```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(void) {
  while (default_xbee) {
    _xbee_end(default_xbee);
  }
}

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

HRESULT DllCanUnloadNow(void) {
  if (default_xbee) return 0;
  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;
}

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

```
  86:   char value[MAX_PATH];
  87:
  88:   wsprintf(key,"CLSID\\%s",dllGUID);
  89:   wsprintf(value,"%s",dlldesc);
  90:   RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)value, lstrlen(value));
  91:
  92:   wsprintf(key,"CLSID\\%s\\InprocServer32",dllGUID);
  93:   GetModuleFileName(glob_hModule,value,MAX_PATH);
  94:   RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)value, lstrlen(value));
  95:
  96:   wsprintf(key,"CLSID\\%s\\ProgId",dllGUID);
  97:   lstrcpy(value,dllid);
  98:   RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)value, lstrlen(value));
  99:
 100:   lstrcpy(key,dllid);
 101:   lstrcpy(value,dlldesc);
 102:   RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)value, lstrlen(value));
 103:
 104:   wsprintf(key,"%s\\CLSID",dllid);
 105:   RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)dllGUID, lstrlen(dllGUID));
 106:
 107:   return S_OK;
 108: }
 109:
 110: /* this is used by the regsrv32 application */
 111: STDAPI DllUnregisterServer(void) {
 112:   char key[MAX_PATH];
 113:   char value[MAX_PATH];
 114:
 115:   wsprintf(key,"%s\\CLSID",dllid);
 116:   RegDeleteKey(HKEY_CLASSES_ROOT,key);
 117:
 118:   wsprintf(key,"%s",dllid);
 119:   RegDeleteKey(HKEY_CLASSES_ROOT,key);
 120:
 121:   wsprintf(key,"CLSID\\%s\\InprocServer32",dllGUID);
 122:   RegDeleteKey(HKEY_CLASSES_ROOT,key);
 123:
 124:   wsprintf(key,"CLSID\\%s\\ProgId",dllGUID);
 125:   RegDeleteKey(HKEY_CLASSES_ROOT,key);
 126:
 127:   wsprintf(key,"CLSID\\%s",dllGUID);
 128:   RegDeleteKey(HKEY_CLASSES_ROOT,key);
 129:
 130:   return S_OK;
 131: }
```