Fundamentals of ADS™

Environment: Windows™/Visual C++™

Introduction

This overview of an ADS development environment for Microsoft® Visual C++ is for the participants in the Autodesk Training Department's course on *Fundamentals of the AutoCAD Development System*.

AutoCAD® Release 12 for Windows supports the Microsoft Visual C++ development environment for real mode ADS applications under Microsoft Windows 3.1 through the ADS library *winadsc7.lib*.

This document describes how to set up a **project** for an ADS application in Visual C++. It describes the **compiler and linker settings** necessary to successfully create ADS binaries in this environment. It explains how to **debug an ADS application** using the built-in debugging environment with the **Watch** window.

This document is based on AutoCAD Release 12 for Windows and Microsoft Visual C++ Version 1.0 for Windows 3.1.

Project Setup

Follow each step to set up a new ADS application project in Visual C++.

1. Create the necessary source, resource and definition files.

- C language application source file, e.g., *hello.c*
- Windows definition file, e.g., *hello.def*
- Windows resource file, e.g., *hello.rc*

Use either the Windows File Manager or an MS-DOS window to copy the files *template.def* and *template.rc* in the *student* directory to the name of your application source file, leaving the extensions intact. For example, if the current application source file is *hello.c*, copy *template.def* and *template.rc* to *hello.def* and *hello.rc*.

Edit the definition and resource files for your application and replace the word "template" with the name of your application. For example, edit *hello.def* and *hello.rc* with your text editor and replace all occurrences of the word "template" with the word "hello".

2. Create a new project file.

Name the project by choosing New from the Project menu. The New Project dialogue appears.

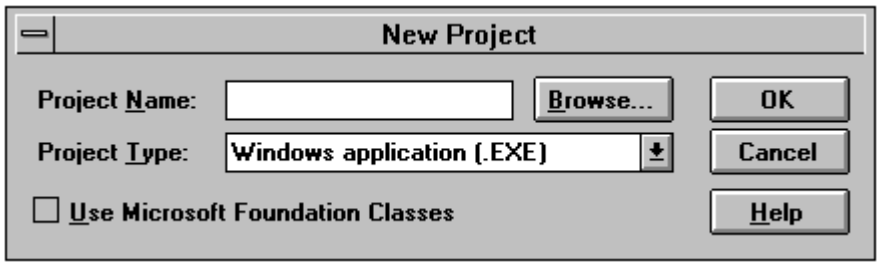

Figure 1. New project dialogue

In the *student* directory, create a new project with the appropriate name, e.g., create a new project named *hello*. Do not use Microsoft Foundation Classes.

3. Set the project type to `Windows application (.EXE)`. This is the default.

4. Add the necessary files to the project. From the *student* directory, he application's course, definition and resource files, e.g.:

- *hello.c*
- *hello.def*
- *hello.rc*.

From the *acadwin/ads* directory:

- *winads.c*
- *ddeml.lib*
- *winadsc7.lib*

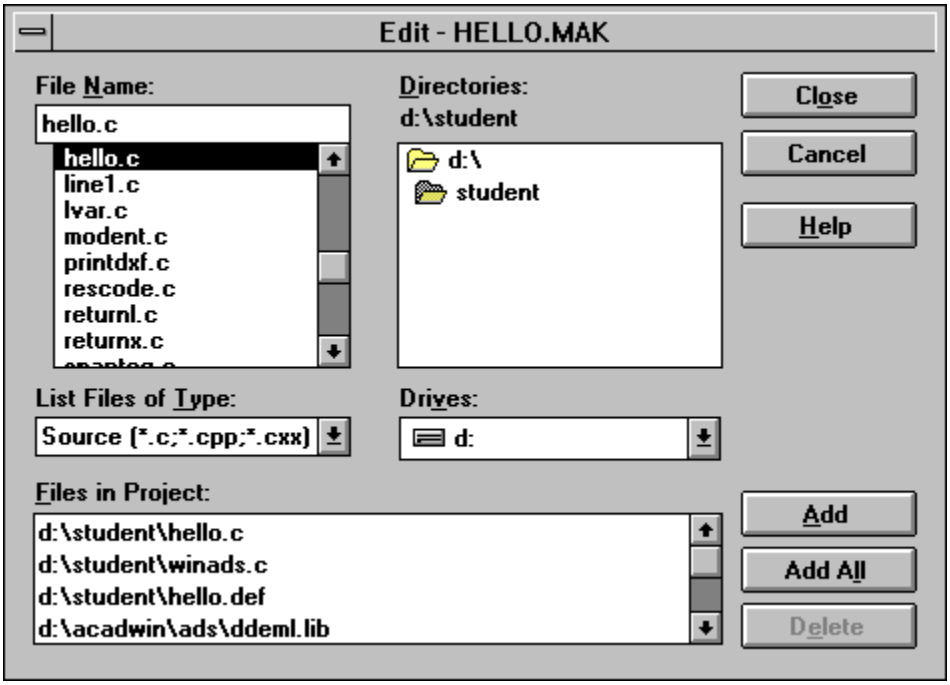

Figure 2. Adding or deleting files in a project

Pick the Close button to close the dialogue.

Choose Edit from the Project menu if you need to add or remove files from an existing project.

5. Set up the appropriate compiler options.

Choose Project from the Options menu.

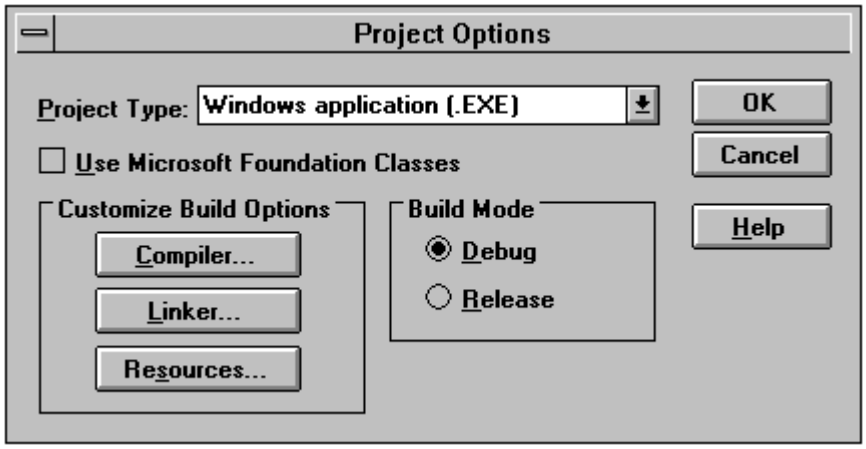

Figure 3.Project options dialogue

Pick Compiler from the Project Options dialogue.

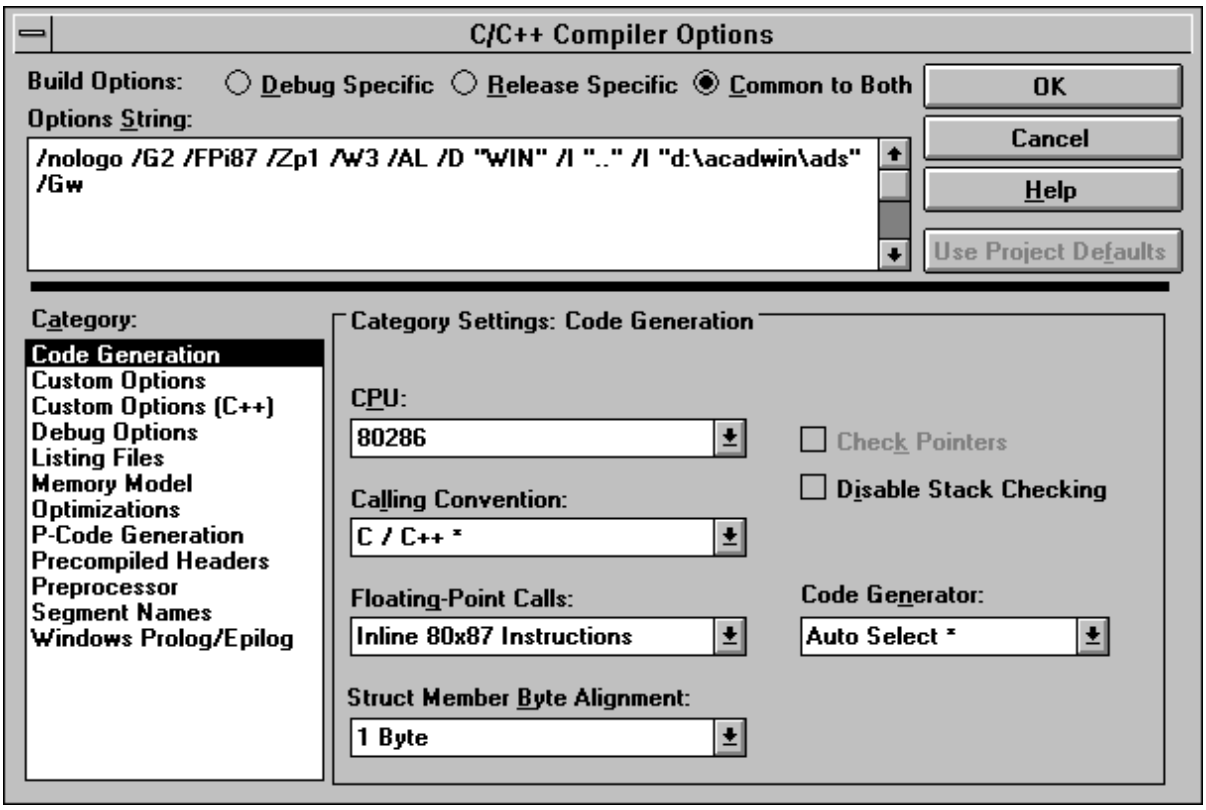

Figure 4. Compiler options dialogue

Set Build Options to Common to Both. This will apply the settings you change to both release and debug builds of the project.

Under Code Generation, set:

Option	Setting
CPU	80286
Floating-Point Calls	Inline 80x87 Instructions
Struct Member Byte Alignment	1 Byte

Table 1. Code generation settings

Under Custom Options, set:

Option	Setting
Other Options:	-DWIN

Table 2. Custom options settings

Under Listing Files, set:

Option	Setting
Include Local Variables	Toggle OFF
Browser Information	Toggle OFF

Table 3. Listing files options

Under Memory Model, set:

Option	Setting
Model	Large

Table 4. Memory model options

Under Preprocessor, set:

Option	Setting
Include Path:	.. \acadwin\ads

Table 5. Preprocessor options

Under Windows Prolog/Epilog, set:

Option	Setting
Generate Prolog/Epilog For	Real Mode __far Functions

Table 6. Windows prolog/epilog options

Miscellaneous Issues

ads.ico File

This file comes in the *acadwin/ads/win* directory. It is referenced by the Windows resource file, or *rc* file, for each ADS project. *ads.ico* must reside in the build directory for your project; in this case, the *student* directory.

Project Build

To Build an Entire Project Into an Executable

Choose Rebuild from the Project menu.

To Build Only Changed Files Into an Executable

Choose Build from the Project menu.

Debugging

To debug an ADS application interactively within the Visual C++ editor, follow these steps.

1. Set a breakpoint in the ADS application source code.

You can set a breakpoint using either the "open palm" icon in the Visual C++ toolbar, or by selecting Debug Breakpoints... from the menu. A good place to set a breakpoint is in the initialization code for the ADS application, as shown below in bold type.

```
{
    int stat;
    short scode = RSRSLT;             /* This is the default result code */

    ads_init(argc, argv);             /* Initialize the interface */

    for ( ;; ) {                      /* Note loop conditions */

      if ((stat = ads_link(scode)) < 0) {

         printf("TEMPLATE: bad status from ads_link() = %d\n", stat);

         /* Can't use ads_printf to display
            this message, because the link failed */
         fflush(stdout);
         exit(1);
      }
```

2. Load the application into AutoCAD.

Use the AutoLISP `xload` function to load the program into the AutoCAD editor. After the program is loaded, the debugger will take over when it reaches the line of code where a breakpoint was set.

Step through the initialization code until you can enter a command in AutoCAD.

Use the F8 key or select Debug Step Into from the menu to step through each line of code in the debugger. When you are no longer able to step through code, AutoCAD will be quiescent, that is, ready to accept a command.

3. Give a command defined by the ADS application.

Enter a command or function defined within the ADS application course at the AutoCAD command prompt. Control will return to the debugger, and you can step through the code a line at a time.

Interactive requests for data in the AutoCAD drawing editor, e.g., `ads_getpoint()` function requests, can be satisfied by switching focus to the

AutoCAD window and responding normally to the request when you execute an appropriate line of code within the debugger.

4. Use the Watch window to check values.

While actively debugging the ADS application, select Window Watch from the menu. In the Watch window, you can enter variable names and see their values displayed as you step through the code.

5. Halt the debugger when you're done.

Select Debug Stop Debugging from the menu, or simply let the application reach the end of its run and unload it from the AutoCAD editor.