

## 팁모음집

금주가 몇번째 주인지 어떻게 구합니까

```
function kcIsLeapYear( nYear: Integer ): Boolean; // 윤년을 계산하는 함수
begin
    Result := (nYear mod 4 = 0) and ((nYear mod 100 <> 0) or (nYear mod 400 = 0));
end;
```

function kcMonthDays( nMonth, nYear: Integer ): Integer; // 한달에 몇일이 있는지를 계산하는 함수

```
const
    DaysPerMonth: array[1..12] of Integer = (31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
begin
    Result := DaysPerMonth[nMonth];
    if (nMonth = 2) and kcIsLeapYear(nYear) then Inc(Result);
end;
```

function kcWeekOfYear( dDate: TDateTime ): Integer; // 위의 두 함수를 써서 몇번째 주인지 계산하는 함수

```
var
    X, nDayCount: Integer;
    nMonth, nDay, nYear: Word;
begin
    nDayCount := 0;
    deCodeDate( dDate, nYear, nMonth, nDay );
    For X := 1 to ( nMonth - 1 ) do
        nDayCount := nDayCount + kcMonthDays( X, nYear );
    nDayCount := nDayCount + nDay;
    Result := ( ( nDayCount div 7 ) + 1 );
end;
```

긴 파일명 사용하기

```
function fileLongName(const aFile: String): String;
var
    aInfo: TSHFileInfo;
```

```

begin
  if SHGetFileInfo(PChar(aFile),0,aInfo,Sizeof(aInfo),SHGFI_DISPLAYNAME)<>0 then
    Result:=StrPas(aInfo.szDisplayName)
  else
    Result:=aFile;
end;

```

## 네트워크 검색

```

connections or persistent (won't normally get here);}
  r:=WNetOpenEnum(ListType,ResourceType,RESOURCEUSAGE_CONTAINER, nil,hEnum);
  { Couldn't enumerate through this container; just make a note of it and continue on: }
  if r<>NO_ERROR then
    begin
      AddShareString(TopContainerIndex,"");
      WNetCloseEnum(hEnum);
      Exit;
    end;

  { We got a valid enumeration handle; walk the resources: }
  while (1=1) do
    begin
      EntryCount:=1;
      NetResLen:=SizeOf(NetRes);
      r:=WNetEnumResource(hEnum,EntryCount,@NetRes,NetResLen);
      case r of
        0: begin
          { Yet another container to enumerate; call this function recursively to handle it: }
          if (NetRes[0].dwUsage=RESOURCEUSAGE_CONTAINER) or (NetRes[0].dwUsage=10)
then
            DoEnumerationContainer(NetRes[0])
          else
            case NetRes[0].dwDisplayType of
              { Top level type: }
              RESOURCEDISPLAYTYPE_GENERIC,
              RESOURCEDISPLAYTYPE_DOMAIN,

```

```

        RESOURCEDISPLAYTYPE_SERVER: AddContainer(NetRes[0]);
    { Share: }
        RESOURCEDISPLAYTYPE_SHARE: AddShare(TopContainerIndex,NetRes[0]);
    end;
end;
ERROR_NO_MORE_ITEMS: Break;
else begin
    MessageDlg('Error #'+IntToStr(r)+' Walking Resources.',mtError,[mbOK],0);
    Break;
end;
end;
end;

{ Close enumeration handle: }
WNetCloseEnum(hEnum);
end;

procedure TfrmMain.FormShow(Sender: TObject);
begin
    DoEnumeration;
end;

// Add item to tree view; indicate that it is a container:
procedure TfrmMain.AddContainer(NetRes: TNetResource);
var
    ItemName: String;
begin
    ItemName:=Trim(String(NetRes.lpRemoteName));
    if Trim(String(NetRes.lpComment))<>" then
        begin
            if ItemName<>" then ItemName:=ItemName+' ';
            ItemName:=ItemName+'('+String(NetRes.lpComment)+)';
        end;
    tvResources.Items.Add(tvResources.Selected,ItemName);
end;

```

```

// Add child item to container denoted as current top:
procedure TfrmMain.AddShare(TopContainerIndex: Integer; NetRes:TNetResource);
var
  ItemName: String;
begin
  ItemName:=Trim(String(NetRes.lpRemoteName));
  if Trim(String(NetRes.lpComment))<>" then
    begin
      if ItemName<>" then ItemName:=ItemName+' ';
      ItemName:=ItemName+'('+String(NetRes.lpComment)+)';
    end;

    tvResources.Items.AddChild(tvResources.Items[TopContainerIndex],ItemName);
end;

```

{ Add child item to container denoted as current top;  
this just adds a string for purposes such as being unable to enumerate a container. That is, the container's shares are not accessible to us. }

```

procedure TfrmMain.AddShareString(TopContainerIndex: Integer;ItemName: String);
begin
  tvResources.Items.AddChild(tvResources.Items[TopContainerIndex],ItemName);
end;

```

{ Add a connection to the tree view.

Mostly used for persistent and currently connected resources to be displayed. }

```

procedure TfrmMain.AddConnection(NetRes: TNetResource);
var
  ItemName: String;
begin
  ItemName:=Trim(String(NetRes.lpLocalName));
  if Trim(String(NetRes.lpRemoteName))<>" then
    begin
      if ItemName<>" then ItemName:=ItemName+' ';
      ItemName:=ItemName+'-> '+Trim(String(NetRes.lpRemoteName));
    end;

    tvResources.Items.Add(tvResources.Selected,ItemName);

```

```
end;

// Expand all containers in the tree view:
procedure TfrmMain.mniExpandAllClick(Sender: TObject);
begin
    tvResources.FullExpand;
end;

// Collapse all containers in the tree view:
procedure TfrmMain.mniCollapseAllClick(Sender: TObject);
begin
    tvResources.FullCollapse;
end;

// Allow saving of tree view to a file:
procedure TfrmMain.mniSaveToFileClick(Sender: TObject);
begin
    if dlgSave.Execute then
        tvResources.SaveToFile(dlgSave.FileName);
end;

// Allow loading of tree view from a file:
procedure TfrmMain.mniLoadFromFileClick(Sender: TObject);
begin
    if dlgOpen.Execute then
        tvResources.LoadFromFile(dlgOpen.FileName);
end;

// Rebrowse:
procedure TfrmMain.btnOKClick(Sender: TObject);
begin
    DoEnumeration;
end;

end.
```

네트워크 드라이브 등록하기

```
procedure TStartForm.NetBtnClick(Sender: TObject);
var
  OldDrives: TStringList;
  i: Integer;
begin
  OldDrives := TStringList.Create;
  OldDrives.Assign(Drivebox.Items);    // Remember old drive list
  // Show the connection dialog
  if WNetConnectionDialog(Handle, RESOURCETYPE_DISK) = NO_ERROR then
  begin
    DriveBox.TextCase := tcLowerCase;           // Refresh the drive list box
    for i := 0 to DriveBox.Items.Count - 1 do
    begin
      if Olddrives.IndexOf(Drivebox.Items[i]) = -1 then
      begin // Find new Drive letter
        DriveBox.ItemIndex := i;               // Updates the drive list box to new drive letter
        DriveBox.Drive := DriveBox.Text[1];    // Cascades the update to connected directory lists,
      end;
    end;
  end;
  DriveBox.SetFocus;
end;
```

다른 윈도우에서 선택된 문자열 복사하기

```
procedure TForm1.WMHotkey(Var msg: TWMHotkey);
var
  hOtherWin,
  hFocusWin: THandle;
  OtherThreadID, ProcessID: DWORD;
begin
  hOtherWin := GetForegroundWindow;
  if hOtherWin = 0 then
    Exit;
```

```

OtherThreadID := GetWindowThreadProcessID( hOtherWin, @ProcessID );
if AttachThreadInput( GetCurrentThreadID, OtherThreadID, True ) then
begin
    hFocusWin := GetFocus;
    if hFocusWin <> 0 then
    try
        SendMessage( hFocusWin, WM_COPY, 0, 0 );
    finally
        AttachThreadInput( GetCurrentThreadID, OtherThreadID, False );
    end;
end;

```

```

Memo1.Lines.Add( Clipboard.AsText );

```

```

if IsIconIC( Application.Handle ) then
    Application.Restore;
end;

```

### 다른 Application 에 Data 전달하기

WM\_COPYDATA-다른 Application 에 Data 전달

```

unit other_ap;
{ 다른 Application 을 찾아서 WM_COPYDATA 로 DATA 를 전달 }
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;

const WM_COPYDATA = $004A;

type
    TForm1 = class(TForm)
        Button1: TButton;
        Memo1: TMemo;
        procedure Button1Click(Sender: TObject);
    private
        { Private declarations }
    end;

```

```

    procedure WMCopyData(var m : TMessage); message WM_COPYDATA;
public
    { Public declarations }
end;

var
    form1: TForm1;

implementation
    {$R *.DFM}

type
    PCopyDataStruct = ^TCopyDataStruct;
    TCopyDataStruct = record
        dwData: LongInt;
        cbData: LongInt;
        lpData: Pointer;
    end;

type
    PRecToPass = ^TRecToPass;
    TRecToPass = packed record
        s : string[255];
        i : integer;
    end;

procedure TForm1.WMCopyData(var m : TMessage);
begin
    Memo1.Lines.Add(PRecToPass(PCopyDataStruct(m.LParam)^.lpData)^.s);
    Memo1.Lines.Add(IntToStr(PRecToPass(PCopyDataStruct(m.LParam)^.lpData)^.i));
end;

procedure TForm1.Button1Click(Sender: TObject);
var
    h : THandle;
    cd : TCopyDataStruct;
    rec : TRecToPass;

```



```

begin
  if Form1.Caption = 'My App' then
    begin
      h := FindWindow(nil, 'My Other App');
      rec.s := 'Hello World - From My App';
      rec.i := 1;
    end
  else
    begin
      h := FindWindow(nil, 'My App');
      rec.s := 'Hello World - From My Other App';
      rec.i := 2;
    end;
  cd.dwData := 0;
  cd.cbData := sizeof(rec);
  cd.lpData := @rec;
  if h <> 0 then
    SendMessage(h, WM_CopyData, Form1.Handle, LongInt(@cd));
  end;

end.

```

델파이 중복실행방지

```

unit PrevInst;

interface

uses
  WinTypes, WinProcs, SysUtils;

type
  PHWND = ^HWND;
  function EnumFunc(Wnd:HWND; TargetWindow:PHWND): bool; export;
  procedure GotoPreviousInstance;

```

implementation

```
function EnumFunc(Wnd:HWND; TargetWindow:HWND): bool;
var
  ClassName : array[0..30] of char;
begin
  Result := true;
  if GetWindowWord(Wnd,GWW_HINSTANCE) = hPrevInst then
  begin
    GetClassName(Wnd,ClassName,30);
    if StrIComp(ClassName,'TApplication') = 0 then
    begin
      TargetWindow^ := Wnd;
      Result := false;
    end;
  end;
end;

procedure GotoPreviousInstance;
var
  PrevInstWnd : HWND;
begin
  PrevInstWnd := 0;
  EnumWindows(@EnumFunc,longint(@PrevInstWnd));
  if PrevInstWnd <> 0 then
  begin
    if IsIconic(PrevInstWnd) then
      ShowWindow(PrevInstWnd, SW_RESTORE)
    else
      BringWindowToTop(PrevInstWnd);
  end;
end.
```

이러한 유닛을 프로젝트에 추가 하신후 DPR 소스의 BEGIN - END 를 다음과 같

이

수정해 주세요

```
begin
  if hPrevInst <> 0 then
    GotoPreviousInstance
  else
    begin
      Application.CreateForm(MyForm, MyForm);
      Application.Run;
    end;
end.
```

### 델파이에서 한글 토글하기

델파이 2.0 이하에서는  
ims.pas 를 이용하여 한영토글을 구현했는데,  
3.0 이상 에서는 한영토글에 대한 간단한 답에 있더군요.  
TEdit 에 ImsMode 프라퍼티를 이용합니다.

```
edit1.ImeMode:=imHangul; //한글모드
edit2.ImeMode:=imAlpha; //영문모드
```

입력이 한글이 많을 경우,  
입력 초기모드를 한글모드로 바꿔준다면,  
사용자의 한/영키를 누르는 것을 없애줄 수 있겠지요.

### 델파이에서 자동으로 한글입력모드로 변경시키는 소스

uses 절에 Imm 을 추가하세요

그런다음 아래 프로시저를 작성하여 OnEnter 이벤트에서  
한글을 on 하시구요 OnExit 이벤트에서 off 하세요

```
procedure TForm1.SetHangeulMode(SetHangeul: Boolean);
var
  tMode : HIMC;
begin
```

```

tMode := ImmGetContext(handle);
if SetHangeul then // 한글모드로
                    ImmSetConversionStatus(tMode, IME_CMODE_HANGEUL,
IME_CMODE_HANGEUL)

                    else // 영문모드로
                        ImmSetConversionStatus(tMode, IME_CMODE_ALPHANUMERIC,
IME_CMODE_ALPHANUMERIC);
end;

```

텔파이어에서 폼을 사정없이 뜯어내는 방법의 소스

```

var
    WindowRgn,HoleRgn : HRgn;
begin
    WindowRgn := 0;
    GetWindowRgn(handle, WindowRgn);
    DeleteObject(WindowRgn);
    WindowRgn := CreateRectRgn(0,0,Width,Height);
    HoleRgn := CreateRectRgn(16,25,126,236);
    CombineRgn(WindowRgn, WindowRgn, HoleRgn, RGN_DIFF);
    SetWindowRgn(handle, WindowRgn, TRUE);
    DeleteObject(HoleRgn);
end;

```

텔파이어에서의 키값

아래에 가상키 값 리스트입니다....

```

vk_LButton   = $01;
vk_RButton   = $02;
vk_Cancel    = $03;
vk_MButton   = $04; { NOT contiguous with L & RBUTTON }
vk_Back      = $08;
vk_Tab       = $09;
vk_Clear     = $0C;

```

```
vk_Return    = $0D;
vk_Shift     = $10;
vk_Control   = $11;
vk_Menu      = $12;
vk_Pause     = $13;
vk_Capital   = $14;
vk_Escape    = $1B;
vk_Space     = $20;
vk_Prior     = $21;
vk_Next      = $22;

vk_End       = $23;
vk_Home      = $24;
vk_Left      = $25;
vk_Up        = $26;
vk_Right     = $27;
vk_Down      = $28;
vk_Select    = $29;
vk_Print     = $2A;
vk_Execute   = $2B;
vk_SnapShot  = $2C;
{ vk_Copy    = $2C not used by keyboards }
vk_Insert    = $2D;
vk_Delete    = $2E;
vk_Help      = $2F;
{ vk_A thru vk_Z are the same as their ASCII equivalents: 'A' thru 'Z' }
{ vk_0 thru vk_9 are the same as their ASCII equivalents: '0' thru '9' }

vk_NumPad0   = $60;
vk_NumPad1   = $61;
vk_NumPad2   = $62;
vk_NumPad3   = $63;
vk_NumPad4   = $64;
vk_NumPad5   = $65;
vk_NumPad6   = $66;
vk_NumPad7   = $67;
```

vk\_NumPad8 = \$68;  
vk\_NumPad9 = \$69;  
vk\_Multiply = \$6A;  
vk\_Add = \$6B;  
vk\_Separator = \$6C;  
vk\_Subtract = \$6D;  
vk\_Decimal = \$6E;  
vk\_Divide = \$6F;  
vk\_F1 = \$70;  
vk\_F2 = \$71;  
vk\_F3 = \$72;  
vk\_F4 = \$73;  
vk\_F5 = \$74;  
  
vk\_F6 = \$75;  
vk\_F7 = \$76;  
vk\_F8 = \$77;  
vk\_F9 = \$78;  
vk\_F10 = \$79;  
vk\_F11 = \$7A;  
vk\_F12 = \$7B;  
vk\_F13 = \$7C;  
vk\_F14 = \$7D;  
vk\_F15 = \$7E;  
vk\_F16 = \$7F;  
vk\_F17 = \$80;  
vk\_F18 = \$81;  
vk\_F19 = \$82;  
vk\_F20 = \$83;  
vk\_F21 = \$84;  
vk\_F22 = \$85;  
vk\_F23 = \$86;  
vk\_F24 = \$87;  
vk\_NumLock = \$90;  
vk\_Scroll = \$91;

## 디렉토리에 관련된 함수

```
function GetCurrentDir: string; // 현재의 Directory
function ExtractFileDir(const FileName: string): string;
// Directory 만 Return .Filename 빼고
function ExtractFileName(const FileName: string): string;
// 파일 이름만 Return
```

## 동작중인 프로그램 죽이기

```
unit Unit1;

interface

uses

    Windows, Messages, SysUtils, Classes, Graphics, Controls,
    Forms, Dialogs, StdCtrls, TlHelp32;

type
    TForm1 = class(TForm)
        ListBox1: TListBox;
        B_Search: TButton;
        B_Terminate: TButton;
        procedure B_SearchClick(Sender: TObject);
        procedure B_TerminateClick(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form1: TForm1;

implementation
{$R *.DFM}
```

```

// kernel32.dll 을 사용하여 현재 떠있는 process 를 읽어온다
procedure Process32List(Slist: TStrings);
var
    Process32: TProcessEntry32;
    SHandle:   THandle; // the handle of the Windows object
    Next:      BOOL;
begin
    Process32.dwSize := SizeOf(TProcessEntry32);
    SHandle          := CreateToolHelp32Snapshot(TH32CS_SNAPPROCESS, 0);

    if Process32First(SHandle, Process32) then
    begin
        // 실행파일명과 process object 저장
        Slist.AddObject(Process32.szExeFile, TObject(Process32.th32ProcessID));
        repeat
            Next := Process32Next(SHandle, Process32);
            if Next then
                Slist.AddObject(Process32.szExeFile, TObject(Process32.th32ProcessID));
        until not Next;
    end;
    CloseHandle(SHandle); // closes an open object handle
end;

procedure TForm1.B_SearchClick(Sender: TObject);
begin
    // 현재 실행중인 process 를 검색
    ListBox1.Items.Clear;
    Process32List(ListBox1.Items);
end;

procedure TForm1.B_TerminateClick(Sender: TObject);
var
    hProcess: THandle;
    ProcId:   DWORD;
    TermSucc: BOOL;

```



```

begin
  // 현재 실행중인 process 를 kill
  if ListBox1.ItemIndex < 0 then System.Exit;
  ProcId := DWORD(ListBox1.Items.Objects[ListBox1.ItemIndex]);
  // 존재하는 process object 의 handle 을 return 한다
  hProcess := OpenProcess(PROCESS_ALL_ACCESS, TRUE, ProcId);
  if hProcess = NULL then
    ShowMessage('OpenProcess error !');
  // 명시한 process 를 강제 종료시킨다
  TermSucc := TerminateProcess(hProcess, 0);
  if TermSucc = FALSE then
    ShowMessage('TerminateProcess error !')
  else
    ShowMessage(Format('Process# %x terminated successfully !', [ProcId]));
end;

end.

```

### 레지스트리를 이용한 모뎀찾기

```

WRegistry := TRegistry.Create;
with Wregistry do
begin
  rootkey := HKEY_LOCAL_MACHINE;
  if OpenKey
    ('\System\CurrentControlSet\Services\Class\Modem\0000',False) then
    Showmessage ('모뎀이 있습니다. ');
  ...
  free..
end;

```

### 마우스의 Enter/Exit Event 사용하기

```

TForm1 = class(TForm)
  Image1 : TImage;
private

```

```

    m_orgProc    : TWndMethod;
    procedure    ImageProc ( var Msg : TMessage );
public
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
end;
:
:
procedure TForm1.FormCreate(Sender:TObject);
begin
    m_orgProc := Image1.WindowProc;
    Image1.WindowProc := ImageProc;
end;

procedure TForm1.FormDestroy(Sender:TObject);
begin
    Image1.WindowProc := m_orgProc;
end;

procedure TForm1.ImageProc( var Msg : TMessage );
begin
    case Msg.Msg of
        CM_MOUSELEAVE:
            begin
                // 여기서 컨트롤에 마우스가 들어왔을 때를 처리합니다.
            end;
        CM_MOUSEENTER:
            begin
                // 여기서 컨트롤로부터 마우스가 벗어날때 부분을 처리합니다.
            end;
    end;
    m_orgProc(Msg);
end;

end;

```

## 마우스의 범위 제한하기

다음 예제는 폼에 2 개의 버튼을 두고 첫번째 버튼을 누르면 마우스가 폼 밖으로 못나가게 하고, 두번째 버튼을 누르면 원래대로 바꿔주는 프로그램입니다...

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
    Rect : TRect;
```

```
begin
```

```
    Rect := BoundsRect;
```

```
    InflateRect(Rect, 0, 0);
```

```
    ClipCursor(@Rect);
```

```
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
```

```
begin
```

```
    ClipCursor(nil);
```

```
end;
```

## Message 박스에 두줄출력하기

```
MessageDlg('문자열' + chr(13) + '문자열', mtInformation, [mbOK], 0);
```

참고 : 윈도우에서는 3 줄까지 가능함. 3 줄 이상의 문자열은 자동으로 정렬하지 않으니 개발자가 주의해야 함.

## 바탕화면 바꾸기

```
GetMem( ThePChar , 255 );
```

```
    StrPCopy( ThePChar , 'wallpaper.bmp');
```

```
    SystemParametersInfo( SPI_SETDESKWALLPAPER , 0 ,
```

```
        ThePChar , SPIF_SENDWININICHANGE );
```

```
    Freemem( ThePChar , 255 );
```

## 브라우저 동작하기

UrlMon 유닛으로 선언되고 있다 HlinkNavigateString Win32 API 을(를) 씁니다.

호출 예 :

```
HlinkNavigateString(nil,'http://www.borland.co.jp/');
```

만약 액티브 폼의 중(안)에서 불러내고 싶은 경우에는 이하와 같이 지정합니다 :

```
HlinkNavigateString(ComObject,'http://www.borland.co.jp/');
```

ShellApi 유닛으로 선언되고 있다 ShellExecute 을(를) 쓰는 것도 가능합니다.

```
ShellExecute(0, 'open', 'http://www.borland.co.jp/', nil, nil, SW_SHOW)
```

#### 사용자가 조합키를 누른것처럼 처리하는 방법

다음 소스를 참고하기 바랍니다. 중요한 부분은 조합키중 키와 키, 키와 같이 홀드(hold) 상태인 키를 확인해서 키값을 포스팅해 주는 것입니다.

완전하다면 더할나위 없이 좋겠지만, 그냥 자신의 프로그램에 덧붙여 사용하거나 외부 참조로 사용해도 무방할 것입니다.

```
procedure PostKeyEx( hWindow: HWND; key: Word; Const shift: TShiftState; Specialkey: Boolean );
```

```
type
```

```
  TBuffers = Array [0..1] of TKeyboardState;
```

```
var
```

```
  pKeyBuffers : ^TBuffers;
```

```
  lparam: LongInt;
```

```
begin
```

```
  if IsWindow( hWindow ) then
```

```
    begin
```

```
      pKeyBuffers := nil;
```

```
      lparam := MakeLong( 0, MapVirtualKey( key, 0 ) );
```

```
      if Specialkey then
```

```

    lparam := lparam or $1000000;
New( pKeyBuffers );
try
    GetKeyboardState( pKeyBuffers^[1] );
    FillChar( pKeyBuffers^[0],Sizeof( TKeyboardState ), 0 );
    if ssShift In shift then
        pKeyBuffers^[0][VK_SHIFT] := $80;
    if ssAlt In shift then
        begin
            pKeyBuffers^[0][VK_MENU] := $80;
            lparam := lparam or $20000000;
        end;
    if ssCtrl in shift then
        pKeyBuffers^[0][VK_CONTROL] := $80;
    if ssLeft in shift then
        pKeyBuffers^[0][VK_LBUTTON] := $80;
    If ssRight in shift then
        pKeyBuffers^[0][VK_RBUTTON] := $80;
    if ssMiddle in shift then
        pKeyBuffers^[0][VK_MBUTTON] := $80;

    SetKeyboardState( pKeyBuffers^[0] );

    if ssAlt in shift then
        begin
            PostMessage( hWnd, WM_SYSKEYDOWN, key, lparam);
            PostMessage( hWnd, WM_SYSKEYUP, key, lparam or $C0000000);
        end
    else
        begin
            PostMessage( hWnd, WM_KEYDOWN, key, lparam);
            PostMessage( hWnd, WM_KEYUP, key, lparam or $C0000000);
        end;

Application.ProcessMessages;

```

```

        SetKeyboardState( pKeyBuffers^[1] );
    finally
        if pKeyBuffers <> nil then
            Dispose( pKeyBuffers );
        end;
    end;
end; { PostKeyEx }

```

```

procedure TForm1.SpeedButton2Click(Sender: TObject);

```

```

Var

```

```

    W: HWND;

```

```

begin

```

```

    W := Memo1.Handle;

```

```

    PostKeyEx( W, VK_END, [ssCtrl, ssShift], False );

```

```

    // 전체 선택

```

```

    PostKeyEx( W, Ord('C'), [ssCtrl], False );

```

```

    // 클립보드로 복사

```

```

    PostKeyEx( W, Ord('C'), [ssShift], False );

```

```

    // "C"로 치환

```

```

    PostKeyEx( W, VK_RETURN, [], False );

```

```

    // 엔터키(새라인)

```

```

    PostKeyEx( W, VK_END, [], False );

```

```

    // 라인의 끝으로

```

```

    PostKeyEx( W, Ord('V'), [ssCtrl], False );

```

```

    // 붙여넣기

```

```

end;

```

## 시스템 About 사용하기

```

ShellAbout(Self.Handle,

```

```

    PChar(Application.Title),

```

```

    'http://home.t-online.de/home/mirbir.st/#13#10'mailto:mirbir.st@t-online.de',

```

```

    Application.Icon.Handle);

```

Self.Handle 은 현재 동작중인 Application 의 실행영역을 리턴하는 것이고....

PChar( Application.Title )은 Title 의 Caption 을 전달하는 것..

'문서영역'은 이 곳에서 만들었다는 표시...

Application.Icon.Handle 은 About 에서 보일 Icon 의 값을 전달하는 방법

### 시스템 Image 를 사용하는 TListView

```
procedure TDirTreeView.FindAllSubDirectories(pNode: TCTreeNode; ItsTheFirstPass: Boolean);
```

```
var
```

```
    srch: TSearchRec;  
    DOSerr: integer;  
    NewText: String;  
    NewPath: string;  
    tNode: TCTreeNode;  
    cNode: TCTreeNode;  
    ImagesHandleNeeded : boolean;  
    cCursor: HCursor;  
    NewList: TStringList;  
    i: integer;  
    tpath: string;
```

```
function TheImage(FileID: string; Flags: DWord; IconNeeded: Boolean): Integer;
```

```
var
```

```
    SHFileInfo: TSHFileInfo;  
begin  
    Result := SHGetFileInfo(pchar(FileID), 0,  
                           SHFileInfo,    SizeOf(SHFileInfo),  
                           Flags);  
  
    if IconNeeded then  
        Result := SHFileInfo.iIcon;  
end;
```

```
function ItHasChildren(const fPath: string): Boolean;
```

```
var
```

```
    srch: TSearchrec;  
    found: boolean;  
    DOSerr: integer;
```

```

begin
  chdir(fPath);
  Found := false;
  DOSerr := FindFirst('*.*',faDirectory,srch);
  while (DOSerr=0) and not(Found) do
    begin
      found := ((srch.attr and faDirectory)=faDirectory)
                and ((srch.name<>'.')
                    and (srch.name<>'..'));

      if not(found) then
        DOSerr := FindNext(srch);
      end;
    sysutils.FindClose(srch);
    chdir('..');
    Result := Found;
  end;
end;

```

```

begin
  tNode := TopItem;
  cCursor := Screen.cursor;
  Screen.cursor := crHourGlass;
  Items.BeginUpdate;
  SortType := stNone;
  tpath := uppercase(fCurrentPath);
  NewList := TStringList.Create;
  getdir(0,NewPath);
  if (NewPath[length(NewPath)]<>'\'') then
    NewPath := NewPath + '\';
  ImagesHandleNeeded := ItsTheFirstPass;
  DOSerr := FindFirst('*.*',faDirectory,srch);
  while DOSerr=0 do
    begin
      if ((srch.attr and faDirectory)=faDirectory) and
          ((srch.name<>'.') and (srch.name<>'..')) then
        begin
          NewText := lowercase(srch.name);

```



```

        NewText[1] := Uppcase(NewText[1]);
        NewList.AddObject(NewText, pointer(NewStr(NewPath+NewText)));
    end;
    DOSerr := FindNext(srch);
end;
sysutils.FindClose(srch);

NewList.Sorted := true;
with NewList do
for i := 0 to Count-1 do
begin
    cNode := Items.AddChildObject(pNode,Strings[i], PString(Objects[i]));
    with cNode do
begin
        NewText := PString(Data)^;
        HasChildren := ItHasChildren(NewText);
        if ImagesHandleNeeded then
begin
            Images.Handle := TheImage(NewText, SHGFI_SYSICONINDEX or
SHGFI_SMALLICON, false);
            ImagesHandleNeeded := false;
end;
            ImageIndex := TheImage(NewText, SHGFI_SYSICONINDEX or SHGFI_SMALLICON,
true);
            SelectedIndex := TheImage(NewText, SHGFI_SYSICONINDEX or SHGFI_SMALLICON or
SHGFI_OPENICON, True);
            if AnsiCompareText(NewText,fCurrentPath)=0 then
begin
                Expanded := true;
                StateIndex := SelectedIndex;
                Self.Selected := cNode;
            end
        else
            if (pos(uppercase(NewText),tPath)=1) then
begin
                Expanded := true;

```

```

        tNode := cNode;
    end;
end;
end;
NewList.Free;
Items.EndUpdate;
if Assigned(tNode) then
    TopItem := tNode;
    Screen.cursor := cCursor;
end;

```

### 실행하기

```

function fileExec(const aCmdLine: String; aHide, aWait: Boolean): Boolean;
var
    StartupInfo : TStartupInfo;
    ProcessInfo : TProcessInformation;
begin
    { setup the startup information for the application }
    FillChar(StartupInfo, SizeOf(TStartupInfo), 0);
    with StartupInfo do
    begin
        cb:= SizeOf(TStartupInfo);
        dwFlags:= STARTF_USESHOWWINDOW or STARTF_FORCEONFEEDBACK;
        if aHide then wShowWindow:= SW_HIDE
            else wShowWindow:= SW_SHOWNORMAL;
    end;

    Result := CreateProcess(nil,PChar(aCmdLine), nil, nil, False,
        NORMAL_PRIORITY_CLASS, nil, nil, StartupInfo, ProcessInfo);
    if aWait then
        if Result then
            begin
                WaitForInputIdle(ProcessInfo.hProcess, INFINITE);
                WaitForSingleObject(ProcessInfo.hProcess, INFINITE);
            end;
        end;
    end;

```

end;

function fileRedirectExec(const aCmdLine: String; Strings: TStrings): Boolean;

var

StartupInfo : TStartupInfo;

ProcessInfo : TProcessInformation;

aOutput : Integer;

aFile : String;

begin

Strings.Clear;

{ Create temp. file for output }

aFile:=FileTemp('.tmp');

aOutput:=FileCreate(aFile);

try

{setup the startup information for the application }

FillChar(StartupInfo, SizeOf(TStartupInfo), 0);

with StartupInfo do

begin

cb:= SizeOf(TStartupInfo);

dwFlags:= STARTF\_USESHOWWINDOW or STARTF\_FORCEONFEEDBACK or  
STARTF\_USESTDHANDLES;

wShowWindow:= SW\_HIDE;

hStdInput:= INVALID\_HANDLE\_VALUE;

hStdOutput:= aOutput;

hStdError:= INVALID\_HANDLE\_VALUE;

end;

Result := CreateProcess(nil,PChar(aCmdLine), nil, nil, False,

NORMAL\_PRIORITY\_CLASS, nil, nil, StartupInfo, ProcessInfo);

if Result then

begin

WaitForInputIdle(ProcessInfo.hProcess, INFINITE);

WaitForSingleObject(ProcessInfo.hProcess, INFINITE);

end;

finally

```

    FileClose(aOutput);
    Strings.LoadFromFile(aFile);
    DeleteFile(aFile);
end;
end;

```

## 외부 Application 의 Window 크기 조절하기

SHOWWINDOW-외부 Application 의 Window 크기 조절

아래 소스는 현재 active 된 window 의 list 를 구한 후 그중 하나를 선택하여 Minimized, Maximized 하는 예제입니다.

```

procedure GetAllWindowsProc(WinHandle: HWND; Slist: TStrings);
var
    P: array[0..256] of Char; {title bar 를 저장 할 buffer}
begin
    P[0] := #0;
    GetWindowText(WinHandle, P, 255); {window's title bar 를 알아낸다}
    if (P[0] <> #0) then
        if IsWindowVisible(WinHandle) then {invisible 한 window 는 제외}
            Slist.AddObject(P, TObject(WinHandle)); {window 의 handle 저장}
        end;
end;

procedure GetAllWindows(Slist: TStrings);
var
    WinHandle: HWND;
Begin
    WinHandle := FindWindow(nil, nil);
    GetAllWindowsProc(WinHandle, Slist);
    while (WinHandle <> 0) do {Top level 의 window 부터 순차적으로 handle 을 구한다}
        begin
            WinHandle := GetWindow(WinHandle, GW_HWNDNEXT);
            GetAllWindowsProc(WinHandle, Slist);
        end;
    end;
end;

```

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

```
begin
```

```
    ListBox1.Items.Clear;
```

```
    GetAllWindows(ListBox1.Items);
```

```
end;
```

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

```
begin
```

```
    if ListBox1.ItemIndex < 0 then
```

```
        System.Exit;
```

```
        {선택한 window 를 maximize}
```

```
        ShowWindow(HWND(ListBox1.Items.Objects[ListBox1.ItemIndex]), SW_MAXIMIZE);
```

```
end;
```

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

```
begin
```

```
    if ListBox1.ItemIndex < 0 then
```

```
        System.Exit;
```

```
        {선택한 window 를 minimize}
```

```
        ShowWindow(HWND(ListBox1.Items.Objects[ListBox1.ItemIndex]), SW_MINIMIZE);
```

```
end;
```

워크그룹의 호스트네임 읽어내기

```
program ShowSelf;
```

```
{$apptype console}
```

```
uses Windows, Winsock, SysUtils;
```

```
function HostIPFromHostEnt( const HostEnt: PHostEnt ): String;
```

```
begin
```

```
    Assert( HostEnt <> nil );
```

```
    // first four bytes are the host address
```

```
    Result := Format( '%d.%d.%d.%d', [Byte(HostEnt^.h_addr^[0]), Byte(HostEnt^.h_addr^[1]),
```

```
        Byte(HostEnt^.h_addr^[2]), Byte(HostEnt^.h_addr^[3])] );
```

```

end;

var
  r: Integer;
  WSAData: TWSAData;
  HostName: array[0..255] of Char;
  HostEnt: PHostEnt;
begin
  // initialize winsock
  r := WSASStartup( MakeLong( 1, 1 ), WSAData );
  if r <> 0 then
    RaiseLastWin32Error;
  try
    Writeln( 'Initialized winsock successfully...' );

    // get the host name (this is the current machine)
    FillChar( HostName, sizeof(HostName), #0 );
    r := gethostname( HostName, sizeof(HostName) );
    if r <> 0 then
      RaiseLastWin32Error;
    Writeln( 'Host name is ', HostName );

    // get host entry (address is contained within)
    HostEnt := gethostbyname( HostName );
    if not Assigned(HostEnt) then
      RaiseLastWin32Error;
    Writeln( 'Got host info...' );

    // dump out the host ip address
    Writeln( 'Host address: ', HostIPFromHostEnt( HostEnt ) );
  finally
    WSACleanup;
  end;
end.

```

윈도우시작메뉴 히스트로에 문서 등록하기

윈도우즈 시작메뉴에 있는 문서 히스토리에 자기가 생성한 파일을 등록할 수 있는 함수가 있습니다.

먼저 다음과 같은 프로시저를 프로그램에 넣어 주세요.

```
use ShellAPI, ShlObj;
```

```
procedure AddToStartDocument(FilePath: string)
begin
  SHAddToRecentDocs(SHARD_PATH, PChar(FilePath));
end;
```

자 이제 이 함수를 사용해 봅시다. 우린 파라미터로 문서의 경로를 넘겨주면 됩니다.

예)

```
AddToStartDocument(C:\Test.txt);
=>책에 이렇게 나와 있는데, 미스 프린팅 같군요.
-> 이렇게 해 주세요. AddToStartDocument('C:\Test.txt');
```

## 윈도우 배경그림바꾸기

Window 배경그림 바꾸기

```
procedure ChangeIt;
var
  Reg: TRegIniFile;
begin
  Reg := TRegIniFile.Create('Control Panel');
  Reg.WriteString('desktop','Wallpaper','c:\windows\kim.bmp');
  Reg.WriteString('desktop','TileWallpaper','1');
  Reg.Free;
  SystemParametersInfo(SPI_SETDESKWALLPAPER,0,nil,SPIF_SENDWININICHANGE);
end;
```

## Status 에 색깔 넣기

Status bar 에 색깔 넣기

StatusBar Font 의 색을 바꾸는 방법은 직접 그려주는 수 밖에 없습니다. 익히 아시겠지만 StatusBar 의 Item 이라 할 수 있는 TStatusPanel 에는 Style 이란게 있습니다. 이 값은 psText 나 psOwnerDraw 란 값을 갖는데 psOwnerDraw 일때에는 해당 Panel 을 그릴 때마다 OnDrawPanel event 가 호출됩니다. 이때에 원하는 색으로 직접 그려주시면 됩니다. psOwnerDraw 일때는 그려주지 않게되면 Text 값을 갖고 있다 하더라도 전혀 나오질 않으므로, 반드시 위에 말한 event 에서 그려주셔야 합니다.

다음에 예제를 보여드립니다.

```
procedure TfmMain.m_statusBarDrawPanel(StatusBar:
  TStatusBar; Panel: TStatusPanel; const Rect: TRect);
begin
  with StatusBar.Canvas do begin
    case Panel.ID of
      0 : Font.Color := clBlue;
      2 : if Panel.Text = '한글' then Font.Color := clRed
          else Font.Color := clBlue;
    end;
    FillRect(Rect);
    TextOut(Rect.Left+2,Rect.Top+2,Panel.Text);
  end;
end;
```

위에 ID 란 property 를 사용했는데요, 이것은 index 와는 약간 차이가 있습니다. index property 와 같이 부여되긴 하지만, item 이 추가, 삭제, 삽입되더라도 ID 의 값은 변하지 않습니다.

다시말해 한번 부여된 ID 는 다시 사용되지 않습니다.

## TreeView 프린트하기

TreeView and Print

paintTo can be made to work, you just have to scale the printer.canvas in the ratio of screen to printer resolution.



```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Printer.BeginDoc;
  try
    printer.canvas.moveto(100,100);
    SetMapMode( printer.canvas.handle, MM_ANISOTROPIC );
    SetWindowExtEx(printer.canvas.handle,
                   GetDeviceCaps(canvas.handle, LOGPIXELSX),
                   GetDeviceCaps(canvas.handle, LOGPIXELSY),
                   Nil);
    SetViewportExtEx(printer.canvas.handle,
                    GetDeviceCaps(printer.canvas.handle, LOGPIXELSX),
                    GetDeviceCaps(printer.canvas.handle, LOGPIXELSY),
                    Nil);
    treeview1.PaintTo( printer.canvas.handle, 100, 100 );
  finally
    printer.enddoc;
  end;
end;
```