procedure TForm1.BtnSaveClick(Sender: TObject);

var

items: integer;

begin

if RadioGreek.Checked = true then

SaveGreek

else

SaveLatin;

Cmb\_Item.SetFocus;

PnlSave.Visible := false;

BtnSave.Enabled := false;

BtnIns.Enabled := false;

BtnUpdate.Enabled := false;

Btn\_Search.Enabled := true;

EdtSearch.Enabled := true;

BtnSelfTest.Enabled := true;

end;

procedure TForm1.BtnSaveKeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

begin

if key = vk\_escape then

PressEsc;

end;

Procedure Visibilities\_Insert\_Update\_Clicked;

begin

if (Form1.Cmb\_Item.Text = '') or (Form1.RichDescr.Text = '') then

exit;

Form1.BtnIns.Enabled := false;

Form1.BtnUpdate.Enabled := false;

Form1.BtnSelfTest.Enabled := false;

Form1.Btn\_Search.Enabled := false;

Form1.EdtSearch.Clear;

Form1.EdtSearch.Enabled := false;

if CurrentItem < TotalItems then

Form1.BtnDel.Enabled := true

else

Form1.BtnDel.Enabled := false;

Form1.PnlSave.Visible := true;

Form1.BtnSave.Enabled := true;

Form1.RadioGreek.Enabled := true;

Form1.RadioLatin.Enabled := true;

end;

procedure TForm1.BtnInsClick(Sender: TObject);

var

ins\_counter, CheckIfExists: integer;

begin

if (length(Cmb\_Item.Text) = 0) or (length(RichDescr.Text) = 2) then //RichDescr has 'empty' length of 2

begin

PressEsc;

exit;

end;

if RadioGreek.Checked = true then //check for duplicate entries .. exit if so.

begin

for CheckIfExists := 1 to TotalItems do

if Cmb\_Item.Text = Arr\_Items[CheckIfExists,1] then

begin

MessageDlg('Duplicate found.', mtError, [mbOK],0);

Cmb\_Item.ItemIndex := Cmb\_Item.Items.IndexOf((Cmb\_Item.Text));

RichDescr.Clear;

RichDescr.Lines.Add(Arr\_Items[Cmb\_Item.ItemIndex+1,2]);

Heading.Caption := Cmb\_Item.Text;

Visibilities;

if CurrentItem = TotalItems then

BtnDel.Enabled := false;

Cmb\_Item.SetFocus;

exit;

end;

end

else

begin

for CheckIfExists := 1 to TotalItems do

if Cmb\_Item.Text = Arr\_Items\_Latin[CheckIfExists,1] then

begin

MessageDlg('Duplicate found.', mtError, [mbOK],0);

Cmb\_Item.ItemIndex := Cmb\_Item.Items.IndexOf((Cmb\_Item.Text));

RichDescr.Clear;

RichDescr.Lines.Add(Arr\_Items\_Latin[Cmb\_Item.ItemIndex+1,2]);

Heading.Caption := Cmb\_Item.Text;

Visibilities;

if CurrentItem = TotalItems then

BtnDel.Enabled := false;

Cmb\_Item.SetFocus;

exit;

end;

end;

if CurrentItem = 0 then

CurrentItem := 1; //fix on first item only (see also FormCreate)

if RadioGreek.Checked = true then

begin

for Ins\_Counter := TotalItems downto CurrentItem do

begin

Arr\_Items[Ins\_Counter+1,1] := Arr\_Items[Ins\_Counter,1];

Arr\_Items[Ins\_Counter+1,2] := Arr\_Items[Ins\_Counter,2];

end;

Arr\_Items[CurrentItem,1] := Cmb\_Item.Text;

RichDescr.Text := copy(RichDescr.Text,1,Length(RichDescr.Text)-2); //remove faulty Enter

Arr\_Items[CurrentItem,2] := RichDescr.Text;

end

else

begin

for Ins\_Counter := TotalItems downto CurrentItem do

begin

Arr\_Items\_Latin[Ins\_Counter+1,1] := Arr\_Items\_Latin[Ins\_Counter,1];

Arr\_Items\_Latin[Ins\_Counter+1,2] := Arr\_Items\_Latin[Ins\_Counter,2];

end;

Arr\_Items\_Latin[CurrentItem,1] := Cmb\_Item.Text;

NewLatinInsertedItem := Cmb\_Item.Text;

RichDescr.Text := copy(RichDescr.Text,1,Length(RichDescr.Text)-2); //remove faulty Enter

Arr\_Items\_Latin[CurrentItem,2] := RichDescr.Text;

end;

TotalItems := TotalItems + 1;

InsertUpdateDelClicked := true;

Saved := false;

FillCombo;

Visibilities\_Insert\_Update\_Clicked;

SearchStringLabel;

Cmb\_Item.SetFocus;

end;

procedure TForm1.BtnInsKeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

begin

if key = vk\_escape then

PressEsc;

end;

procedure TForm1.BtnCloseClick(Sender: TObject);

begin

if BtnSave.Enabled = true then

begin

if RadioGreek.Checked = true then

begin

if MessageDlg('Save file GkVocab.txt before closing?', mtWarning, [mbYes, mbNo],0) = mrYes then

SaveGreek;

end

else

if MessageDlg('Save file LatinVocab.txt before closing?', mtWarning, [mbYes, mbNo],0) = mrYes then

SaveLatin;

Application.Terminate;

end

else

application.Terminate;

end;

procedure TForm1.BtnCloseKeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

begin

if key = vk\_escape then

PressEsc;

end;

procedure TForm1.BtnCustomiseKeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

begin

if key = vk\_escape then

PressEsc;

end;

procedure TForm1.BtnDelClick(Sender: TObject);

var

DelItem: integer;

begin

if MessageDlg('Delete item?', mtWarning, [mbOK, mbCancel],0) = mrOK then

begin

if CurrentItem = 0 then

CurrentItem := 1; // fix only for first item;

if RadioGreek.Checked = true then

begin

for DelItem := CurrentItem to TotalItems-1 do

begin

Arr\_Items[DelItem,1] := Arr\_Items[DelItem+1,1];

Arr\_Items[DelItem,2] := Arr\_Items[DelItem+1,2];

end;

Arr\_Items[TotalItems,1] := '';

Arr\_Items[TotalItems,2] := '';

end

else

begin

for DelItem := CurrentItem to TotalItems-1 do

begin

Arr\_Items\_Latin[DelItem,1] := Arr\_Items\_Latin[DelItem+1,1];

Arr\_Items\_Latin[DelItem,2] := Arr\_Items\_Latin[DelItem+1,2];

end;

Arr\_Items\_Latin[TotalItems,1] := '';

Arr\_Items\_Latin[TotalItems,2] := '';

end;

TotalItems := TotalItems - 1;

if (TotalItems = 1) or (Cmb\_Item.ItemIndex = TotalItems - 1) then

begin

BtnDel.Enabled := false;

BtnUpdate.Enabled := false;

end;

PnlSave.Visible := true;

BtnSave.Enabled := true;

BtnSelfTest.Enabled := false;

InsertUpdateDelClicked := True;

Saved := false;

FillCombo;

SearchStringLabel;

ResetRandomArray;

RandomItemsTillNow := 0;

Cmb\_Item.SetFocus;

end

else

Cmb\_Item.SetFocus;

end;