```c
/*
    libxbee - a C library to aid the use of Digi's Series 1 XBee modules
              running in API mode (AP=2).

    Copyright (C) 2009  Attie Grande (attie@attie.co.uk)

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.
*/

/* ################################################################# */
/* ### Win32 DLL Code ############################################## */
/* ################################################################# */

/*  this file contains code that is used by Win32 ONLY */
#ifndef _WIN32
#error "This file should only be used on a Win32 system"
#endif

int ver(HWND hwnd, HINSTANCE hinst, LPWSTR lpszCmdLine, int nCmdShow) {
  char t[256];
  sprintf(t,"libxbee.dll\n%s\n%s",xbee_svn_version(),xbee_build_info());
  MessageBox(NULL, t, "libxbee Win32 DLL", MB_OK);
  return 0;
}

void xbee_UNLOADALL(xbee_hnd xbee) {
  if (xbee->next) xbee_UNLOADALL(xbee->next);
  _xbee_end(xbee);
}

/* this gets called when the dll is loaded and unloaded... */
BOOL APIENTRY DllMain(HANDLE hModule, DWORD dwReason, LPVOID lpReserved) {
  xbee_hnd xbee, xbee2;
  if (dwReason == DLL_PROCESS_DETACH) {
    /* ensure that libxbee has been shut down nicely */
    xbee_UNLOADALL(default_xbee);
  } else if (dwReason == DLL_PROCESS_ATTACH || dwReason == DLL_THREAD_ATTACH) {
    if (!glob_hModule) {
      /* keep a handle on the module */
      glob_hModule = (HMODULE)hModule;
    }
    if (!callbackmutexInitialized) {
      xbee_mutex_init(callbackmutex);
      callbackmutexInitialized = 1;
    }
  }
  return TRUE;
}

HRESULT DllCanUnloadNow(void) {
  xbee_UNLOADALL(default_xbee);
  return 1;
}

/* ################################################################# */
/* ### Win32 DLL COM Code ########################################## */
/* ################################################################# */

/* this function is from this tutorial:
     http://www.codeguru.com/Cpp/COM-Tech/activex/tutorials/article.php/c5567 */
BOOL RegWriteKey(HKEY roothk, const char *lpSubKey, LPCTSTR val_name,
                 DWORD dwType, void *lpvData,  DWORD dwDataSize) {
  /*  roothk:     HKEY_CLASSES_ROOT, HKEY_LOCAL_MACHINE, etc
      lpSubKey:   the key relative to 'roothk'
      val_name:   the key value name where the data will be written
      dwType:     REG_SZ,REG_BINARY, etc.
      lpvData:    a pointer to the data buffer
      dwDataSize: the size of the data pointed to by lpvData */
  HKEY hk;
  if (ERROR_SUCCESS != RegCreateKey(roothk,lpSubKey,&hk) ) return FALSE;
  if (ERROR_SUCCESS != RegSetValueEx(hk,val_name,0,dwType,(CONST BYTE *)lpvData,dwDataSize)) return FALSE;
  if (ERROR_SUCCESS != RegCloseKey(hk))   return FALSE;
  return TRUE;
}
```

```c

/* this is used by the regsrv32 application */
STDAPI DllRegisterServer(void) {
  char key[MAX_PATH];
  char value[MAX_PATH];

  wsprintf(key,"CLSID\\%s",dllGUID);
  wsprintf(value,"%s",dlldesc);
  RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)value, lstrlen(value));

  wsprintf(key,"CLSID\\%s\\InprocServer32",dllGUID);
  GetModuleFileName(glob_hModule,value,MAX_PATH);
  RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)value, lstrlen(value));

  wsprintf(key,"CLSID\\%s\\ProgId",dllGUID);
  lstrcpy(value,dllid);
  RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)value, lstrlen(value));

  lstrcpy(key,dllid);
  lstrcpy(value,dlldesc);
  RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)value, lstrlen(value));

  wsprintf(key,"%s\\CLSID",dllid);
  RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)dllGUID, lstrlen(dllGUID));

  return S_OK;
}

/* this is used by the regsrv32 application */
STDAPI DllUnregisterServer(void) {
  char key[MAX_PATH];
  char value[MAX_PATH];

  wsprintf(key,"%s\\CLSID",dllid);
  RegDeleteKey(HKEY_CLASSES_ROOT,key);

  wsprintf(key,"%s",dllid);
  RegDeleteKey(HKEY_CLASSES_ROOT,key);

  wsprintf(key,"CLSID\\%s\\InprocServer32",dllGUID);
  RegDeleteKey(HKEY_CLASSES_ROOT,key);

  wsprintf(key,"CLSID\\%s\\ProgId",dllGUID);
  RegDeleteKey(HKEY_CLASSES_ROOT,key);

  wsprintf(key,"CLSID\\%s",dllGUID);
  RegDeleteKey(HKEY_CLASSES_ROOT,key);

  return S_OK;
}
```