An MTConnect CUSTOM ACTION for Windows Installer Scripts (msi)

This document briefly describes the process of creating a Visual Studio Setup and Deployment project (now obsolete). Readers can look at <http://bonemanblog.blogspot.com/2005/11/custom-action-tutorial-part-ii.html>

for a more detailed explanation.

## Creating Custom Action DLL Project

Step 1 is to create a C++ win32 Dll MyCustomAction project to handle Install/Uninstall options. Eventually you will want to ATL/WTL capabilities so we can display windows (static linking to ATL lib) but these are best done manually.

Step 2 is to modify the MyCustomAction.def DLL definitions file, which should be written as:

; MyCustomAction.def

;

; defines the exported functions which will be available to the MSI engine

LIBRARY "MyCustomAction"

EXPORTS

Install

Commit

Rollback

Uninstall

And the basic MyCustomAction.cpp file should contain:

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul\_reason\_for\_call, LPVOID lpReserved )

{

return TRUE;

}

extern "C" UINT \_\_stdcall Install(MSIHANDLE hInstall){return ERROR\_SUCCESS;}

extern "C" UINT \_\_stdcall Commit(MSIHANDLE hInstall){return ERROR\_SUCCESS;}

extern "C" UINT \_\_stdcall Rollback(MSIHANDLE hInstall){return ERROR\_SUCCESS;}

extern "C" UINT \_\_stdcall Uninstall (MSIHANDLE hInstall){return ERROR\_SUCCESS;}

You will need to add the MSI functionality to the source file to compile and link:

#include <msi.h>  
#include <msiquery.h>

#pragma comment(lib, "msi.lib")

with the following explanation of each step.

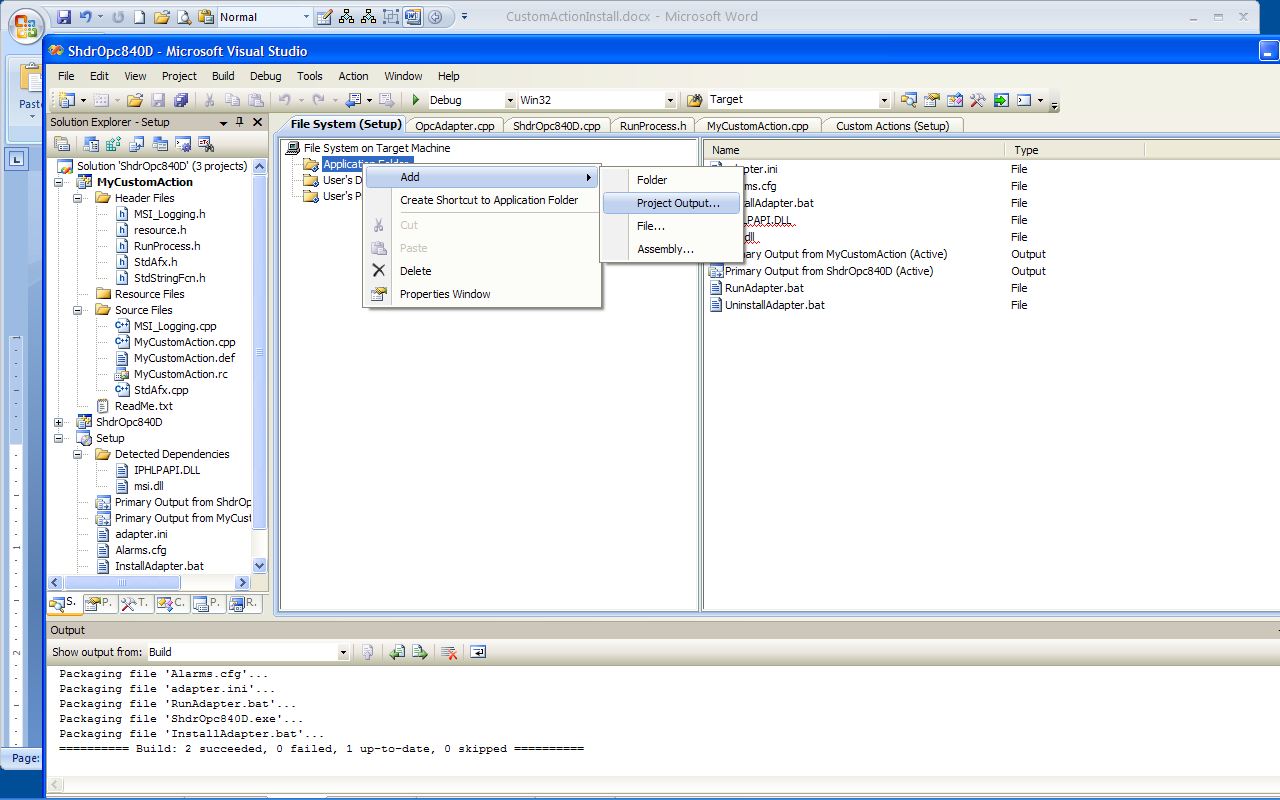
Install

Commit **-** Custom actions are executed upon successful completion of the installation script. If the [InstallFinalize action](http://msdn.microsoft.com/en-us/library/windows/desktop/aa369505(v=vs.85).aspx) is successful, the installer will then run any existing Commit Custom actions. The only mode parameter the installer sets in this case is MSIRUNMODE\_COMMIT.

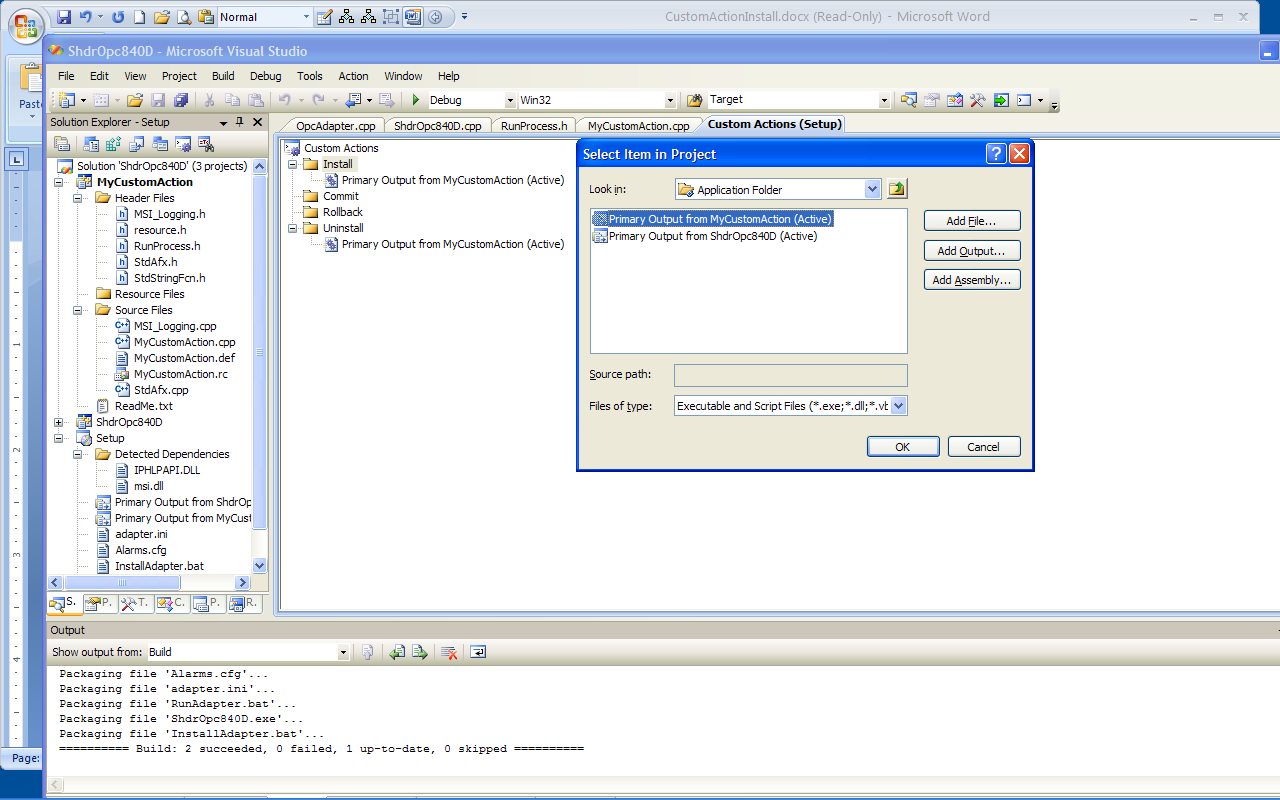
Rollback - If an installation is unsuccessful, the installer attempts to rollback the changes made during the installation and restore the original state of the computer. A rollback custom action is a type of [deferred execution custom action](http://msdn.microsoft.com/en-us/library/windows/desktop/aa368268(v=vs.85).aspx), because its execution is deferred when it is invoked during the installation sequence. It differs from a regular deferred custom action in that it is only executed during a rollback. A rollback custom action must always precede the deferred custom action it rolls back in the action sequence. A rollback custom action should also handle the case where the deferred custom action is interrupted in the middle of execution. For example, if the user were to press the Cancel button while the custom action was executing.

## Insert Custom Action DLL into MSI Setup Project

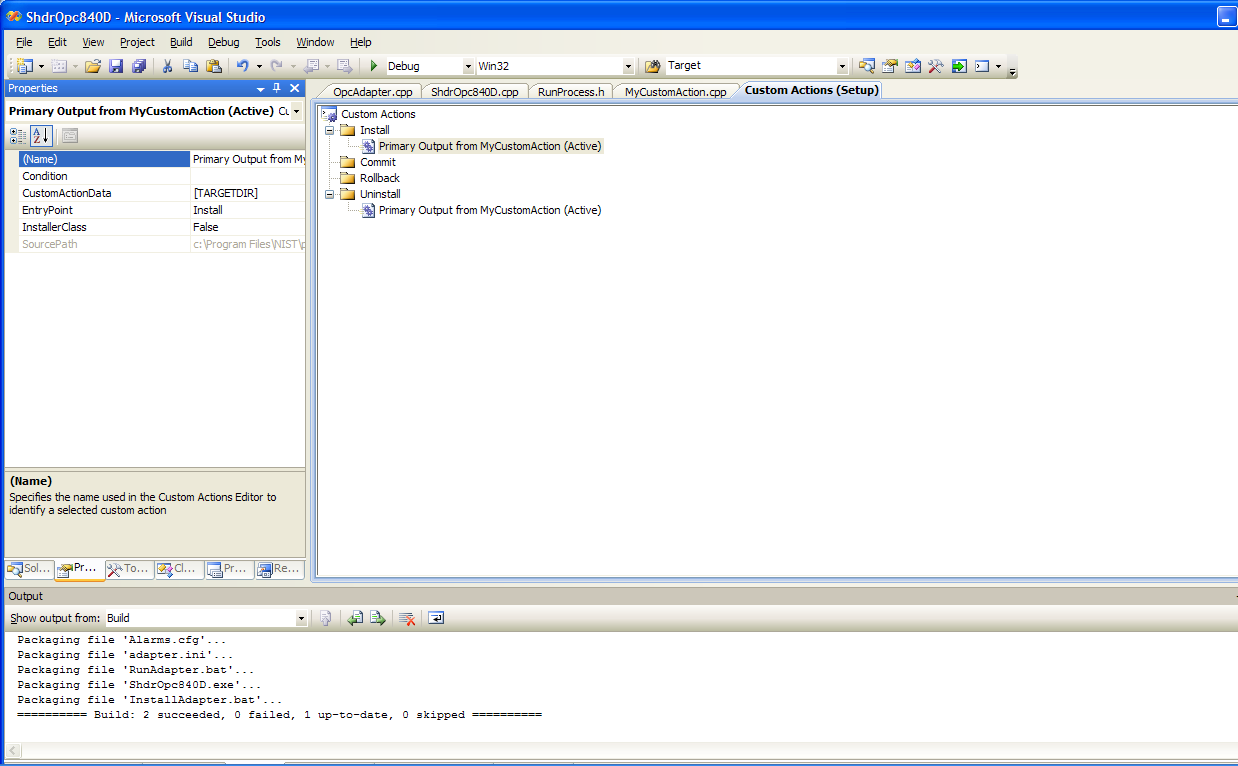
Compile and then insert the DLL into setup project file area as a Project Output:



Then add the DLL to the Custom Action to the Install folder by right clicking (ADD) and then selecting it from the browse folder dialog.



Note the entry point point matches the Install subroutine. If you will want to pass parameters, you will need to add them to the CustomActionData field, in the screen shot case below, we will send in [TARGETDIR] - note capitalization and brackets are requires. You can only read the entire CustomActionData field in the DLL Install method.



Here is the code to read the CustomActionData (note capitalization) in the DLL project

extern "C" UINT \_\_stdcall Install(MSIHANDLE hInstall)

{

TCHAR szBuffer1[MAX\_PATH] = {'0'};

DWORD dwLen = MAX\_PATH;

// Returns ERROR\_SUCCESS, ERROR\_MORE\_DATA, ERROR\_INVALID\_HANDLE, ERROR\_BAD\_ARGUMENTS

UINT hr = MsiGetProperty(hInstall, \_T("CustomActionData"), szBuffer1, &dwLen);

std::string path(szBuffer1);

}

Note, each time you MsiGetProperty reset dwLen to the MAX\_PATH. You can search the internet for ways to read large portions of CustomActionData.

Using this path, you can then configure an Installation file for example.

## Reinstalling Issues

Click the Project Icon, (below Setup), then make sure no Install item is open, then click Properties tab next to solution

DetecNewerInstallation - True

To make sure custom action is invoked, Set RemovePreviousVersions - True

