Instructions for building PKIF

- Install Visual Studio .NET 2005
- Download and install Microsoft Windows SDK for Windows 7 from http://www.microsoft.com/downloads/en/details.aspx?FamilyID=35aeda01-421d-4ba5-b44b-543dc8c33a20
- Add include and lib folders from Windows SDK for Windows 7 to Visual Studio using the Tools -> Options -> Projects and Solutions -> VC++ Directories dialog. Make sure they are at the top of the list.
- Install boost version 1_44. For installation instructions visit www.boost.org.
    - Add the following guard to line 1191 of boost/filesystem/v2/operations.hpp: #if !defined(BOOST_WINDOWS_API) || defined(BOOST_FS_HARD_LINK) as referenced in https://svn.boost.org/trac/boost/ticket/4630
    - Specify BOOST_HOME environment variable pointing to the boost installation location.
    - Specify BOOST_INCLUDE environment variable pointing to the boost include location.
- Install Python. It can be downloaded from http://www.activestate.com/
- Install Perl. It can be downloaded from http://www.activestate.com/
- Install latest Java 5.0 JDK (Needed to build JPKIF). It can be downloaded from http://java.sun.com/javase/downloads/index_jdk5.jsp.
    - Specify JAVA_HOME environment variable pointing to the location of java jdk. Place the bin folder of java installation in your path.
- Download and extract swig version 2.0.1 (Needed to build JPKIF and C# PKIF code). It can be downloaded from http://www.swig.org/. Place the extracted swig in your path.
- Download and build wxWidgets from http://www.wxwidgets.org/ version 2.8.11 (Needed by PKIF Resources).
    - Specify WXWIN environment variable and point it to the location of the wxWidgets.

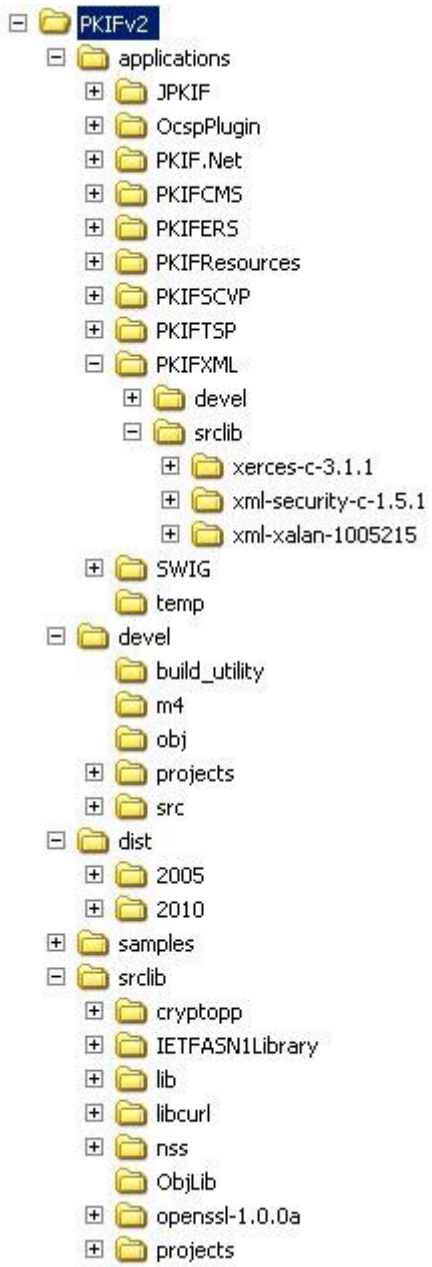
- Build NSS
    - Download and install the current MozillaBuild package from: http://ftp.mozilla.org/pub/mozilla.org/mozilla/libraries/win32/MozillaBuildSetup-Latest.exe
    - Launch `start-msvc8.bat` batch file located in MozillaBuild installation directory.
    - cd into <project>\srclib\nss directory.
    - Enter perl grabbuild.pl . build debug for building debug
    - Enter perl grabbuild.pl . build release for building release
    - Create an environment variable NSSOSVERSION indicating operating system version (WINNT5.1, WINNT6.0)
- Build OpenSSL
    - Download openssl-1.0.0c source code from http://www.openssl.org/ and extract it into <project>\srclib directory.
    - From <project>\srclib\openssl-1.0.0c directory execute following commands
        - perl Configure VC-WIN32
        - ms\do_ms.bat
        - nmake /f ms\nt.mak

        - copy out32*.lib into <project>\srclib\lib\2005
        - nmake /f ms\ntdll.mak (needed by PKIFXML)
    - Create an environment variable OPENSSL_HOME pointing to <project>\srclib\openssl-1.0.0c
    - Create an environment variable PKIF_SRCLIB and point it to the <project>\srclib directory
    - Create an environment variable OBJLIB_HOME and point it to the <project>\srclib\ObjLib directory
- Build PKIF dependencies
    - Download PKIFv2_1-XXX-srclib.zip from the PKIF sourceforge site.
    - Extract the contents into the <project>\srclib folder (the srclib folder will now contain nss, openssl-1.0.0c, cryptopp, IETFASN1Library, ObjLib, libcurl, projects, lib folders)
    - Open the BuildSrcLibDependencies.sln located in <project>\srclib\projects\2005 folder using Visual Studio 2005.
    - Build srclib dependencies Release and/or Debug target
    - Create an environment variable CRYPTOCROOT and point it to <project>\srclib\cryptopp folder.
- Build PKIFv2
    - Download PKIFv2_1-XXX-source.zip from the PKIF sourceforge site.
    - Extract the contents so the devel, applications, and dist folders are peer to <project>\srclib folder
    - Create an environment variable PKIF_HOME and point it to the <project>\dist/2005 directory
    - Open PKIF.sln located in <project>\devel\projects\2005 folder using Visual Studio 2005
    - Build PKIF Release and/or Debug target
- Build PKIFCMS, PKIFTSP, PKIFResources, PKIFERS, and PKIFSCVP
    - Open PKIFCMS.sln located in <project>\applications\PKIFCMS\proj\2005 folder using Visual Studio 2005
    - Build PKIFCMS Release and/or Debug target
    - Open PKIFTSP.sln located in <project>\applications\PKIFTSP\proj\2005 folder using Visual Studio 2005
    - Build PKIFTSP Release and/or Debug target
    - Open PKIFERS.sln located in <project>\applications\PKIFERS\proj\2005\ PKIFERS folder using Visual Studio 2005
    - Build PKIFERS Release and/or Debug target
    - Open PKIFSCVP.sln located in <project>\applications\ PKIFSCVP \proj\2005\ PKIFSCVP folder using Visual Studio 2005
    - Build PKIFSCVP Release and/or Debug target
    - Open AllResources.sln located in <project>\applications\PKIFResources\proj folder using Visual Studio 2005
    - Build PKIFResources Release and/or Debug target
- Build PKIF JAVA wrappers
    - Download JPKIF2_1-XXX-source.zip from the PKIF sourceforge site.
    - Extract the contents into <project>\applications folder.

    - Create a directory JPKIFsln in <project>\applications and copy JPKIF.sln file there from JPKIF folder
    - Open <project>\JPKIF\ JPKIF.sln using Visual Studio 2005.
    - Build PKIF Java Release and/or Debug target
- Build PKIF C# wrappers
    - Download PKIF.Net2_1-XXX-source.zip from the PKIF sourceforge location.
    - Extract the contents into <project>\applications folder. Note: If you are building PKIF JAVA, SWIG folder was already created by the step above; just copy the contents of the extracted swig folder into existing one.
    - Open <project>\applications\PKIF.Net\PKIF.Net.sln using Visual Studio 2005.
    - Build PKIF .Net Release and/or Debug target
- Build PKIFXML
    - Download PKIFv2_1-XXX-PKIFXML-srclib-xalan.zip, PKIFv2_1-XXX-PKIFXML-srclib-xerces.zip, PKIFv2_1-XXX-PKIFXML-srclib-xmlsec.zip from PKIF sourceforge site.
    - Extract the contents into <project>/applications\PKIFXML\srclib folder.
    - Build xerces Release and/or Debug target using the solution provided in the project folder .
    - Create XERCESCROOT environment variable and point it to <project>\applications\PKIFXML\srclib\xerces-c-src_2_7_0
    - Build xalan Release and/or Debug target using the solution provided in the project folder.
        - Add xerces dlls to the PATH to generate LocalMsgData.hpp
    - Create XALANCROOT environment variable and point it to <project>\applications\PKIFXML\srclib\xml-xalan\c
    - Build xmlsec Release and/or Debug target using the solution provided in the project folder.
    - Create XMLSECROOT environment variable and point it to <project>\applications\PKIFXML\srclib\xml-security-c-1.3.1
    - Download PKIF2_1-XXX-PKIFXML-source.zip and extract the devel folder into <project>\applications\PKIFXML.
    - Open PKIFXSECProvider.sln located in <project>\applications\PKIFXML\devel\proj\2005 using Visual Studio 2005.
    - Build PKIFXSECProvider Release and/or Debug target

NOTE: If you encounter a linker error LNK1103 indicating that debugging symbols for a standard system library are corrupt.

i.e. shell32.lib(shguid.obj) : fatal error LNK1103: debugging information corrupt; recompile module
This is a known bug with a hotfix available here: http://support.microsoft.com/kb/949009



Resulting PKIF tree

*Note XXX indicates current release.