**Building an Agent Plugin Project in Visual Studio 2010**

The Microsoft C++ libraries installed with Visual Studio 2010 have some breaking changes over the libraries installed with Visual Studio 2008. This means that binaries compiled using Visual Studio 2008 will cause linker errors when you try to use them to compile a C++ project in Visual Studio 2010. As of June 2012, the NS Agent libraries are still built against the VS 2008 versions of the Microsoft C++ libraries. Consequently, an attempt to build an Agent Plugin project in VS 2010 will result in linker errors.

The SDK Team has created a temporary workaround to allow Agent Plugin projects developed in VS2010 to build against the VS2008-based Agent libraries. This document explains how to use this workaround in steps 8-10, below.

First, however, you may need to update some path settings in VS2010. Depending on when you began development of your Agent Plugin project in VS2010 or when you converted your VS2008 project to VS2010, your settings may not match the installed SDK paths. Steps 1-7 below explain this update process.

 **\*\*\*\*\*\*\*\*\*\*\*\*\*\* I M P O R T A N T ! \*\*\*** \*\*\*\*\*\*\*\*\*\*\*\*\*

The following workaround will allow you to build your Agent Plugin project so that you can continue developing and testing your solution in Visual Studio 2010. However, this workaround will result in undefined behavior in the event of code that either (1) attempts to create or use a *string* object whose length exceeds the maximum length as defined by the C++ libraries, OR (2) attempts to access a character within a *string* object by using an illegal index (such as *-1*).

**Therefore, it is important that, before releasing your solution to final QA and production, you remove any reference to the workaround library file discussed below, and instead build your plugin project against the (future) build of the NS Agent binaries done with the VS2010 compiler.**

Note, however, that you should leave the path settings as you have modified them in steps 3-7, below.

  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

To build a VS2010 Agent Plugin project, do the following on your development machine:

1. Open a command line window and run *set*. Check the value for *AltirisSDKRoot*. If the value of this environment variable does not have a backslash(“*\*”) on the end, it needs to be added. To do this, at the command prompt type: ***set AltirisSDKRoot=%AltirisSDKRoot%\***
2. Make sure you have the latest version of the *Visual Studio 2010 Integration* module installed. To do this, run the Symantec SDK Installer. (You probably already have the SDK Installer on your development machine; if not, you can install it by running [*\\ali-netapp1.linus.sen.symantec.com\altirissdk\SDKComponents\setup\setup.exe*](file:///%5C%5Cali-netapp1.linus.sen.symantec.com%5Caltirissdk%5CSDKComponents%5Csetup%5Csetup.exe)*.*) Navigate in the left pane of the *Simple* tab to *Altiris Platform -> 8.0 -> Common Modules -> Visual Studio 2010 Integration*. If a newer version is available, right-click the node and select *Install* from the menu, then click the *Install* button in the lower right corner. Click *Update* in the next dialog to install the latest module.
3. **NOTE:** The following steps regarding paths to link to NSCore Agent binaries apply to SMP 7.5 binaries. If you are still developing against SMP 7.1 binaries, make sure that the paths exist as specified, but do **\*not\*** add “*Release*” or “*Debug*” to the end of the paths.

Open your solution in Visual Studio 2010. Make sure that your build configuration is set to *Release* mode. In Solution Explorer, right-click on your Agent Plugin project node, and select *Properties*. In the left pane of the Property Pages window, navigate to *Configuration Properties -> Linker -> General*. Select the *Additional Library Directories* property in the right pane and go into edit mode by clicking on the down arrow in the right-hand column and selecting *<Edit…>*. In the editor dialog, you should see an entry for the path of the version of the NSCore that you are building your agent plugin project against, in the form of *$(AltirisSDKRoot)Lib\[Win64|Win32]\NSCore\[Version].* To this entry, add “*\Release*” to the end of the path, so it now reads in the form ***$(AltirisSDKRoot)Lib\[Win64|Win32]\NSCore\[Version]\Release***. For example, the entry should now read something like: ***$(AltirisSDKRoot)Lib\Win64\NSCore\7.5.1213\Release*** . Click *OK* to exit the editor.

1. Now navigate in the Property Pages to *Configuration Properties -> C/C++ -> General*. Select the *Additional Include Directories* property in the right pane and open the editor as in step 3. Check to see if there is an entry whose path exactly matches the entry you added in step 3, above. If not, you need to add it. In other words, add a new entry in the ***$(AltirisSDKRoot)Lib\[Win64|Win32]\NSCore\[Version]\Release*** format, just as you did in step 3. (Do not delete any existing paths, just add this new one.) Notice that you are adding a “*lib*” path into the *Additional Include Directories.* This may not seem logical, but it is necessary so that the C++ compiler’s preprocessor can find the required type libraries (\*.tlb files).
2. With the Property Pages window still open, now change the Configuration setting of the project to *Debug* (do this in the Configuration list in the upper left corner of the Property Pages dialog). Repeat the steps 3 and 4, but this time use “*\Debug*” at the end of the path instead of “*\Release*”.
3. If you plan to build both x86 and x64 versions of your Agent Plugin, change the Platform setting at the top of the Property Pages window to the other target architecture and repeat the process of steps 3 and 4 for both *Debug* and *Release* mode for the new architecture.
4. When you have finished updating the paths in the Property Pages, click *OK*.
5. Now you need to add the “workaround” library to your Agent Plugin project. There are two versions of this library file, one targeting x86 agents and another targeting x64 agents. The x86 library is at *C:\Program Files (x86)\Altiris\Altiris SDK\Lib\Win32\ libvisual2008\_visual2010\_x86.lib* and the x64 version is at *C:\Program Files (x86)\Altiris\Altiris SDK\Lib\Win64\ libvisual2008\_visual2010\_x64.lib*. Choose the version that is appropriate for the target architecture of your Agent Plugin build, and copy the library file to your local Plugin project. Put the copy under the root directory of your Agent Plugin project (where you project source files are located). Next, in Solution Explorer, right-click on the *DLL* project sub-folder in your plugin project, and select *Add -> Existing Item.* Select the workaround library that you just copied into your project, and click the *Add* button.
6. Rebuild your Agent Plugin project, and the linker errors should be gone.
7. If you do a second build of your Plugin which targets a different architecture (i.e., you want builds of your Plugin for both x86 and x64), then you will need to remove the existing workaround library from your project, copy the version of the workaround library that targets the new architecture into your project directory, and add it to the project. Then build again.