

## KODLAR

### Dosya Kopyalama (1)

```

Procedure FileCopy( Const sourcefilename, targetfilename: String );
Var
S, T: TFileStream;
Begin
S := TFileStream.Create( sourcefilename, fmOpenRead );
try
T := TFileStream.Create( targetfilename,
fmOpenWrite or fmCreate );
try
T.CopyFrom(S, S.Size ) ;
finally
T.Free;
end;
finally
S.Free;
end;
End;

```

### Dosya Kopyalama (2)

```

procedure FileCopy(const FromFile, ToFile: string);
var
FromF, ToF: file;
NumRead, NumWritten: Word;
Buf: array[1..2048] of Char;
begin
AssignFile(FromF, FromFile);
Reset(FromF, 1); { Record size = 1 }
AssignFile(ToF, ToFile); { Open output file }
Rewrite(ToF, 1); { Record size = 1 }
repeat
BlockRead(FromF, Buf, SizeOf(Buf), NumRead);
BlockWrite(ToF, Buf, NumRead, NumWritten);
until (NumRead = 0) or (NumWritten <> NumRead);
CloseFile(FromF);
CloseFile(ToF);
end;

```

### Dosya Kopyalama (3)

```

procedure CopyFile(FromFileName, ToFileName: string);
var

```

```

FromFile, ToFile: File;
begin
AssignFile(FromFile, FromFileName); { Assign FromFile to FromFileName }
AssignFile(ToFile, ToFileName); { Assign ToFile to ToFileName }
Reset(FromFile); { Open file for input }
try
Rewrite(ToFile); { Create file for output }
try
if LZCopy(TFileRec(FromFile).Handle, TFileRec(ToFile).Handle) < 0
then
raise EInOutError.Create('Error using LZCopy')
finally
CloseFile(ToFile); { Close ToFile }
end;
finally
CloseFile(FromFile); { Close FromFile }
end;
end;

```

### **Directory Adi degistirme**

SysUtils unitesinin içindeki RenameFile function bu isi görmektedir.

### **Ttreeview componentinin durumunu kaydetme ve gösterme**

Kaydetmek için

```
TreeView.SaveToFile('Dosya.adi');
```

### **Dosyayi açmak için**

```
TreeView.LoadFromFile('Dosya.adi');
```

### **Dosyayi yalnızca okumak (read only) seklinde açma**

```

AssignFile(F, Dosya);
FileMode := 0; (read only modunda açmak)
Reset(F);
.
.
.
CloseFile(F);

```

### **Açılan dosyanın tarih ve zamanını ayarlamak**

```

var
f: file;
begin
Assign(f, DirInfo.Name);

```

```
Reset(f);
SetFTime(f, Time);
Close(f);
end;
```

### **Deltree**

```
{ $I- } { $I+ }
```

```
procedure delTree (DirName: string);
var
  FileSearch: SearchRec;
begin

  chDir (DirName);
  FindFirst ('*.*', Directory, FileSearch);
  while (DosError = 0) do begin
    if (FileSearch.name <> '.') AND (FileSearch.name <> '..') AND
      ( (FileSearch.attr AND Directory) <> 0)
    then begin
      if DirName[length(DirName)] = '\' then
        delTree (DirName+FileSearch.Name)
      else
        delTree (DirName+'\'+FileSearch.Name);
      ChDir (DirName);
    end;
    FindNext (FileSearch)
  end;
```

```
  FindFirst ('*.*', AnyFile, FileSearch);
  while (DosError = 0) do begin
    if (FileSearch.name <> '.') AND (FileSearch.name <> '..') then
      Remove (workdir);
    end;
    FindNext (FileSearch)
  end;
  rmDir (DirName)
end;
```

### **Dbgrid'de (Ctrl-Del diyince) dosya silmesini engelleme**

```
if (ssctrl in shift) and (key=vk_delete) then
begin
  key:=0;
end;
```

### **String'i renk'e renk'i stringe çevirme**

```
Uses graphics;
```

```
form1.Color:=stringtcolor('121');  
label1.caption:= ColorToString(form1.color);
```

### **Mouse'un yerini degistirmek**

```
randomize;  
SetCursorPos(random(100),random(100));
```

### **ComboBox'in asagiya listelemesinin farkli bir yolu (DropComboBox)**

```
SendMessage(ComboBox1.handle , 1039, 1, 0);
```

### **Basligi Gizlemek ve Göstermek**

```
Procedure HideTitlebar;  
Var  
Save : LongInt;  
Begin  
If form1.BorderStyle=bsNone then Exit;  
Save:=GetWindowLong(form1.Handle,gwl_Style);  
If (Save and ws_Caption)=ws_Caption then Begin  
Case form1.BorderStyle of  
bsSingle,  
bsSizeable : SetWindowLong(form1.Handle,gwl_Style,Save and  
(Not(ws_Caption)) or ws_border);  
bsDialog : SetWindowLong(form1.Handle,gwl_Style,Save and  
(Not(ws_Caption)) or ds_modalframe or ws_dlgframe);  
End;  
form1.Height:=form1.Height-getSystemMetrics(sm_cyCaption);  
form1.Refresh;  
End;  
end;  
  
Procedure ShowTitlebar;  
Var  
Save : LongInt;  
begin  
If form1.BorderStyle=bsNone then Exit;  
Save:=GetWindowLong(form1.Handle,gwl_Style);  
If (Save and ws_Caption)<>ws_Caption then Begin  
Case form1.BorderStyle of  
bsSingle,  
bsSizeable : SetWindowLong(form1.Handle,gwl_Style,Save or ws_Caption or  
ws_border);  
bsDialog : SetWindowLong(form1.Handle,gwl_Style,Save or ws_Caption or  
ds_modalframe or ws_dlgframe);  
End;  
form1.Height:=form1.Height+getSystemMetrics(sm_cyCaption);  
form1.Refresh;  
End;
```

end;

### **GradientFill (eyimli renk) kullanımı**

Önce uses bölümüne chart ekleyin.

Formun üstüne bir tane image componenti yerleştirin ve formun create olayına aşağıdaki kodu yazın.

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  GradientFill(Image1.Canvas,Image1.ClientRect,clred,clblue,false);
end;
```

### **Küçük bir kolaylık ctrl-space**

Delphide esitlik kurucaginiz zaman ctrl-space'e bastiginizda verebileceginiz degerleri göstermektedir.

Label1.caption:= (burada ctrl-space bastiginizda verebileceginiz degerleri görürsünüz)

### **Dbf dosyasini pack etmek (silinen dosyalari diskten silmek)**

```
Table1.Exclusive := True;
Table1.Active := True;
Error := DbfPackTable(Table1.DBHandle, Table1.Handle, nil, szdBASE, True);
Table1.Active := False;
Table1.Exclusive := False;
```

### **BDE' nin yüklü olup olmadigini anlamak**

HKEY\_LOCAL\_MACHINE\Software\Borland\Database Engine bölümü bde'ye ayrılmıştır ve yüklü olup olmadigini aşağıdaki degerleri kontrol ederek bulabilirsiniz.

```
DLLPATH
CONFIGFILE01
```

### **Clipboard'a kopyalama ve yapistirma (copy,past)**

```
procedure CopyButtonClick(Sender: TObject);
begin
  If ActiveControl is TMemo then TMemo(ActiveControl).CopyToClipboard;
  If ActiveControl is TDBMemo then TDBMemo(ActiveControl).CopyToClipboard;
  If ActiveControl is TEdit then TEdit(ActiveControl).CopyToClipboard;
  If ActiveControl is TDBedit then TDBedit(ActiveControl).CopyToClipboard;
end;
```

```
procedure PasteButtonClick(Sender: TObject);
begin
  If ActiveControl is TMemo then TMemo(ActiveControl).PasteFromClipboard;
  If ActiveControl is TDBMemo then TDBMemo(ActiveControl).PasteFromClipboard;
```

```
If ActiveControl is TEdit then TEdit(ActiveControl).PasteFromClipboard;  
If ActiveControl is TDBedit then TDBedit(ActiveControl).PasteFromClipboard;  
end;
```

### **TStringgrid'i kaydetme ve açma**

```
Procedure SaveGrid;  
var  
f:textfile;  
x,y:integer;  
begin  
assignfile (f,'Filename');  
rewrite (f);  
writeln (f,stringgrid.colcount);  
writeln (f,stringgrid.rowcount);  
For X:=0 to stringgrid.colcount-1 do  
For y:=0 to stringgrid.rowcount-1 do  
writeln (F, stringgrid.cells[x,y]);  
closefile (f);  
end;
```

```
Procedure LoadGrid;  
var  
f:textfile;  
temp,x,y:integer;  
tempstr:string;  
begin  
assignfile (f,'Filename');  
reset (f);  
readln (f,temp);  
stringgrid.colcount:=temp;  
readln (f,temp);  
stringgrid.rowcount:=temp;  
For X:=0 to stringgrid.colcount-1 do  
For y:=0 to stringgrid.rowcount-1 do begin  
readln (F, tempstr);  
stringgrid.cells[x,y]:=tempstr;  
end;  
closefile (f);
```

### **Avi dosyasının görünüm alanini seçilen panele esitleme**

```
begin  
with MediaPlayer1 do begin  
DeviceType := dtAutoSelect;  
visible := false;  
FileName := InputBox('AVI', 'Enter AVI file name', 'c:\windows\borland.avi');  
display := panel1;  
open;  
DisplayRect := rect(0, 0, panel1.width, panel1.height); { This is it!}
```

```
rewind;  
play;  
end;  
end;
```

### **Windows'a çizgi çizmenin farklı bir yolu**

```
procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);  
var dc:hdc;  
begin  
dc:=getdc(0);  
Windows.LineTo(dc,x,y);  
end;  
  
end.
```

### **Memo componentindeki cursorun pozisyonunu bulma**

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
Row, Col: Integer;  
begin  
Row := SendMessage(Memo1.Handle, EM_LINEFROMCHAR, Memo1.SelStart, 0);  
Col := Memo1.SelStart - SendMessage(Memo1.Handle, EM_LINEINDEX, Row, 0);  
Label1.caption := 'Row= ' + IntToStr(Row+1) + ' Col= ' + IntToStr(Col+1);  
end;
```

### **Form üstüne dbclick yapilınca maximize (tam ekran) olması**

```
type  
TForm1 = class(TForm)  
Button1: TButton;  
procedure Button1Click(Sender: TObject);  
private  
procedure WMNCHitTest(var M: TWMNCHitTest); message wm_NCHitTest;  
end;  
  
var  
Form1: TForm1;  
implementation  
{$R *.DFM}  
procedure TForm1.WMNCHitTest(var M: TWMNCHitTest);  
begin  
inherited;  
if M.Result = htClient then  
M.Result := htCaption;  
end;  
  
procedure TForm1.Button1Click(Sender: TObject);  
begin
```

```
Close;  
end;  
end.
```

### **Control paneldeki bölümleri açma (cpl dosyalarını açma)**

```
unit open_cpl;  
  
interface  
  
function RunControlPanelApplet( sAppletFileName : string ) : integer;  
  
implementation  
  
uses Windows;  
  
function RunControlPanelApplet( sAppletFileName : string ) : integer;  
begin  
Result := WinExec( PChar('rundll32.exe shell32.dll,'+ 'Control_RunDLL  
' + sAppletFileName), SW_SHOWNORMAL);  
end;  
  
end.
```

access.cpl: Accessibility Properties  
appwiz.cpl: Add/Remove Programs Properties  
desk.cpl: Display Properties  
intl.cpl: Regional Settings Properties  
joy.cpl: Joystick Properties  
main.cpl: Mouse Properties  
mmsys.cpl: Multimedia Properties  
modem.cpl: Modems Properties  
sysdm.cpl: System Properties  
timedate.cpl: Time/Date Properties

### **Kullanılan printer adı**

```
uses Printers;  
function GetDefaultPrinterName : string;  
begin GetDefaultPrinterName := Printer.Printers[ Printer.PrinterIndex ];  
end;
```

### **Cd sürücüsünün seri numarası**

```
function GetDiskVolSerialID( cDriveName : char ) : DWord;  
var dwTemp1, dwTemp2 : DWord;  
begin GetVolumeInformation( PChar( cDriveName + ':' ), Nil, 0, @Result, dwTemp2,  
dwTemp2, Nil, 0 );  
end;
```



```
MessageDlg( 'Serial number: ' + Format( '%X', [ GetDiskVolSerialID( 'E' ) ] ),
mtInformation, [mbOk], 0 );
```

### **Capslock'un durumunu öğrenme**

```
function IsCapsLockOn : boolean;
begin Result := 0 <> (GetKeyState(VK_CAPITAL) and $01);
end;
```

### **Renklerin hex degerlerini bulma**

```
function
TColorToHex( Color : TColor ) : string;
begin
Result := IntToHex( GetRValue( Color ), 2 ) + IntToHex( GetGValue( Color ), 2 ) +
IntToHex( GetBValue( Color ), 2 );
end;
```

### **Cursor'u gizleyip tekrar gösterme**

```
Showcursor(true); //cursoru göster
Showcursor(false); //cursoru gizle
```

### **internete bagli olup olmadiginizi anlamak**

Forma bir tane tcp componenti ve bir tane buton yerlestirdikten sonra  
Buttonnun içine asagidaki yazi yazilir.

```
if TCP1.LocalIp = '0.0.0.0' then
ShowMessage('Bagli degilsiniz!');
```

### **Html Sayfasini printerden cikarma**

```
uses Printers;
var
EskCur: TCursor;
begin
EskCur := Screen.Cursor;
with Printer do begin
BeginDoc;
HTML1.AutoPrint(handle);
Title := HTML1.URL;
EndDoc;
end;
Screen.Cursor := EskCur;
end;
```

### **Menu'ye clicklenirken hangi tusla basildigini algilama (Control,Shift,Alt)**

```

procedure TForm1.Menu1Click(Sender: TObject);
begin
if HiWord(GetKeyState(VK_SHIFT)) <> 0 then
Label1.Caption := 'Shift'
else
if HiWord(GetKeyState(VK_CONTROL)) <> 0 then
Label1.Caption := 'Control'
else
if HiWord(GetKeyState(VK_MENU)) <> 0 then
Label1.Caption := 'Alt'
else
Label1.Caption := 'None';
end;

```

### **Formu taskbardan gizleme**

```
ShowWindow(Application.Handle, SW_HIDE);
```

### **Silinmis Dosyaları gösterme (DBF);**

```

type
TForm1 = class(TForm)
Table1: TTable;
DataSource1: TDataSource;
DBGrid1: TDBGrid;
DBNavigator1: TDBNavigator;
CheckBox1: TCheckBox;
procedure CheckBox1Click(Sender: TObject);
public
procedure ShowDeleted(Table: TTable; ShowDeleted: Boolean);
end;

```

```

var
Form1: TForm1;

```

```
implementation
```

```
uses DBTYPES, DBIERRS, DBIPROCS;
```

```
{ $R *.DFM }
```

```

procedure TForm1.ShowDeleted(Table: TTable; ShowDeleted: Boolean);
var
rslt: DBIResult;
szErrMsg: DBIMSG;
begin
Table.DisableControls;
try
Check(DbSetProp(hDBIObj(Table.Handle), curSOFTDELETEON,
LongInt(ShowDeleted)));

```

```
finally
Table.EnableControls;
end;
Table.Refresh;
end;

procedure TForm1.CheckBox1Click(Sender: TObject);
begin
ShowDeleted(Table1, CheckBox1.Checked);
end;

end.
```

### **Form hakkında kısa bilgi**

```
Form1.Show; {Formu göster}
Form1.Showmodal { Formu showmodal olarak göster (Showmodal iken baska hiçbir forma
ulasilamaz) }
myForm.Hide; {Formmu gizle }
myForm.Free; {Formu sil }
```

### **Formu seffaf hale getirilmesi**

OnCreate olayina Brush.Style:=bsClear; yazmanız yeterli.

### **ESC tusuyla programdan çıkma**

OnKeyPress olayina if Key = #27 then halt; yazın.

### **Bir Resim Dosyası Açıp içine Çizim Yapma**

```
var
Resim: TBitmap;

Resim := TBitmap.Create;
with Resim do
try
Width := 100;
Height := 100;
with Canvas do
begin
Rectangle(0, 0, 100, 100);
MoveTo(0, 0);
LineTo(100, 100);
MoveTo(0, 100);
LineTo(100, 0);
end;
SaveToFile('test.bmp')
finally
```

```
Free;  
end;
```

### **Asla Yazı Kutusundan Çıkamazsin**

```
procedure TForm1.Edit1Exit(Sender: TObject);  
begin  
  showmessage('Asla yazi kutusundan çikamassin');  
  postmessage(handle,WM_NEXTDLGCTL,0,0);  
  postmessage(handle,WM_NEXTDLGCTL,1,0);  
end;
```

### **Formun Boyutlarini Sabitleme**

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;
```

```
type
```

```
TForm1 = class(TForm)
```

```
private
```

```
{ Private declarations }
```

```
public
```

```
procedure WMGetMinMaxInfo( var Message :TWMGetMinMaxInfo ); message  
WM_GETMINMAXINFO;
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```
Form1: TForm1;
```

```
implementation
```

```
{ $R *.DFM }
```

```
procedure TForm1.WMGetMinMaxInfo( var Message :TWMGetMinMaxInfo );  
begin
```

```
  with Message.MinMaxInfo^ do
```

```
  begin
```

```
    ptMaxSize.X := 200;
```

```
    ptMaxSize.Y := 200;
```

```
    ptMaxPosition.X := 99;
```

```
    ptMaxPosition.Y := 99;
```

```
    ptMinTrackSize.X := 100;
```

```
    ptMinTrackSize.Y := 100;
```

```

ptMaxTrackSize.X := 300;
ptMaxTrackSize.Y := 300;
end;
Message.Result := 0;
inherited;
end;

```

end.

### **Windows'a True Type Font Ekleme**

```

uses Registry;
procedure TForm1.Button1Click(Sender: TObject);
var
reg: TRegistry;
b : bool;
begin CopyFile('C:\DOWNLOAD\FP000100.TTF', 'C:\WINDOWS\FONTS\FP000100.TTF',
b);
reg := TRegistry.Create; reg.RootKey := HKEY_LOCAL_MACHINE;
reg.LazyWrite := false; reg.OpenKey('Software\Microsoft\Windows\CurrentVersion\Fonts',
false); reg.WriteString('TESTMICR (TrueType)', 'FP000100.TTF');
reg.CloseKey; reg.free;
AddFontResource('c:\windows\fonts\FP000100.TTF');
SendMessage(HWND_BROADCAST, WM_FONTCHANGE, 0, 0);
RemoveFontResource('c:\windows\fonts\FP000100.TTF');
SendMessage(HWND_BROADCAST, WM_FONTCHANGE, 0, 0);
end;

```

### **Dosya Tasima**

```

MoveFile('C:\Source\sourcefile.txt', 'C:\Dest\destfile.txt');

```

### **Trim Fonksiyonunun açılımı**

```

unit StrFunc;
interface
procedure LTrim(var s : string);
procedure RTrim(var s : string);
;procedure Trim(var s : string);
implementation
{$R *.DFM}
procedure LTrim(var s : string);
begin
while ((length(s) > 0) and (s[1] = #32)) do Delete(s, 1, 1);
end;
procedure RTrim(var s : string);
begin
while ((length(s) > 0) and (s[Length(s)] = #32)) do Delete(s, Length(s), 1);
end;
procedure Trim(var s : string);

```

```
begin
LTrim(s); RTrim(s);
end;
```

### **Screen Capture Programı (Ekran Çalma)**

Bir tane image componenti eklemek zorundasınız.

```
procedure TForm1.FormCreate(Sender: TObject);
var
DCDesk: HDC;

begin
DCDesk:=GetWindowDC(GetDesktopWindow);
BitBlt(Image1.Canvas.Handle, 0, 0, Screen.Width, Screen.Height,DCDesk, 0, 0,SRCCOPY);
ReleaseDC(GetDesktopWindow, DCDesk);
end;
```

### **Saga yada Sola hizalama function'u**

```
function hiza(sagsol,yer:string;sayi:integer):string;
var
i,bosluk,uzunluk:integer;
begin
trim(yer); //Bosluklari temizle
uzunluk:=length(yer); //Verilen stringin uzunlugunu öğren
bosluk:=sayi-uzunluk; //Verilen sayidan uzunlugu çıkar
if sagsol='sag' then begin //Sagami solami yaslanacagini öğren
for i:=1 to bosluk do begin //Fark kadar saga bosluk ekle
yer:=' '+yer; //Bosluk ekle
end;
result:=yer; //Sonucu bildir
exit;
end;
if sagsol='sol' then
begin
trim(yer);
for i:=1 to bosluk do begin //Fark kadar sola ekle
yer:=yer+' ';
end;
result:=yer; //Sonucu bildir
exit;
end;
```

### **Sayıyı virgüllü yazma functionu**

Bunun için forma birtane label componenti yerlestin ve visible özelligini false yapin.

```
function frmt(yer:string):string;
var
```

```

s:string;
uzunluk:integer;
kalan,k,dgr:integer;
begin
s:='.';
uzunluk:=Length(yer);
dgr:=uzunluk div 3;
kalan:=uzunluk - (dgr*3);
label1.caption:=inttostr(kalan);
if yer='0' then
begin
yer:='0,';
result:=yer;
exit;
end;
if label1.caption<>'0' then
begin
insert(s,yer,kalan+1);
for k:=1 to dgr do begin
if k=dgr then begin
insert(', ',yer,(kalan +(k*3)+k+1));
result:=yer;
exit;
end;
insert(s,yer,(kalan +(k*3)+k+1));
end;
end;
end;

```

### **Comp degerindeki bir sayiyi string yapma**

```

function Comp2Str(N: Comp): String;
var
TS: String;
begin
Str(N:0:0, TS);Comp2Str := TS;
end;

```

### **Dosya açilis yollari**

Windows programinin açilisi

```
WinExec('C:\WINDOWS\notepad.exe', SW_SHOWNORMAL);
```

Komut satiri ile birlikte :

```
WinExec('C:\WINDOWS\notepad.exe C:\WINDOWS\win.ini',
SW_SHOWNORMAL);
```

Pop Up dos kutusuyla :

```
WinExec('command.com', SW_SHOWNORMAL);
```

Normal bir dosyayın komutla açılması

```
WinExec('COMMAND.COM /C DIR *.*', SW_SHOWNORMAL);
```

### **Dekstop'i refresh etmek**

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  SendMessage(FindWindow('Progman', 'Program Manager'), WM_COMMAND, $A065, 0);
end;
```

### **Programın windows'la birlikte açılması**

```
uses Registry, {For Win32}
  IniFiles; {For Win16}
{$IFDEF WIN32}
const
  MAX_PATH = 144;
{$ENDIF}
{For Win32}
procedure TForm1.Button1Click(Sender: TObject);
var
  reg: TRegistry;
begin
  reg := TRegistry.Create;
  reg.RootKey := HKEY_LOCAL_MACHINE;
  reg.LazyWrite := false; reg.OpenKey('Software\Microsoft\Windows\CurrentVersion\Run',
  false); reg.WriteString('My App', Application.ExeName);
  reg.CloseKey;
  reg.Free;
end;
```

### **Formun ara çubuğunu gizleme**

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  SetWindowLong(Form1.Handle, GWL_STYLE, GetWindowLong(Handle, GWL_STYLE) and
  not WS_CAPTION);
  Height := ClientHeight;
end;
```

### **Flash icon**

```
var Flash : bool;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  FlashWindow(Form1.Handle, Flash);
  FlashWindow(Application.Handle, Flash); Flash := not Flash;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
```



```
Flash := False;  
end;
```

### **Taskbar'in handle sayisi**

```
hTaskbar := FindWindow('Shell_TrayWnd', Nil );
```

### **Exe yada Dll'deki iconların alınması**

```
Uses shellapi;  
procedure TForm1.FormCreate(Sender: TObject);  
var TheIcon: TIcon;  
begin TheIcon := TIcon.Create;  
TheIcon.Handle := ExtractIcon(hInstance,'C:\dosya\dosya.EXE',0);  
image1.picture.Icon:=Theicon;  
end;
```

### **Windows ekranına çizim yapma**

```
procedure TForm1.Button1Click(Sender: TObject);  
var dc : hdc;  
begin dc := GetDc(0);  
MoveToEx(Dc, 0, 0, nil);  
LineTo(Dc, 300, 300);  
ReleaseDc(0, Dc);  
end;
```

### **Edit1'deki yazıyı edit2'ye GetTextBuf yoluyla aktarma**

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
Buffer: PChar;  
Size: Byte;  
begin  
Size := Edit1.GetTextLen; {Get length of string in Edit1}  
Inc(Size); {Add room for null character}  
GetMem(Buffer, Size); {Creates Buffer dynamic variable}  
Edit1.GetTextBuf(Buffer,Size); {Puts Edit1.Text into Buffer}  
Edit2.Text := StrPas(Buffer); {Converts Buffer to a Pascal-style string}  
FreeMem(Buffer, Size); {Frees memory allocated to Buffer}  
  
end;
```

### **Form1'in aynısını printerden çıkartma**

```
Form1.Print;
```

O anki sürücünün kapasitesini ve sürücüdeki boş yer miktarını bulmak;

DiskFree(0) //o anki sürücüdeki bos yer miktarini byte cinsinden döndürür.  
DiskSize(0) //o anki sürücünün kapasitesini byte cinsinden döndürür.  
DiskSize(0) div 1024 //o anki sürücünün kapasitesini KB cinsinden döndürür.

Bir menü öğesinin enabled özelligini false yapmak;

```
mainmenu1.items[0].items[1].enabled:=False;
```

Listbox veya Combobox'ta seçili bir veya birden fazla öğeyi seçilmemis duruma getirmek için;

```
Listbox1.itemindex:=-1;
```

Listbox, Combobox ve Memo'ya bir seferde ekleme yapmak;

```
Listbox1.items.SetText('aaa'#13'bbb'#13'ccc');  
Memo1.Lines.SetText('aaa'#13'bbb'#13'ccc');
```

Harddiskin seri numarasinin bulunmasi;

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
    VolumeSerialNumber : DWORD;  
    MaximumComponentLength : DWORD;  
    FileSystemFlags : DWORD;  
    SerialNumber : string;  
begin  
  
    GetVolumeInformation('C:\',nil,0,@VolumeSerialNumber,MaximumComponentLength  
    ,FileSystemFlags,nil,0);  
    SerialNumber := IntToHex(HiWord(VolumeSerialNumber), 4) + '-'  
+IntToHex(LoWord(VolumeSerialNumber), 4);  
    Memo1.Lines.Add(SerialNumber);  
end;
```

Bir string'in basindaki ve sonundaki bosluklari atmak için;

```
Trim(string)  
TrimLeft (string) //stringin sadece basindaki bosluklari atmak için  
TrimRight (string) //stringin sadece sonundaki bosluklari atmak için
```

sifreli bir table için programin sifre istememesi için;

Table'in Active özelligini False yapin ve Form'un OnCreate olayina asagidaki kodu ekleyin

```
Session.AddPassword('sifre');  
Table1.Active:=True;
```

Pencereyi minimize etmek;

```
Application.Minimize; //taskbar'a minimize
CloseWindow(handle)
WindowState := wsMinimized;
```

Windows'u kapatmak veya yeniden baslatmak(reboot);

```
var
    i:dword;
begin
    ExitWindowsEx(EWX_SHUTDOWN); //yeniden baslatmak için EWX_REBOOT
end;
```

Help menüsünden About kısmına girin. Alt tusuna basili tutarak TEAM veya DEVELOPERS yazın. Delphi'yi geliştirenlerin isimlerini görebilirsiniz. Database Desktop'tada Help menüsünden About kısmına girin ve delphi yazın.

ico'dan bmp'ye çevirme;

```
var
    Icon : TIcon;
    Bitmap : TBitmap;
begin
    Icon := TIcon.Create;
    Bitmap := TBitmap.Create;
    Icon.LoadFromFile('c:\picture.ico');
    Bitmap.Width := Icon.Width;
    Bitmap.Height := Icon.Height;
    Bitmap.Canvas.Draw(0, 0, Icon );
    Bitmap.SaveToFile('c:\picture.bmp');
    Icon.Free;
    Bitmap.Free;
end;
```

CD-Rom sürücüyü açmak ve kapamak;

uses kısmına MMSystem unitini ekleyin.

```
mciSendString('Set cdaudio door open wait', nil, 0, handle); //aç
mciSendString('Set cdaudio door closed wait', nil, 0, handle); //kapa
```

CapsLock ve Numlock tuslarini açıp-kapama;

```
procedure TMyForm.Button1Click(Sender: TObject);
Var
    KeyState : TKeyboardState;
begin
    GetKeyboardState(KeyState);
    if (KeyState[VK_CAPITAL] = 0) then
        KeyState[VK_CAPITAL] := 1
    else
        KeyState[VK_CAPITAL] := 0;
    SetKeyboardState(KeyState);
end;
```

Numlock tusu için VK\_CAPITAL yerine VK\_NUMLOCK yazınız.

## Menü'ye bitmap (resim) ekleme;

```
procedure TForm1.FormCreate(Sender: TObject);
var
    Bmp1 : TPicture;
begin
    Bmp1 := TPicture.Create;
    Bmp1.LoadFromFile('c:\deneme\turkey.bmp');
    SetMenuItemBitmaps( deneme1.Handle,
                        0,
                        MF_BYPOSITION,
                        Bmp1.Bitmap.Handle,
                        Bmp1.Bitmap.Handle);
end;
```

## Alt + Tab ve Ctrl + Esc tuslarinin kullanilmaz hale getirilmesi;

```
var
    OldVal : LongInt;
begin
    SystemParametersInfo (97, Word (True), @OldVal, 0)
    //Word(False) ile kullanirsaniz tuslari tekrar kullanabilirsiniz.
```

## Windows ve System klasörlerinin bulunmasi

```
procedure TForm1.Button1Click(Sender: TObject);
var
    a : Array[0..144] of char;
begin
    GetWindowsDirectory(a, sizeof(a));
    ShowMessage(StrPas(a));
    GetSystemDirectory(a, sizeof(a));
    ShowMessage(StrPas(a));
end;
```

---

## Speakerdan Beep sesi çıkartma

```
MessageBeep(word(-1));
```

---

## Belgeler menüsüne bir dosya ekleme

### uses kısmına ShIOBJ unitini ekleyin;

```
procedure TForm1.Button1Click(Sender: TObject);
var
    s : string;
begin
    s := 'C:\Download\deneme.html';
    SHAddToRecentDocs(SHARD_PATH, pChar(s));
end;
```

---

## Belgeler menüsünü temizleme

uses kısmına ShlOBJ unitini ekleyin;

```
SHAddToRecentDocs(SHARD_PATH, nil);
```

---

Bir web adresini açma

uses kısmına Shellapi unitini ekleyin;

```
ShellExecute(Handle,  
              'open',  
              'http://emreoz.cjb.net',  
              nil,  
              nil,  
              sw_ShowMaximized);
```

---

Bir DOS programını çalıştırma ve çalışması bitince penceresini kapatma

```
WinExec("command.com /c progdos.exe",sw_ShowNormal); //progdos.exe çalıştırılıyor.
```

```
//eger ikinci parametreyi sw_Hide yaparsanız kullanıcı programın çalıştığını görmez.
```

---

Uygulamanızın Görev Çubuğundaki butonunu gizleme

Uygulamanızın Görev Çubuğundaki butonunu gizlemek için programınızın ana formunun OnCreate olayına aşağıdaki kodu yazın;

```
SetWindowLong(Application.Handle,GWL_EXSTYLE, WS_EX_TOOLWINDOW);
```

---

Ekran koruyucusunu kapatmak ve açmak

```
//kapatmak için  
SystemParametersInfo(SPI_SETSCREENSAVEACTIVE,  
                      0,  
                      nil,  
                      0);  
  
//açmak için  
SystemParametersInfo(SPI_SETSCREENSAVEACTIVE,  
                      1,  
                      nil,  
                      0);
```

---

Alt+F4 tus kombinasyonu ile programın kapanmaması için

Formun OnCreate olayına;

```
KeyPreview := true;
```

Formun OnKeyDown olayina;

```
if ((ssAlt in Shift) and (Key = VK_F4)) then  
    Key := 0;
```

---

Hareketli imleç(animated cursor)leri kullanma

```
procedure TForm1.Button1Click(Sender:TObject);  
var  
    h : THandle;  
begin  
    h := LoadImage(0,  
        'C:\TheWall\Magic.ani',  
        IMAGE_CURSOR,  
        0,  
        0,  
        LR_DEFAULTSIZE or  
        LR_LOADFROMFILE);  
    if h = 0 then ShowMessage('Cursor not loaded') else begin  
        Screen.Cursors[1] := h;  
        Form1.Cursor := 1;  
    end;  
end;
```

---

Windows lisans bilgilerinin (isim ve sirket) bulunmasi

uses kismina Registry unitini ekleyin;

```
procedure TForm1.Button1Click(Sender:TObject);  
var  
    reg: TRegIniFile;  
begin  
    reg := TRegIniFile.create('SOFTWARE\MICROSOFT\MS SETUP (ACME)\');  
    Mem1.Lines.Add(reg.ReadString('USER INFO',  
        'DefName',  
        'Emre ÖZ'));  
    Mem1.Lines.Add(reg.ReadString('USER INFO',  
        'DefCompany',  
        'Bilgisayar Bilimleri Müh.'));  
    reg.free;  
end;
```

---

Çok Satirli ipucu

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
    SpeedButton1.Hint:='Çok satirli ipucunu '+chr(13)+  
        'mutlaka denemelisiniz '+chr(13)+  
        'çok güzel';  
end;
```

---

Bir klasörü ve onun altındaki tüm dosyalari ve klasörleri silme

Ancak salt okunur (read only) özelliği olan ve kullanımda olan dosyalar silinmez.

```
procedure TForm1.Button1Click(Sender: TObject);
var
  DirInfo: TSearchRec;
  r : Integer;
begin
  r := FindFirst('C:\Download\Test\*.*', FaAnyfile, DirInfo);
  while r = 0 do begin
    if ((DirInfo.Attr and FaDirectory <> FaDirectory) and
        (DirInfo.Attr and FaVolumeId <> FaVolumeID)) then
      if DeleteFile(pChar('C:\Download\test\' + DirInfo.Name))
        = false then
        ShowMessage('C:\Download\test\' + DirInfo.Name + ' silinmiyor!!!');
    r := FindNext(DirInfo);
  end;
  SysUtils.FindClose(DirInfo);
  if RemoveDirectory('C:\Download\Test') = false then
    ShowMessage('C:\Download\test klasörü silinmiyor!!!');
end;
```

---

Baslat butonunu gizlemek veya kullanılmaz hale getirmek

```
procedure TForm1.Button1Click(Sender: TObject);
var
  Rgn : hRgn;
begin
  // Baslat butonunu gizle
  Rgn := CreateRectRgn(0, 0, 0, 0);
  SetWindowRgn(FindWindowEx(FindWindow('Shell_TrayWnd', nil),
                                0,
                                'Button',
                                nil),
                Rgn,
                true);
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  //Gizlenen Baslat butonunu eski haline döndürmek için
  SetWindowRgn(FindWindowEx(FindWindow('Shell_TrayWnd', nil),
                                0,
                                'Button',
                                nil),
                0,
                true);
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  //Baslat butonunu kullanılmaz yap
  EnableWindow(FindWindowEx(FindWindow('Shell_TrayWnd', nil),
                                0,
                                'Button',
                                nil),
                false);
end;

procedure TForm1.Button4Click(Sender: TObject);
```

```
begin
  //Kullanılmaz yapılan Baslat butonunu eski haline getirmek için
  EnableWindow(FindWindowEx(FindWindow('Shell_TrayWnd', nil),
                                0,
                                'Button',
                                nil),
                                true);
end;
```

---

Windows Gezgini istediniz bir klasörle açma

uses kısmına Shellapi unitini ekleyin.

```
ShellExecute(0,
  'explore',
  'C:\WINDOWS', //açmak istediginiz dizin
  nil,
  nil,
  SW_SHOWNORMAL);
```

---

Duvar kagidini degistirmek

```
var
  s: string;
begin
  s := 'c:\windows\athena.bmp';
  SystemParametersInfo(SPI_SETDESKWALLPAPER, 0, PChar(s), 0);
```

---

Form'un arka kısmına bir resmi dösemek

```
Bitmap: TBitmap;

procedure TForm1.FormCreate(Sender: TObject);
begin
  Bitmap := TBitmap.Create;
  Bitmap.LoadFromFile('C:\WINDOWS\cars.BMP');
end;

procedure TForm1.FormPaint(Sender: TObject);
var
  X, Y, W, H: LongInt;
begin
  with Bitmap do begin
    W := Width;
    H := Height;
  end;
  Y := 0;
  while Y < Height do begin
    X := 0;
    while X < Width do begin
      Canvas.Draw(X, Y, Bitmap);
      Inc(X, W);
    end;
    Inc(Y, H);
  end;
```



```
end;  
end;
```

---

## Bir Denetim Masasi uygulamasini calistirmak

Control Panel uygulamalari Windows\System klasörü altında bulunur. \*.CPL uzantili dosyalardir. Bu uygulamalari Control.Exe programi ile calistirabilirsiniz. Bazi Control Panel uygulamalari Windows\System klasöründe bulunmaz. Bunlarin ismini vererek calistirabilirsiniz.

```
WinExec('C:\WINDOWS\CONTROL.EXE TIMEDATE.CPL', sw_ShowNormal);  
WinExec('C:\WINDOWS\CONTROL.EXE MOUSE', sw_ShowNormal);  
WinExec('C:\WINDOWS\CONTROL.EXE PRINTERS', sw_ShowNormal);
```

---

## Sayilari virgüllerle yazmak

Sayi windows'unuz ayarina göre 12.345.678 veya 12,345,678 seklinde gösterilir.

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
    i : integer;  
begin  
    i := 12345678;  
    Mem1.Lines.Add(FormatFloat('#, ', i));
```

---

## Sistem Tarihini ve Saatini Degistirmek

Sistemin tarihini ve saatini degistirmek için SetLocalTime fonksiyonunu kullanabilirsiniz.

```
var  
    t:TSystemTime;  
begin  
    t.wYear:=1998;  
    t.wMonth:=5;  
    t.wDay:=23;  
    t.wHour:=12;  
    t.wMinute:=34;  
    SetLocalTime(t);  
end;
```

---

## Fareyi mesgul sekilde göstermek

```
try  
    Screen.Cursor := crHourGlass;  
    {buraya kodunuzu yazin...}  
finally  
    Screen.Cursor := crDefault;  
end;  
Application.ProcessMessages;
```

---

```
Dos programi kullanmak için;  
WinExec('command.com /c progdos.exe',sw_ShowNormal);
```

---

#### Çevirme

```
Edit1.OnChange := NIL;  
if Edit1.Text = 'Emre ÖZ' then  
  Edit1.Text := 'Emre';  
Edit1.OnChange := Edit1Change;
```

---

ilk harf büyük diğerleri küçük

Edit1'e yazılan yazının ilk harfini büyük diğer harflerini küçük yapıyor.

```
var  
OldStart : Integer;  
begin  
  With Edit1 do  
    if Text <> '' then begin  
      OnChange := NIL;  
      OldStart := SelStart;  
      Text := UpperCase(Copy(Text,1,1))+LowerCase(Copy(Text,2,Length(Text)));  
      SelStart := OldStart;  
      OnChange := Edit1Change;  
    end;  
  end;
```

---

#### Taskbar gizleme

```
procedure hideTaskbar;  
var  
  wndHandle : THandle;  
  wndClass : array[0..50] of Char;  
begin  
  StrPCopy(@wndClass[0], 'Shell_TrayWnd');  
  wndHandle := FindWindow(@wndClass[0], nil);  
  ShowWindow(wndHandle, SW_HIDE);  
end;
```

#### Taskbar gösterme

```
procedure showTaskbar;  
var  
  wndHandle : THandle;  
  wndClass : array[0..50] of Char;  
begin  
  StrPCopy(@wndClass[0], 'Shell_TrayWnd');  
  wndHandle := FindWindow(@wndClass[0], nil);
```

```
ShowWindow(wndHandle, SW_RESTORE);
end;
```

---

## Ses kaydetme

Bu program mikrofondan sesi alip bir wav dosyasi içine kayit ediyor.

```
uses
MMSystem,
WinCrt;
function RecordSound(MMsecs: LongInt): LongInt;
var
DeviceID: Word;
Return: LongInt;
MciOpen: TMCI_Open_Parms;
MciRecord: TMCI_Record_Parms;
MciPlay: TMCI_Play_Parms;
MciSave: TMCI_SaveParms;
Result2: LongInt;
Flags: Word;

begin
MciOpen.lpstrDeviceType := 'waveaudio';
MciOpen.lpstrElementName := '';
Flags := Mci_Open_Element or Mci_Open_Type;
Result2 := MciSendCommand(0, MCI_OPEN, Flags, LongInt(@MciOpen));
DeviceID := MciOpen.wDeviceId;

WriteLn('Kayit');
MciRecord.dwTo := MMsecs;
Flags := Mci_To or Mci_Wait;
Result2 := MciSendCommand(DeviceID, Mci_Record, Flags, LongInt(@MciRecord));

WriteLn('Durdur');
mciPlay.dwFrom := 0;
Flags := Mci_From or Mci_Wait;
MciSendCommand(DeviceId, Mci_Play, Flags, LongInt(@MciPlay));

mciSave.lpfileName := 'Ses.Wav';
Flags := MCI_Save_File or Mci_Wait;
Result := MciSendCommand(DeviceID, MCI_Save, Flags, LongInt(@MciSave));
MciSendCommand(DeviceID, Mci_Close, 0, LongInt(nil));

end;

begin
WriteLn('Basla');
```

```
RecordSound(10000);  
WriteLn('Bitir');
```

---

## DELPHİ NASIL EXE YAPIYOR ?

Delphi , delphi ile birlikte gelen dcc.exe ile projeleri derliyor. (Delphi 3'de dcc32.exe)

Örnek olarak

```
dcc /cw /b uls.dpr
```

komutlar :

/cw : Windows için,

/cd : Dos için,

/cp: Korumali mod için.

/b: Hepsini tekrar olustur,

/m: Hepsinin güncel oldugundan emin ol.

---

## BRC Nedir ?

Brc Res dosyalarini derliyor.

Bir res dosyasi açin ,(Notepad ile)

içine Icon ICON "emre.ico" yazin. (emre.ico dosyasi ayni klasörde bulunmeli. )

BRC -r EMRE.RES olarak derleyin.

Delphideki projenizin basinada {\$ Emre.rc} yazin. Böylece ikon dosyalarini tasimadan kullana bilirsiniz.

Sadece icon degil , ayrica BITMAP,TEXT de kullana bilirsiniz.

---