

```

/*****
/*
/*----- B R O W S E . C -----*/
/* Task      : Demonstration of SHBrowseForFolders() function */
/*-----*/
/* Authors    : Michael Tischer and Bruno Jennrich */
/* developed on : 09/28/1995 */
/* last update  : 09/28/1995 */
/*****
#include <windows.h>
#include <shlobj.h>

#include "resource.h"

#include "paintpic.h"

//----- Global Variables -----

// DisplayName of selected object -----
char g_szDisplayName[ MAX_PATH ];

// Icon index of selected object to system ImageList -----
int g_iIcon;

// Allocation table between checkboxes and SpecialFolder locations -----
int SpecialFolders[][2] = { { IDC_DESKTOP, CSIDL_DESKTOP },
                           { IDC_PROGRAMS, CSIDL_PROGRAMS },
                           { IDC_CONTROLS, CSIDL_CONTROLS },
                           { IDC_PERSONAL, CSIDL_PERSONAL },
                           { IDC_STARTUP, CSIDL_STARTUP },
                           { IDC_RECENT, CSIDL_RECENT },
                           { IDC_SENDTO, CSIDL_SENDTO },
                           { IDC_BITBUCKET, CSIDL_BITBUCKET },
                           { IDC_STARTMENU, CSIDL_STARTMENU },
                           { IDC_DESKTOPDIRECTORY,
                             CSIDL_DESKTOPDIRECTORY },
                           { IDC_DRIVES, CSIDL_DRIVES },
                           { IDC_NETWORK, CSIDL_NETWORK },
                           { IDC_NETHOOD, CSIDL_NETHOOD },
                           { IDC_FONTS, CSIDL_FONTS },
                           { IDC_TEMPLATE, CSIDL_TEMPLATES },
                           { IDC_PRINTERS, CSIDL_PRINTERS },
                           { -1, -1 } };

/*****
/* BrowseCallbackProc - Callback function for Browse dialog messages */
/*-----*/
/* Parameters:      hWnd    - Window handle of Browse dialog box */
/*                  uMsg     - Message */
/*                  lParam   - Message dependent data */
/*                  lpData   - Message dependent data */
/* Return value : always 0 */
/*****
int WINAPI BrowseCallbackProc( HWND    hWnd,
                              UINT     uMsg,
                              LPARAM   lParam,
                              LPARAM   lpData )
{
    switch( uMsg )
    {
        case BFFM_INITIALIZED:
            break;

        case BFFM_SELCHANGED:
        {
            // lpData contains address of a pidl
            char szPath[ MAX_PATH + 6 ];

            lstrcpy( szPath, "Path: " );
            SHGetPathFromIDList( (LPITEMIDLIST)lParam, &szPath[ 6 ] );
            SendMessage( hWnd,
                          BFFM_SETSTATUSTEXT,
                          0,
                          strlen( &szPath[6] ) ? (LPARAM)szPath :
                                                              (LPARAM)"Path: No path!" );
        }
    }
}

```

```

        break;
    }
    return 0; // Always returns 0
}

/*****
/* BrowseForFolder - Wrapper for SHBrowseForFolder
/*-----*/
/* Parameters:      hWnd      - Window handle of owner window, in case
/*                   Browse dialog application model is to
/*                   be displayed.
/*                   lpTitle   - Address of title string
/*                   uFlags    - Flags
/*                   pidl      - PIDL of starting point, where the
/*                   Browser is to begin displaying folders.
/*                   dwUser    - User information
/* Return value : Address of selected object's PIDL
/*               The caller has to release this address via
/*               CoTaskMemFree().
*****/
LPITEMIDLIST OpenBrowser( HWND      hWnd,
                          LPSTR     lpTitle,
                          UINT      uFlags,
                          LPITEMIDLIST pidl,
                          DWORD     dwUser )
{
    BROWSEINFO bi;
    LPITEMIDLIST pidlRetVal; // Memory for SHBrowse...() Return value

    bi.hwndOwner      = hWnd; // Owner window (for dialog modality)
    bi.pidlRoot       = pidl; // Starting point (PIDL)

    // Memory for DisplayName of selected object (draw MAX_PATH)
    bi.pszDisplayName = g_szDisplayName;

    bi.lpszTitle      = lpTitle; // Title to be displayed in dialog
    bi.ulFlags        = uFlags; // Dialog flag
    bi.lpfncb         = BrowseCallbackProc; // Callback function
    bi.lpParam        = dwUser; // User data (see BFFM_INITIALIZED)

    pidlRetVal = SHBrowseForFolder( &bi ); // Display dialog

    g_iIcon = bi.iImage; // Save icon index of selected object

    return pidlRetVal; // Return PIDL of selected object
}

/*****
/* BrowseDlgProc - Dialog function for SHBrowseForFolder() control
/*-----*/
/* Parameters:      default parameters
/* Return value : default return value
*****/
BOOL WINAPI BrowseDlgProc( HWND      hWnd,
                           UINT      uMsg,
                           WPARAM    wp,
                           LPARAM    lp )
{
    switch( uMsg )
    {
        case WM_INITDIALOG:
            // At start of dialog select "Null" as starting point
            SendMessage( GetDlgItem( hWnd, IDC_NULL ), BM_SETCHECK, TRUE, 0 );

            // Set default title -----
            SendMessage( GetDlgItem( hWnd, IDC_EDITTITLE ),
                          WM_SETTEXT,
                          0,
                          (LPARAM)"My wonderful Browser dialog" );

            break;
        case WM_COMMAND: // Which button was selected?
            switch( LOWORD( wp ) )
            {
                case IDCANCEL: // Close button of window
                    EndDialog( hWnd, 0 );
                    break;
            }
    }
}

```

```

case IDC_BROWSE:                                     // "Browse..."
{
    UINT          uFlags;                            // Browse for folder flags
    LPITEMIDLIST pidlStart,                          // Starting PIDL
                                           pidlReturn; // PIDL of selected object

    char szTitleText[ MAX_PATH ];                    // BFF title
    int i;

    // Put together flags from state of checkboxes -----
    uFlags = ( IsDlgButtonChecked( hWnd, IDC_BROWSEFORCOMPUTER ) ?
        BIF_BROWSEFORCOMPUTER : 0 ) |
        ( IsDlgButtonChecked( hWnd, IDC_BROWSEFORPRINTER ) ?
        BIF_BROWSEFORPRINTER : 0 ) |
        ( IsDlgButtonChecked( hWnd, IDC_DONTGOBELOWDOMAIN ) ?
        BIF_DONTGOBELOWDOMAIN : 0 ) |
        ( IsDlgButtonChecked( hWnd, IDC_RETURNFSANCESTORS ) ?
        BIF_RETURNFSANCESTORS : 0 ) |
        ( IsDlgButtonChecked( hWnd, IDC_RETURNONLYFSDIRS ) ?
        BIF_RETURNONLYFSDIRS : 0 );

    uFlags |= BIF_STATUSTEXT;
    // Store title to be displayed in dialog box in buffer -----
    SendMessage( GetDlgItem( hWnd, IDC_EDITTITLE ),
        WM_GETTEXT,
        sizeof( szTitleText ),
        (LPARAM)szTitleText );

    // Get PIDL of starting point from where the BFF dialog
    // box is to begin displaying Shell objects
    pidlStart = NULL;

    // Get CSIDL_... constant from radio buttons and --
    // SpecialFolder array --
    i = 0;
    while( ( SpecialFolders[ i ][ 0 ] > 0 ) && ( i >= 0 ) )
    {
        if( IsDlgButtonChecked( hWnd, SpecialFolders[ i ][ 0 ] ) )
        {
            SHGetSpecialFolderLocation( NULL,
                SpecialFolders[ i ][ 1 ],
                &pidlStart );

            i = -1; // End loop
        }
        else i++;
    }

    // Call BFF dialog box. Since this dialog box is displayed as
    // a modular box, szStatusText and szTitleText are allowed to
    // be local variables of this function.
    pidlReturn = OpenBrowser( hWnd,
        szTitleText,
        uFlags,
        pidlStart,
        0 );

    if( pidlReturn ) // Was BFF dialog box exited with OK?
        // Release returned PIDL -----
        CoTaskMemFree( pidlReturn );
    // Release starting PIDL -----
    if( pidlStart ) CoTaskMemFree( (LPVOID)pidlStart );
}
break;
}
break;
case WM_PAINT:
    // Draw PCI bitmap
    PaintPicture( (HINSTANCE)GetWindowLong( hWnd, GWL_HINSTANCE ),
        hWnd,
        IDB_PCI,
        GetDlgItem( hWnd, IDC_PCIPICTURE ) );
    break;
}
return FALSE;
}

```

```

/*****
/* WinMain - Start function */
/*-----*/
/* Parameters:    default parameters */
/* Return value : default return value */
/*****
int WINAPI WinMain( HINSTANCE hInst,
                   HINSTANCE hInstPrev,
                   LPSTR      lpzCmdLine,
                   int         nCmdShow )
{
    CoInitialize( NULL );           // Initialize OLE (for CoTaskMemFree() )

    // Display control dialog box for BrowseForFolder -----
    DialogBox( hInst,
               MAKEINTRESOURCE( IDD_BROWSEDDIALOG ),
               NULL,
               BrowseDlgProc );
    CoUninitialize();               // Uninstall OLE
    return 0;
}

```