

```

/*****
/*
/*----- S H O R T C U T . C -----*/
/* Task      : Functions for editing shortcuts */
/*-----*/
/* Authors      : Michael Tischer and Bruno Jennrich */
/* developed on  : 09/15/1995 */
/* last update   : 10/03/1995 */
/*****
#include <windows.h>
#include <shlobj.h>

//----- Typedefs -----

typedef IShellLink *LPSHELLLINK;          // Reproduce interface pointer
typedef IPersistFile *LPPERSISTFILE;

/*****
/* CreateShortcut : Function for creating a shortcut */
/*-----*/
/* Parameters:    pszPath      - Path to which the shortcut refers */
/*                pszArguments - Program arguments */
/*                pszLocation  - Where and under what name shall we */
/*                             save the shortcut? */
/*                pszWorkingDir - Working directory */
/*                nCmdShow     - How shall we display the application? */
/*                             window after the start? */
/* Return value : TRUE  - Shortcut was created */
/*               FALSE - Error */
/*****
BOOL CreateShortcut( LPSTR pszPath,
                    LPSTR pszArguments,
                    LPSTR pszLocation,
                    LPSTR pszWorkingDir,
                    int  nCmdShow )
{
    LPSHELLLINK pShellLink;
    HRESULT hrCoInit;
    HRESULT hr;

    hrCoInit = CoInitialize( NULL );
    hr = CoCreateInstance( &CLSID_ShellLink,
                          NULL,
                          CLSCTX_INPROC_SERVER,
                          &IID_IShellLink,
                          &pShellLink );

    if( SUCCEEDED( hr ) )
    {
        LPPERSISTFILE pPersistFile;

        // Get PersistFile interface -----
        if(SUCCEEDED(pShellLink->lpVtbl->QueryInterface(pShellLink,
                                                    &IID_IPersistFile,
                                                    &pPersistFile)))
        {
            WORD wsz[ MAX_PATH ];

            // Set program path -----
            hr = pShellLink->lpVtbl->SetPath(pShellLink,pszPath);

            // Set arguments -----
            if( SUCCEEDED( hr ) )
                hr = pShellLink->lpVtbl->SetArguments(pShellLink,
                                                    pszArguments );

            // Set working directory -----
            if( SUCCEEDED( hr ) )
                hr = pShellLink->lpVtbl->SetWorkingDirectory(pShellLink,
                                                            pszWorkingDir );

            // Set ShowCommand -----
            if( SUCCEEDED( hr ) )
                hr = pShellLink->lpVtbl->SetShowCmd(pShellLink,
                                                    nCmdShow );

            // Set path where shortcut is to be saved -----

```

```

        if( SUCCEEDED( hr ) )
        {
            MultiByteToWideChar( CP_ACP, 0, pszLocation, -1, wsz, MAX_PATH);
            hr = pPersistFile->lpVtbl->Save( pPersistFile, wsz, TRUE );
            hr = pPersistFile->lpVtbl->SaveCompleted( pPersistFile, wsz );
        }
        hr = pPersistFile->lpVtbl->Release( pPersistFile);
    }
    pShellLink->lpVtbl->Release( pShellLink );
}
// If OLE was initialized here, then we also have to uninitialize
// it here.
if( SUCCEEDED( hrCoInit ) ) CoUninitialize();

return SUCCEEDED( hr );
}

/*****
/* SetShortcutIcon : Changes icon of a shortcut
/*-----*/
/* Parameters:      pszLocation    - Name or file of shortcut
/*                  pszIconDLL     - Name of DLL containing icons
/*                               as a resource
/*                               (e.g.; moricons.dll)
/*                  iIcon          - Index of icon to use
/* Return value : TRUE  - Icon was set
/*                  FALSE - Error
*****/
BOOL SetShortcutIcon( LPSTR pszLocation,
                     LPSTR pszIconDLL,
                     int  iIcon )
{
    LPSHELLLINK pShellLink;
    HRESULT hrCoInit;
    HRESULT hr;

    hrCoInit = CoInitialize( NULL );
    hr = CoCreateInstance( &CLSID_ShellLink,
                          NULL,
                          CLSCTX_INPROC_SERVER,
                          &IID_IShellLink,
                          &pShellLink );

    if( SUCCEEDED( hr ) )
    {
        LPPERSISTFILE pPersistFile;

        // Get PersistFile interface -----
        if(SUCCEEDED(pShellLink->lpVtbl->QueryInterface(pShellLink,
                                                    &IID_IPersistFile,
                                                    &pPersistFile)))
        {
            WORD wsz[ MAX_PATH ];
            char szPath[ MAX_PATH ];

            // Load shortcut file -----

            MultiByteToWideChar( CP_ACP, 0, pszLocation, -1, wsz, MAX_PATH );
            hr = pPersistFile->lpVtbl->Load( pPersistFile,wsz,STGM_READWRITE);

            // Set icon of shortcut -----
            if( SUCCEEDED( hr ) )
            {
                WIN32_FIND_DATA fd;

                pShellLink->lpVtbl->GetPath( pShellLink, szPath,MAX_PATH,&fd,0);

                hr = pShellLink->lpVtbl->SetIconLocation( pShellLink,
                                                         pszIconDLL,
                                                         iIcon );

                // Save shortcut again under same name -----
                pPersistFile->lpVtbl->Save( pPersistFile, wsz, TRUE );
                pPersistFile->lpVtbl->SaveCompleted( pPersistFile, wsz );
            }
            pPersistFile->lpVtbl->Release( pPersistFile );
        }
        pShellLink->lpVtbl->Release( pShellLink );
    }
}

```

```
}  
// If OLE was initialized here, then we also have to uninstall it  
// here.  
if( SUCCEEDED( hrCoInit ) ) CoUninitialize();  
  
return SUCCEEDED( hr );  
}
```