

Description: Guide to updating Visual Studio 6.0 (C++) projects created with ProjectWorX32 version 9 project add-on wizard.

OS Requirement: Win 2000, XP Pro, Server 2003, Vista, Server 2008, Windows 7

General Requirement: GENESIS32 version 9.x installed with all of the latest service packs, Microsoft Visual C++ Version 6.0, and a general knowledge of C++ Programming.

Introduction

Among the many improvements to GENESIS32 version 9.x was a trimming down of reliance upon the registry and subsequently, the elimination of the requiring administrative privileges to run GENESIS32 products. Although ProjectWorX32 continues as a tool for administrators and is exempt from this change, it too has weaned itself off of the registry and made use of the ICONICS IcoSetup.ini setting storage used across all version 9.x products. Among the items that are affected by this change from the registry to IcoSetup.ini file are all of the internal and customer-based ProjectWorX32 add-ons. The remainder of this document will guide you in the process of modifying existing projects to comply with ProjectWorX32 version 9.x. Projects that are not updated will not appear in version 9.x.

Getting Started

The first think you will need to do is to locate the directory that houses your ProjectWorX32 add-on projects (Visual C++ 6.0). In that directory, you will need to extract the contents of the UpgradeFiles.zip.



Figure 1 - Add Files to Your Project Directory

NOTE: Please contact ICONICS Technical Support for the UpgradeFiles.zip mentioned above.

Adding the New Libraries

- Open your workspace in Microsoft Visual Studio 6.0 and right-click on your project (just beneath the workspace) and

select the “New Folder” menu item. You want to create a new folder called LIB, containing extensions of .lib.



Figure 2 - Adding the LIB Folder

- Once you have created this folder, you want to add a couple of library files to it. Right-click on the new LIB folder, and select “Add Files to Folder”.
- In the “Add File” dialog that comes up, you want to select “Library Files (lib)” next to “Files of Type”, then select “PwXReg.lib” and “UnifiedSetupStorage.lib”(these are items that were extracted from the UpgradeFiles.zip file).

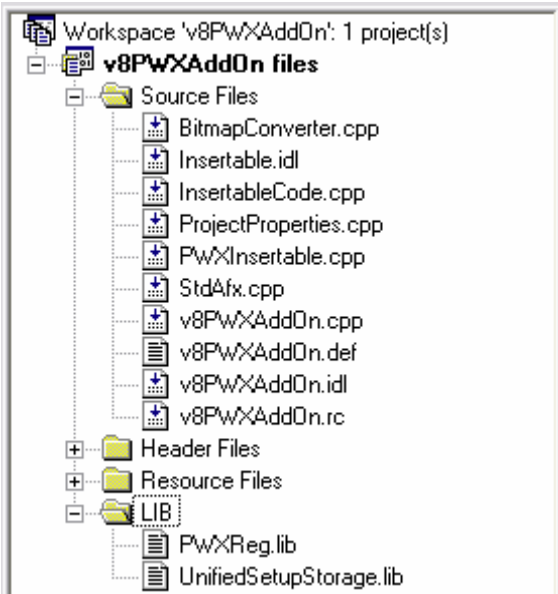


Figure 3 - Library Files Added

Code Changes

All necessary code changes will take place in the main.cpp file of your project ([ProjectName].cpp). The only things you need to do are to add a #include reference to PwXReg.h and remove both definition and implementation of “AddPwxRegistry” and “SubtractPwxRegistry”. Your code near the top of the file should look similar to the code below (changes required are in bold):

```
#include "stdafx.h"
#include "resource.h"
#include <initguid.h>
#include "v8PWXAddOn.h"

#include "Insertable_i.c"
#include "v8PWXAddOn_i.c"
#include "PWXInsertable.h"
#include "PWXReg.h"

CComModule_Module;

BEGIN_OBJECT_MAP(ObjectMap)
OBJECT_ENTRY(CLSID_PWXInsertable, CPWXInsertable)
END_OBJECT_MAP()
```

```
////////////////////////////////////////
// DLL Entry Point
/*
void AddPWXRegistry(const TCHAR* KeyName);
void SubtractPWXRegistry(const TCHAR* KeyName);
*/
```

You will also need to delete or comment out the implementation of the functions `AddPWXRegistry()` and `SubtractPWXRegistry()`. Their implementations are found at the end of the source file.

Compiling

You are now ready to compile your project. It is strongly recommended that you compile your project as non-Unicode. If you still have your default configurations from the time the wizard ran, you should select the “Win32 – Release MinDependency” as the active configuration (or Win32 Debug if you wish to debug).

If you have set up different configurations than the default, it is recommended that you remove the `_UNICODE` flag from whichever configuration you intend to compile as the active one.

If you have never compiled under the “Release MinDependency” configuration, you may get a compile error saying that “main” is not defined. This is a known issue in Visual Studio 6.0. To fix this, you need to go into the project properties (Project → Settings). On the C/C++ tab, in the “General” Category, you will need to remove `ATL_MIN_CRT` from the list of preprocessor definitions.

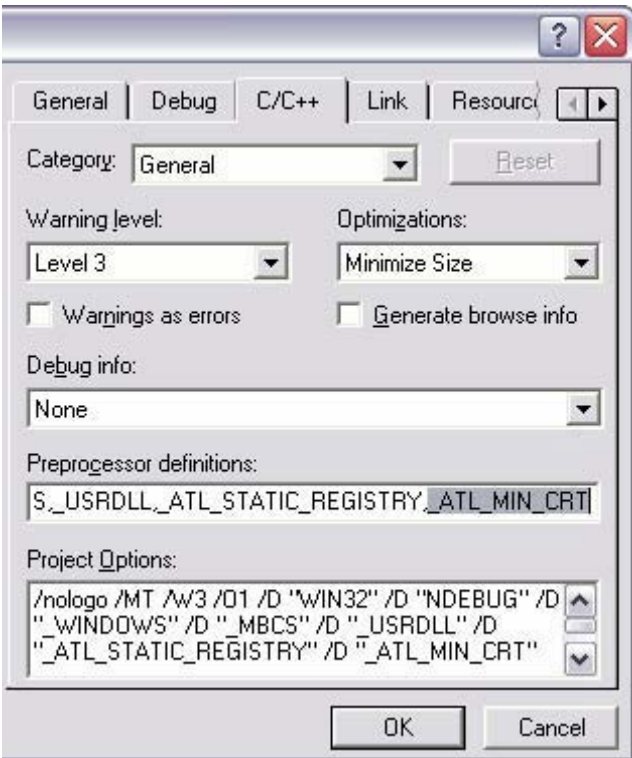


Figure 5 - Removing ATL_MIN_CRT for Smoother Compiling

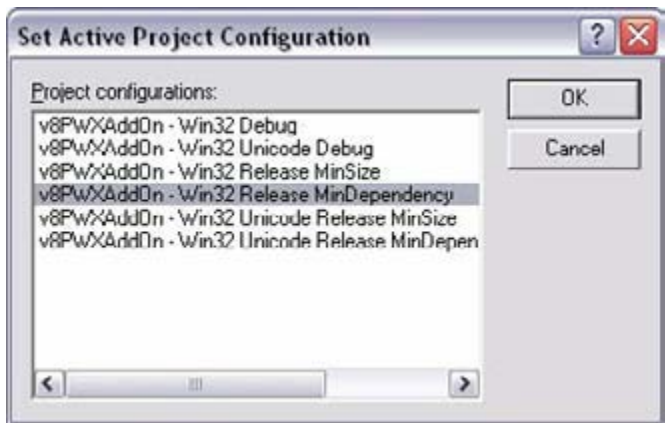


Figure 4 - Recommended Build Configuration