RFC: NeXus API porting for Microsoft Visual Studio

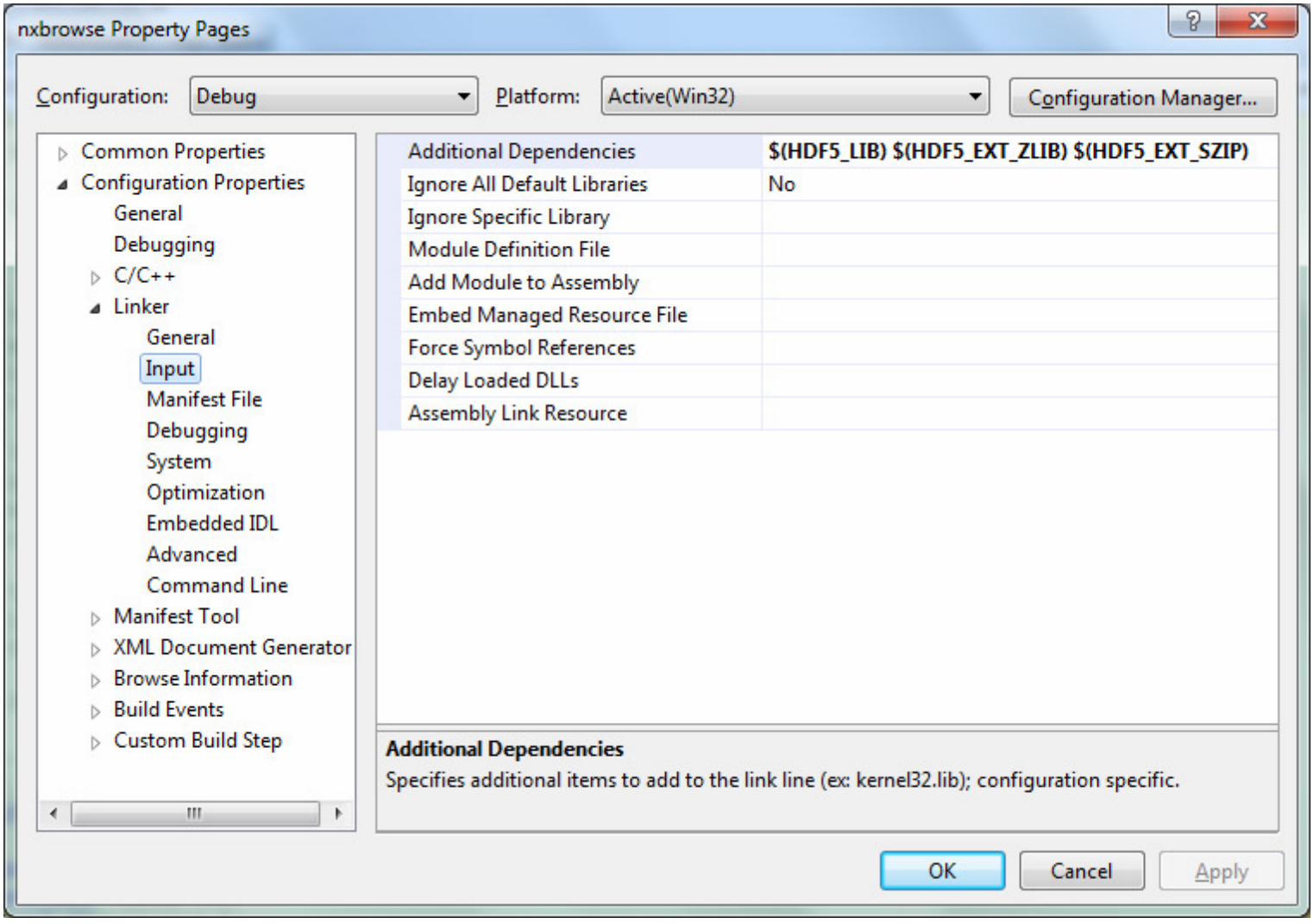
Pedro Vicente (pedro.vicente@space-research.org)

This is Part 2 (Applications project settings) of a Request for Comments (RFC) series of documents regarding a porting of the NeXus API for the Microsoft Visual Studio 2008 IDE.

1 Applications project settings

This section shows how to modify the Visual Studio project settings to build the NeXus API and applications with the underlying libraries HDF5, HDF4 or NXXML.

Figure 4.1: Linker Input Property Page and additional libraries in Additional Dependencies

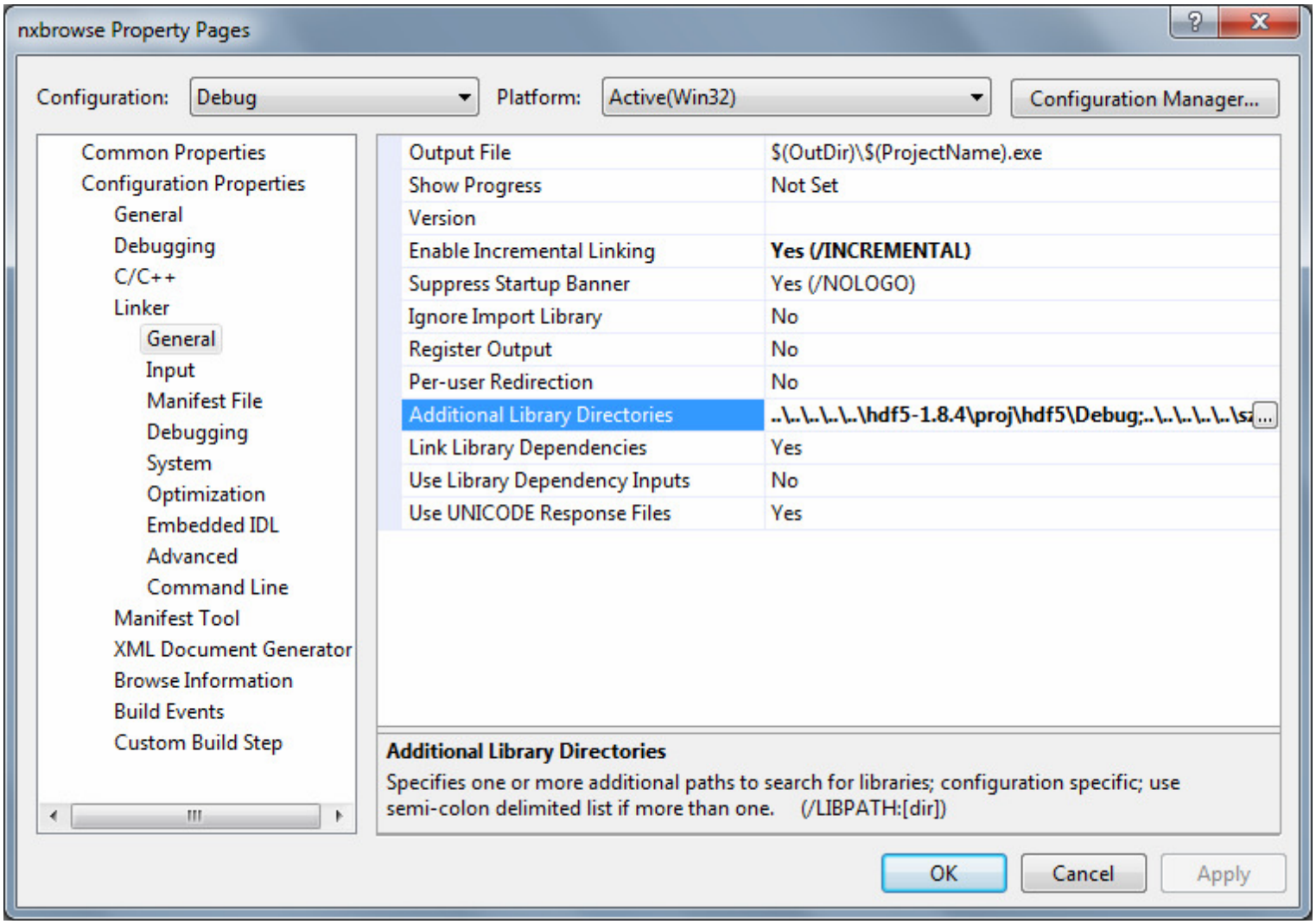


To link a NeXus application with a underlying library, two actions must be done, add the library name and path for each application:

1.1 Add the library name to be linked for each application

To library name to be linked is entered in the Linker Input Property Page (Figure 4.1), in the Additional Dependencies field. Either the literal library name or a Windows system environment variable[1] with the library name can be used. Figure 4.1 shows an example of NXBrowse linked with the HDF5 library: The

Figure 4.2: Linker General, Additional Library Directories



following environment variables were defined, for the HDF5, ZLIB and SZIP libraries[2], HDF5_LIB, HDF5_EXT_ZLIB, HDF5_EXT_SZIP with the names of those libraries. Then, in the NeXus application Linker Input Property Page, in the Additional Dependencies field, the following must be entered

[1] To define an environment variable in Windows 7, click Start, right-click Computer, and select Properties. Select Advanced System Settings. Then, click the Environment Variables button.

$(HDF5_LIB) $(HDF5_EXT_ZLIB) $(HDF5_EXT_SZIP)

1.2 Add the library path to be linked for each application

This is done in the Linker General, Additional Library Directories field, shown in Figure 4.2

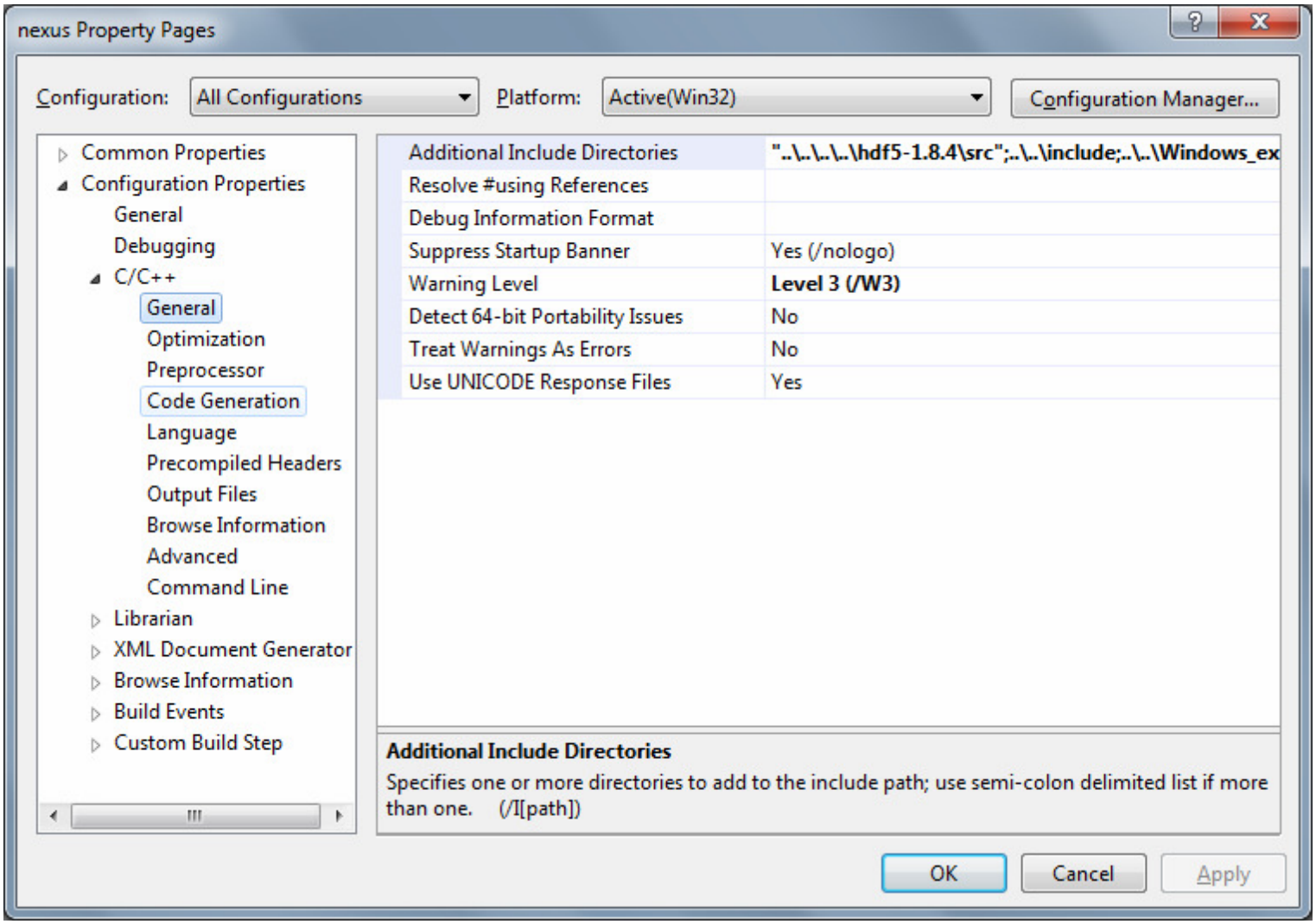
2 Nexus library project settings

To compile the NeXus library with any of the HDF5, HDF4 or NXXML underlying libraries, the following steps must be done in the Visual Studio editor.

2.1 Add the library path of the underlying library

This is done in the C/C++, General, Additional Include Directories of the **nexus** project file, Figure 5.1.

Figure 5.1: the C/C++, General, Additional Include Directories

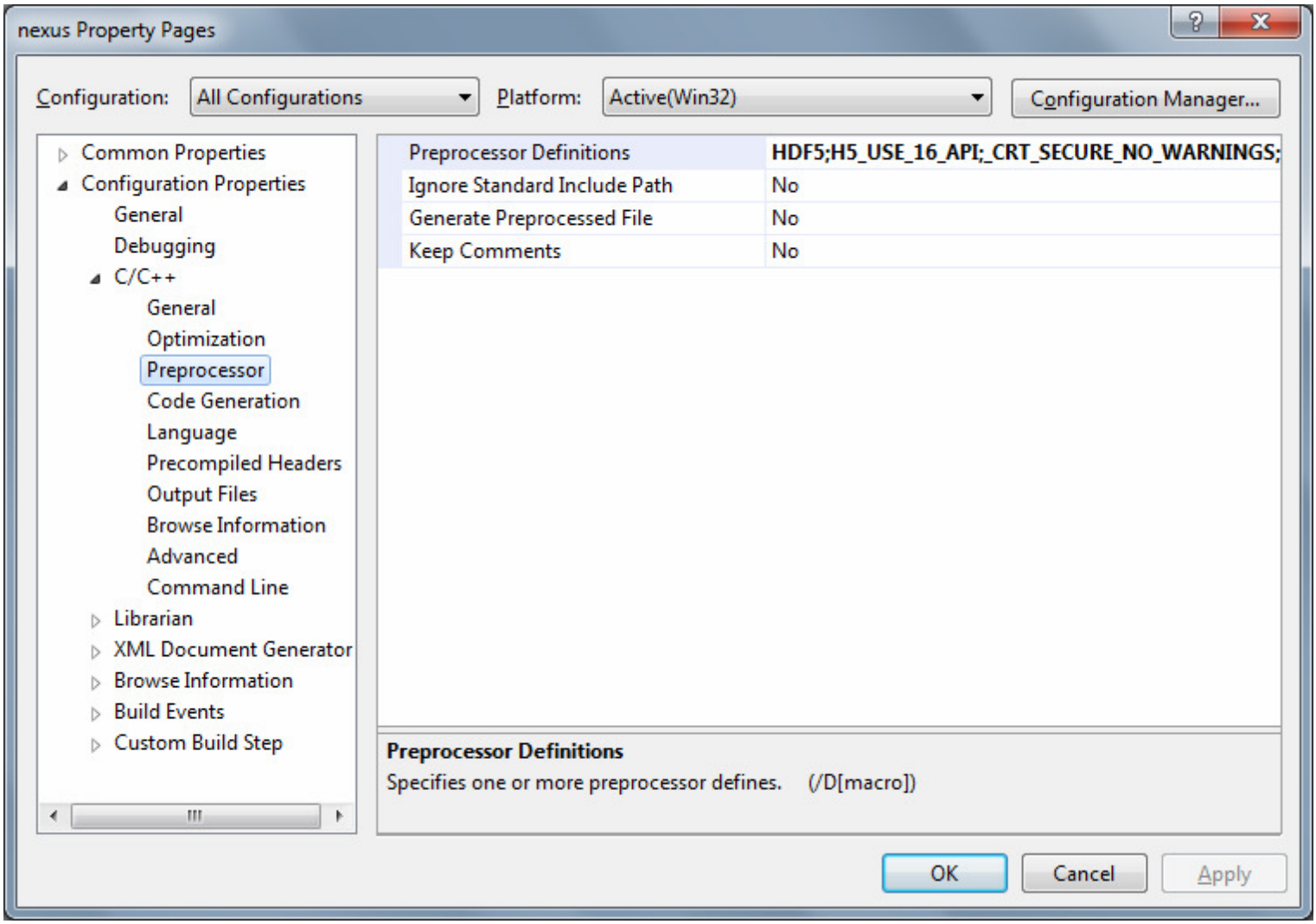


[2] The HDF5 library needs the SZIP and ZLIB libraries

2.2 Add a preprocessor definition for the underlying library

This is done in the C/C++, Preprocessor, Preprocessor Definitions field of the **nexus** project file, Figure 5.2.

Figure 5.1: the C/C++, Preprocessor, Preprocessor Definitions



For compiling with the HDF5 library, use the following definitions: HDF5, H5_USE_16_API. For compiling with the HDF4 library, use the following definition: HDF4. For compiling with the NXXML library, use the following definition: NXXML.

These symbols are used in the NeXus library C source code files, to check if the code calling each underlying library should be compiled. This is achieved in the files napi5.c, napi4.c, nxio.c and nxxml.c by means of the following conditional compilation preprocessor directives at the beginning of each source file, #ifdef HDF5, #ifdef HDF4, #ifdef NXXML. This means that, for example if the HDF5 symbol is defined, the code inside the file napi5.c is not compiled at all. This is a change of the source code being introduced with this porting for Visual Studio.