Observação: se verificar na janela acima não existe a propriedade DATABASE, normalmente encontrada nos componentes do FB/IB, este componente manipula o ISC4.GDB, arquivo de configuração do FB/IB, e não o banco da aplicação.

Vamos agora criar a função que envia o Hint para a stbStatus (StatusBar) da aplicação, no envento on-create do form coloque a seguinte função:

```
procedure TfrmMain.FormCreate(Sender: TObject);
begin
    with Application do begin
        HintColor := clBtnFace;
        HintPause := 200;
        OnHint := ShowHints;
    end;
end;
```

Para que funcione corretamente precisamos declarar a procedure ShowHints, antes de declarar declare a procedure na seção private da classe.

```
private
   procedure ShowHints(Sender: TObject);
   { Private declarations }
public
   { Public declarations }
end;
```

Depois de declarada a procedure, vamos colocar o código:

```
procedure TfrmMain.ShowHints(Sender: TObject);
begin
   stbStatus.Panels.Items[0].Text := Application.Hint;
end;
```

No próximo passo colocaremos o código no evento OnClick do rg_Servidor(RagioGroup).

```
procedure TfrmMain.rg_ServidorClick(Sender: TObject);
begin
try
    case RG_servidor.itemIndex of // Conecta o Servidor
    0: ibssSenhas.Active := True;
    1: ibssSenhas.Active := False;
    end;
    if ibssSenhas.Active then begin
      stbStatus.Panels[1].Text := '  Conectado';
    end else begin
      stbStatus.Panels[1].Text := ' Desconectado';
    end;
```

```
      btnCria.Enabled := ((ibssSenhas.Active) and (edtUsuario.Text <> '')
                          and (edtSenha.Text <> ''));
    finally
      if ibssSenhas.Active then
        PegaUsers
      else clbUsers.clear;
    end;
    end;
```

Repare que na procedure anterior existe um evento chamado **PegaUsers**, este método, vai buscar os nomes dos usuários para demonstrá-los no clbUsers (CheckListBox). Para isso devemos declarar outro procedimento na seção private da classe:

```
    private
      procedure ShowHints(Sender: TObject);
      procedure PegaUsers;
      { Private declarations }
    public
      { Public declarations }
    end;
```

Agora vamos colocar o código na procedure, um bom atalho para criar a declaração da procedure, é pressionar Ctrl+Shit+C, que a cria automaticamente:

```
    procedure TfrmMain.PegaUsers;
    var i : integer;
    begin
      ibssSenhas.DisplayUsers;
      clbUsers.clear;
      for i := 0 to ibssSenhas.UserInfoCount -1 do
        clbUsers.Items.Add(ibssSenhas.UserInfo[i].UserName);
    end;
```

No evento OnChange dos Edits (edtUsuario, edtSenha), vamos colocar o código que ficará responsável por ativar e desativar o botão criar;

```
    procedure TfrmMain.edtUsuarioChange(Sender: TObject);
    begin
      btnCria.Enabled := ((ibssSenhas.Active) and (edtUsuario.Text <> '')
                          and (edtSenha.Text <> ''));
    end;
```

Agora vamos colocar no evento OnClick do clbUsers (CheckListBox), para controlarmos o número de usuários estão selecionados, pois podemos permitir apenas um por vez, e também o código que ativa e desativa o btnExcluir (BitBtn).

```
procedure TfrmMain.clbUsersClick(Sender: TObject);
var i, j : Integer;
begin
  j := 0;
  j := clbUsers.ItemIndex;
  for i := 0 to clbUsers.Items.Count - 1 do begin
    if j <> i then
      clbUsers.checked[i] := false;
  end;
  btnExcluir.Enabled := clbUsers.Checked[j];
end;
```

Criamos agora no evento OnClick do BtnCriar, o método para criar usuários.

```
procedure TfrmMain.btnCriaClick(Sender: TObject);
begin
 try
    ibssSenhas.Active := false;
    ibssSenhas.Active := true;
    btnCria.enabled := false;
    with ibssSenhas do begin
      UserName := edtUsuario.Text;
      Password := edtSenha.Text;
      AddUser;
    end;
    edtUsuario.clear;
    edtSenha.clear;
 finally
    ibssSenhas.Active := false;
    ibssSenhas.Active := true;
    PegaUsers;
 end;
end;
```

Agora basta criarmos o método que apaga o usuário através do btnExcluir (BitBtn):

```
procedure TfrmMain.btnExcluirClick(Sender: TObject);
begin
 btnExcluir.enabled := false;
 with ibssSenhas do begin
   if clbUsers.Checked[clbUsers.ItemIndex] then begin
      UserName := clbUsers.Items.Strings[clbUsers.ItemIndex];
      if MessageDlg('Deseja excluir este usuário?',
        mtConfirmation, [mbYes, mbNo], 0) = mrYes then begin
          DeleteUser;
          clbUsers.DeleteSelected;
      end else clbUsers.Checked[clbUsers.ItemIndex] := false;
   end  else  ShowMessage('É necessário escolher um Usuário para
                                        excluir!')
 end;
end;
```

Com Isso encerramos esta pequena aplicação, fico a disposição quaisquer duvidas que tenham ficado, as propriedades, Hint, Name, Color, Caption, e o próprio formato da aplicação, ficam a cargo do programador, pois como todos sabemos o que importa é a “alma” do sistema, e não a sua cara.

Sou programador de Delphi há 2 anos, crio aplicações profissionais para clínicas de Diálise.

Disponibilizado no site www.firebase.com.br
Aplicação e código source disponíveis em :
http://www.firebase.com.br/fb/imgdocs/userman.zip