

```

/*****
/*
/*----- G E T I C O N S . C -----*/
/*
/* Task      : Copies the icons of selected files to the Clipboard
/*-----*/
/* Authors    : Michael Tischer and Bruno Jennrich
/* developed on : 09/25/1995
/* last update  : 10/03/1995
/*****
#include <windows.h>
#include <shlobj.h>

/*****
/* CountFileNames - Counts the number of filenames returned by
/*
/*      GetOpenFileName() or GetSaveFileName().
/*-----*/
/* Parameter : pFileName - Address of string returned by
/*
/*      dialog box functions.
/* Return value : Number of strings in pFileName.
/*****
int CountFileNames( LPSTR pFileName )
{
    int iCnt;
    int iLen;

    iCnt = 0;
    iLen = strlen( pFileName );
    while( iLen )           // the last string has a length of 0
    {
        iCnt++;
        pFileName = &pFileName [iLen + 1 ];           // Get next filename
        iLen = strlen( pFileName );
    }
    return iCnt;
}

/*****
/* CopyIconsToClipboard - Places file icons in the Clipboard as
/*
/*      bitmaps.
/*-----*/
/* Parameters: pFileName - Address of filenames whose icons
/*
/*      are to be obtained.
/* Return value : none
/*****
void CopyIconsToClipboard( LPSTR pFileName )
{
    SHFILEINFO sfi;
    HDC         hMemDC;           // Memory DC
    HBITMAP     hBitmap, hbmOld;  // Bitmap handles
    HDC         hDCDesktop;      // DC of Desktop
    HBRUSH      hOldBrush;       // Brush handle
    int         iLen,            // Length of the current filename
                i,
                iNumFiles;       // Number of filenames

    iNumFiles = CountFileNames( pFileName );           // Number of files

    // Get DC of Desktop to create bitmap that is compatible with it
    hDCDesktop = GetDC( GetDesktopWindow() );

    // Create bitmap -----
    hBitmap = CreateCompatibleBitmap( hDCDesktop, 96, iNumFiles * 32 );

    // Release Desktop DC -----
    ReleaseDC( GetDesktopWindow(), hDCDesktop );

    hMemDC = CreateCompatibleDC( NULL );               // Create screen DC
    hbmOld = SelectObject( hMemDC, hBitmap );          // Select memory bitmap

    // Select brush for deleting the memory bitmap -----
    hOldBrush = SelectObject( hMemDC, GetStockObject( WHITE_BRUSH ) );
    // Delete memory bitmap -----
    Rectangle( hMemDC, 0, 0, 96, iNumFiles * 32 );
    SelectObject( hMemDC, hOldBrush );

    // With multiple selections the first filename is the directory from

```

```

// which the files that follow were selected. This directory is first
// made the current directory.
if( iNumFiles > 1 )
{
    SetCurrentDirectory( pFileName );
    pFileName = &pFileName[ strlen( pFileName ) + 1];
}

// Get icons of remaining files -----
i = 0;
iLen = strlen( pFileName );
while( iLen )
{
    SHGetFileInfo( pFileName,                                // Get small icon
        0,
        &sfi,
        sizeof( sfi ),
        SHGFI_ICON | SHGFI_SMALLICON );
    DrawIconEx( hMemDC, 0, i * 32,                            // Draw icon
        sfi.hIcon, 16, 16, 0,
        NULL, DI_NORMAL );
    DestroyIcon( sfi.hIcon );                                // Destroy icon

    SHGetFileInfo( pFileName,                                // Get large icon
        0,
        &sfi,
        sizeof( sfi ),
        SHGFI_ICON | SHGFI_LARGEICON ); // SHGFI_LARGEICON = 0
    DrawIconEx( hMemDC, 32, i*32,                            // Draw icon
        sfi.hIcon, 32, 32, 0,
        NULL, DI_NORMAL );
    DestroyIcon( sfi.hIcon );                                // Destroy icon

    SHGetFileInfo( pFileName,                                // Get large shortcut icon
        0,
        &sfi,
        sizeof( sfi ),
        SHGFI_ICON | SHGFI_LINKOVERLAY | SHGFI_LARGEICON );
    DrawIconEx( hMemDC, 64, i*32,                            // Draw icon
        sfi.hIcon, 32, 32, 0,
        NULL, DI_NORMAL );
    DestroyIcon( sfi.hIcon );                                // Destroy icon

    pFileName = &pFileName [iLen + 1 ];                    // Get next filename
    iLen = strlen( pFileName );
    i++;
}

OpenClipboard( NULL );                                     // Open Clipboard for current task
EmptyClipboard();                                         // Delete previous Clipboard contents

SetClipboardData( CF_BITMAP, hBitmap );                   // Bitmap to Clipboard
CloseClipboard();                                         // Release Clipboard

SelectObject( hMemDC, hbmOld );                           // Old bitmap back to memory DC
DeleteDC( hMemDC );                                       // Destroy memory DC

// The memory bitmap is destroyed by the Clipboard -----
}

/*****
*/ WinMain - Start function
*/-----*/
/* Parameters:      default parameters
/* Return value : default return value
*****/
int WINAPI WinMain( HINSTANCE hInst,
                    HINSTANCE hInstPrev,
                    LPSTR      lpCmdLine,
                    int         nCmdShow )
{
    char g_szFileName[ MAX_PATH * 20 ];
    OPENFILENAME ofn;                                     // Structure for opening file

    ofn.lStructSize = sizeof( ofn );

```

```

ofn.hwndOwner = NULL;
ofn.hInstance = hInst;
ofn.lpstrFilter = "All files (*.*)\0*.*";
ofn.lpstrCustomFilter = NULL;
ofn.nMaxCustFilter = 0;
ofn.nFilterIndex = 0;
ofn.lpstrFile = g_szFileName;           // Buffer for filenames
ofn.nMaxFile = sizeof( g_szFileName );
ofn.lpstrFileTitle = 0;
ofn.nMaxFileTitle = 0;
ofn.lpstrInitialDir = NULL;             // Start directory
ofn.lpstrTitle = "Get icon - (c) MITI & BHJ"; // Dialog box title
ofn.Flags = OFN_EXPLORER | OFN_ALLOWMULTISELECT; // Win95 style
ofn.nFileOffset = 0;
ofn.nFileExtension = 0;
ofn.lpstrDefExt = "";
ofn.lCustData = 0;
ofn.lpfnHook = NULL;
ofn.lpTemplateName = NULL;

if( GetOpenFileName( &ofn ) )          // Display Open File dialog box
{
    CopyIconsToClipboard( g_szFileName );
    MessageBox( NULL, "Icons copied to Clipboard", "Message", 0 );
}
return 0;
}

```