Creating a BREW® Application from Scratch in Visual Studio 2005

Introduction

As a part of the suite of tools available to BREW application developers, the **BREW Add-Ins 3.0 for Microsoft Visual Studio** provide an easy way to create a simple Visual Studio solution/project files and ARM makefiles to support BREW development. Due to changes in the Visual Studio plug-in architecture in Visual Studio 2005, the existing BREW Add-Ins will need to be updated. To assist developers in the interim, this guide demonstrates the process of manually creating and configuring the Visual Studio 2005 solution and generating makefiles for the ARM compiler.

After completing the steps outlined in this document, you will have created a Visual Studio 2005 solution containing a simple BREW application, and you will have modified a template ARM/GCC makefile to compile this application for a BREW device.

Supplied Materials

The following items are provided with this package:

1. Creating a BREW Application from Scratch in Visual Studio 2005 (this document)
2. helloworld2005.c (the main source file for the app)
3. HELLOWORLD2005.BID (defines the class ID for this application)
4. helloworld2005_arm.mak (an arm makefile for the application)
5. helloworld2005_gcc.mak (a gcc makefile for the application)

Prerequisites

This document assumes the following:

1. You have the BREW SDK installed, and your BREWDIR environment variable is set correctly
2. You are familiar with using the BREW Simulator
3. You are familiar with the BREW concepts of MIF and BID files
    a. Refer to the Online Knowledge Base for further information on this topic
4. You have the ARM or GCC compiler installed and correctly configured

Creating the Visual Studio 2005 Solution

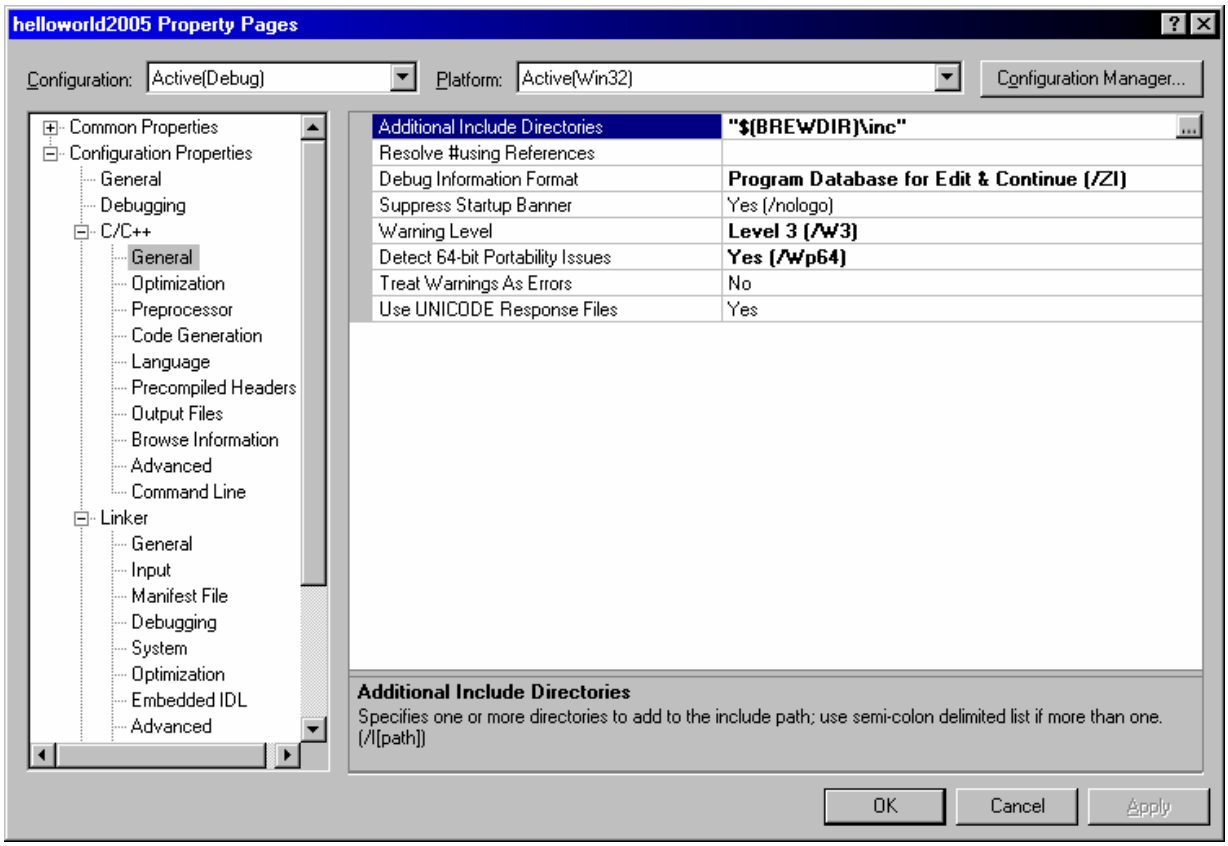
Follow these instructions to create a new Visual Studio 2005 Solution that contains the sample code provided with this document. Creating a solution for your own projects would follow similar steps.

1. Launch Visual Studio 2005 and create a new Win32 project called helloworld2005. Ensure that the “Create directory for solution” checkbox is unchecked.
2. On the Application Settings dialog, specify the Application type as “DLL” and check the “Empty Project” option.
3. Add the following source files to your newly created project:
    - AEEAppGen.c (under your BREW SDK’s /src directory)

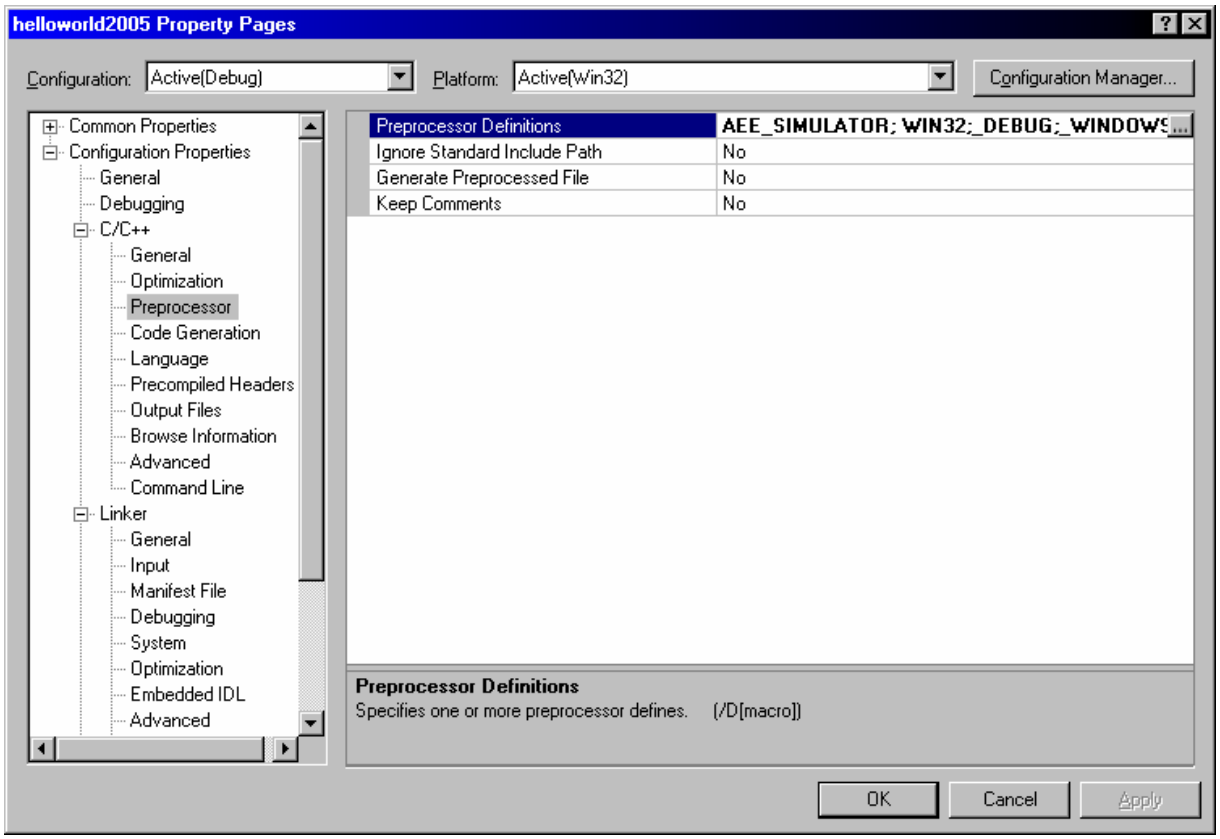
- AEEModGen.c (under your BREW SDK's /src directory)
- helloworld2005.c (included with this document)
- HELLOWORLD2005.BID (included with this document)

4. Ensure your project compiles correctly by opening the Project Property Pages and making the following modifications:

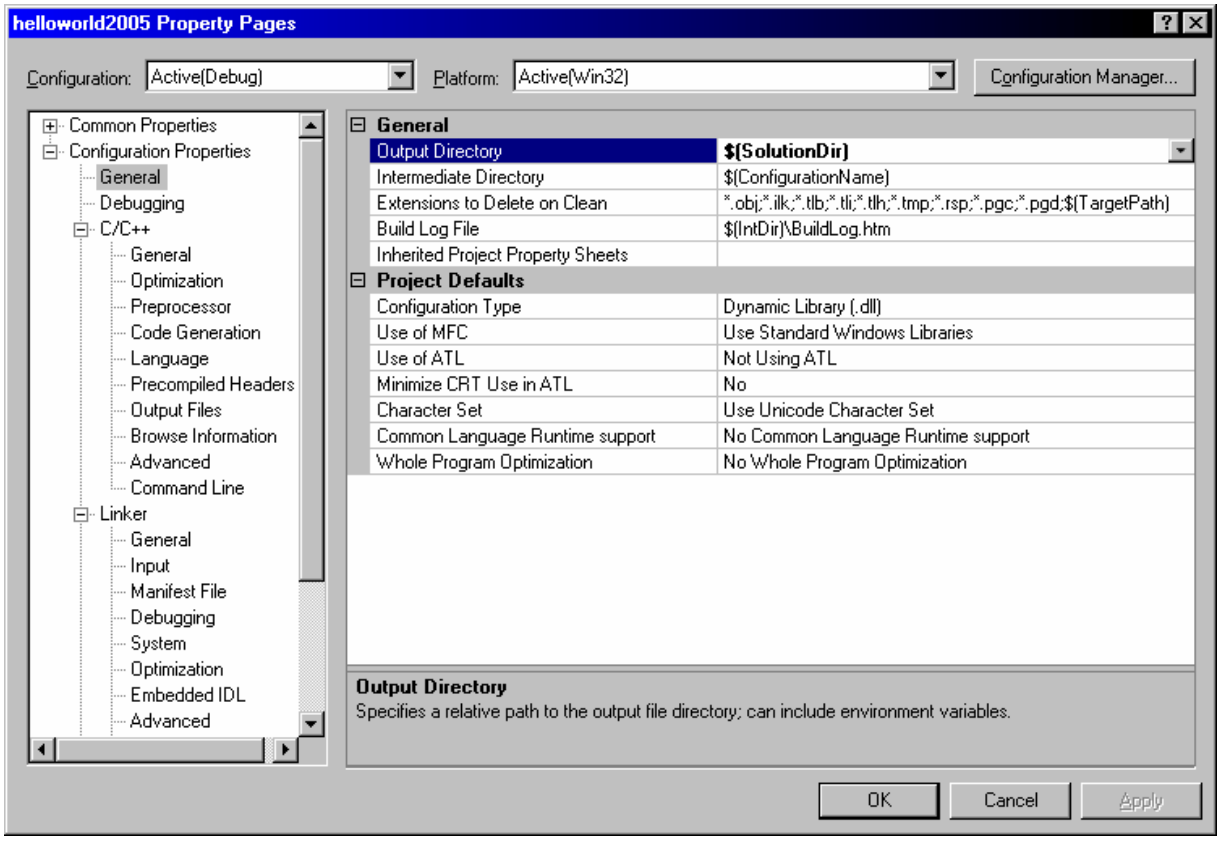
   - Add **`$(BREWDIR)\inc`** to the list of Additional Include Directories. This will allow the compiler to find the BREW API headers.



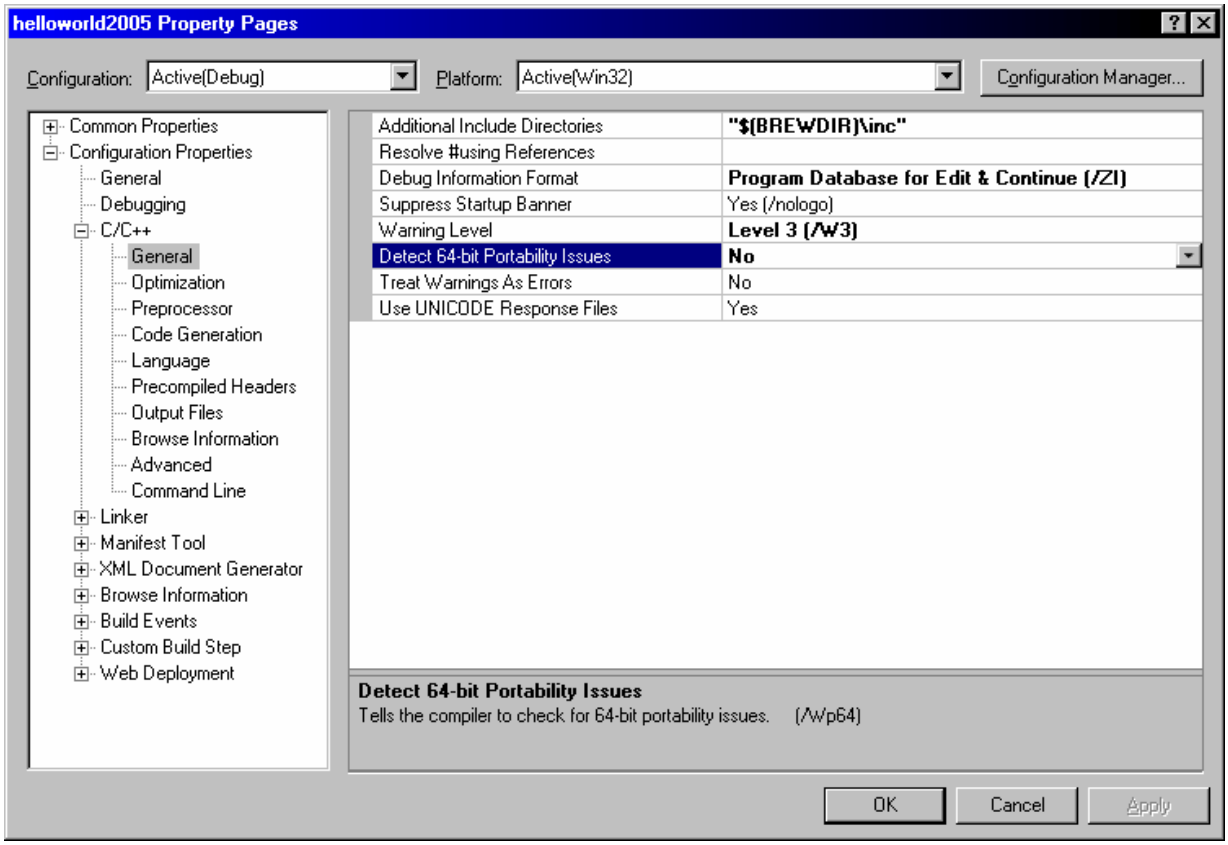
- Add **AEE_SIMULATOR** to the list of Preprocessor Definitions. This ensures that your project is compiled correctly for execution in the BREW Simulator.



- Modify the Output Directory from **`$(SolutionDir)$(ConfigurationName)`** to simply **`$(SolutionDir)`**. This will cause the DLL to be generated in the root of your Solution directory (i.e. "\helloworld2005" rather than "\helloworld2005\Debug")

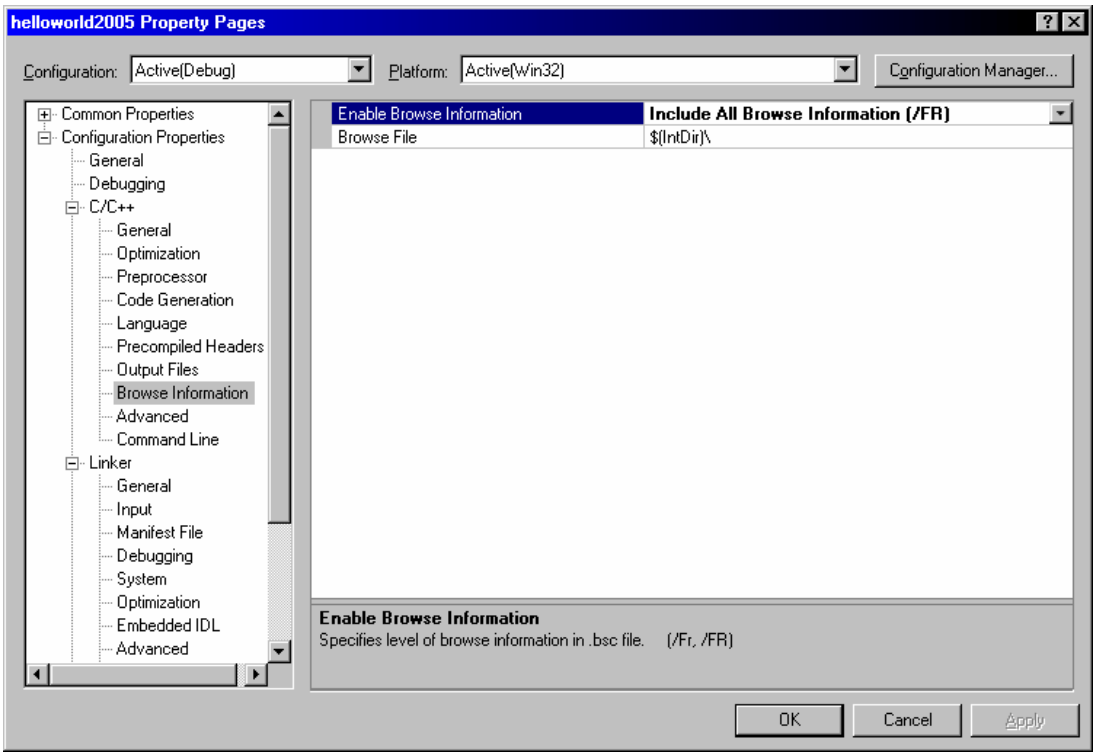


- Turn **`off`** the Detect 64-bit Portability Issues feature. This will disable some extraneous compiler warnings.

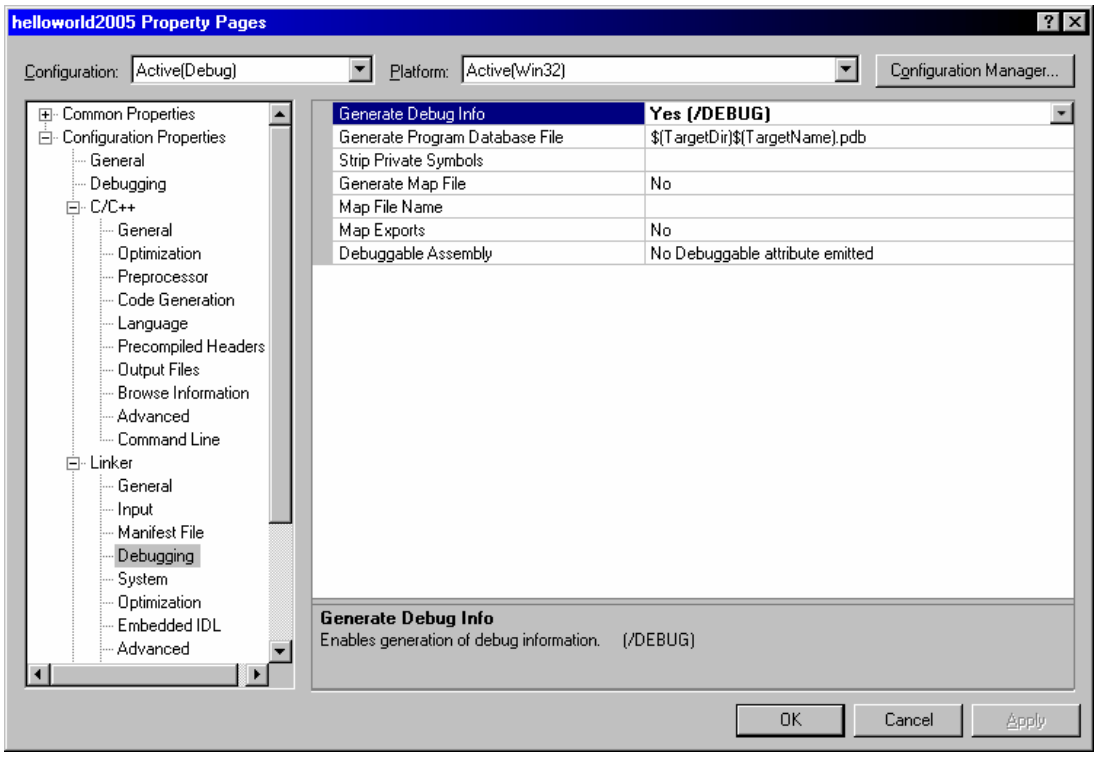


- Enable debugging in your application by making the following modifications:

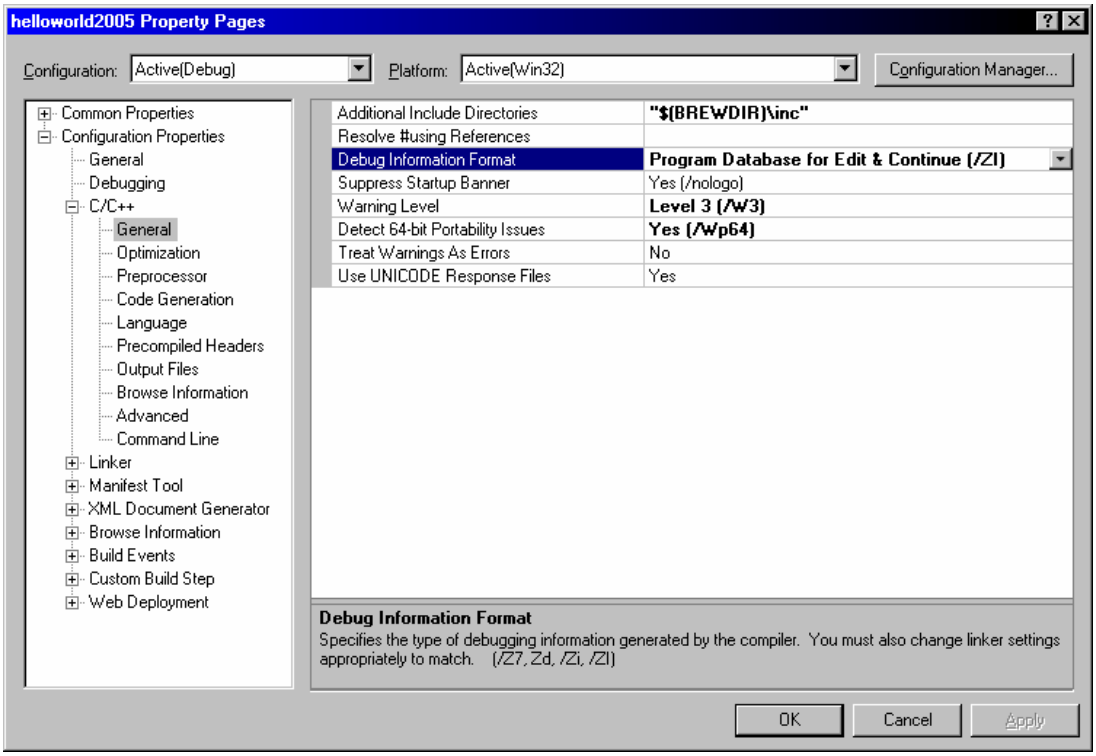
1. Set the Enable Browse Information field to Include **`All Browse Information (/FR)`**.



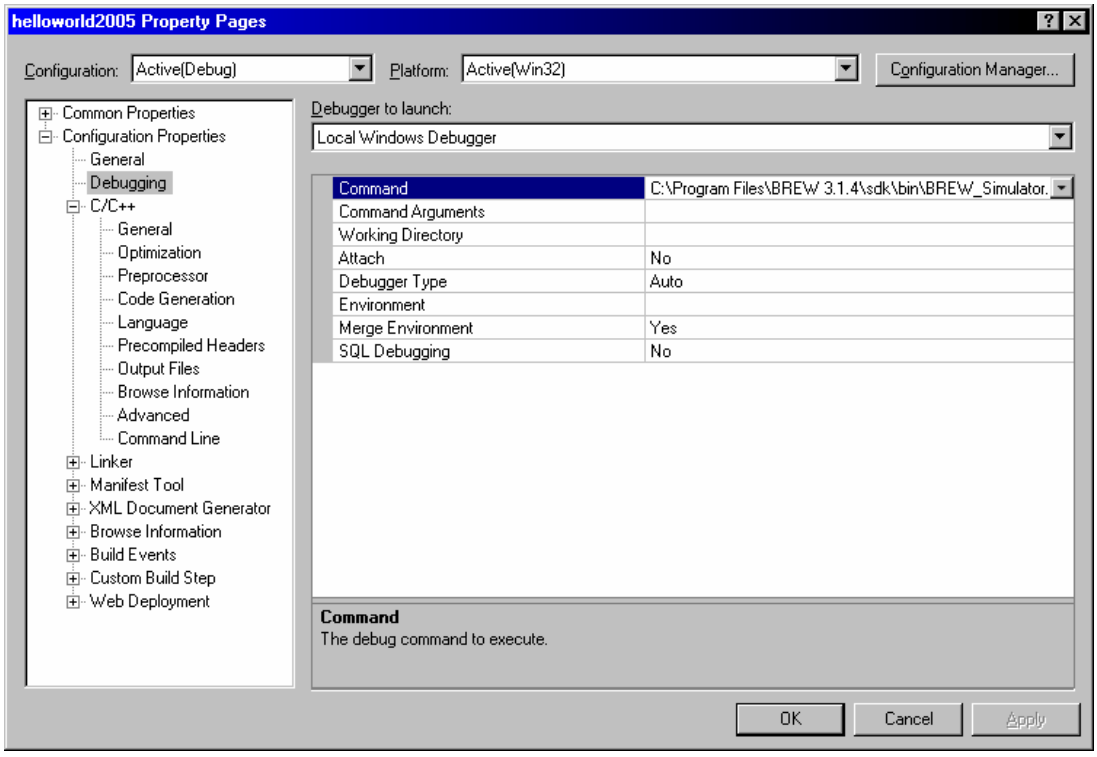
2. Set the Generate Debug Info field to **`Yes (/DEBUG)`**.



3. Set the Debug Information Format to **`Program Database for Edit & Continue (/ZI)`**.



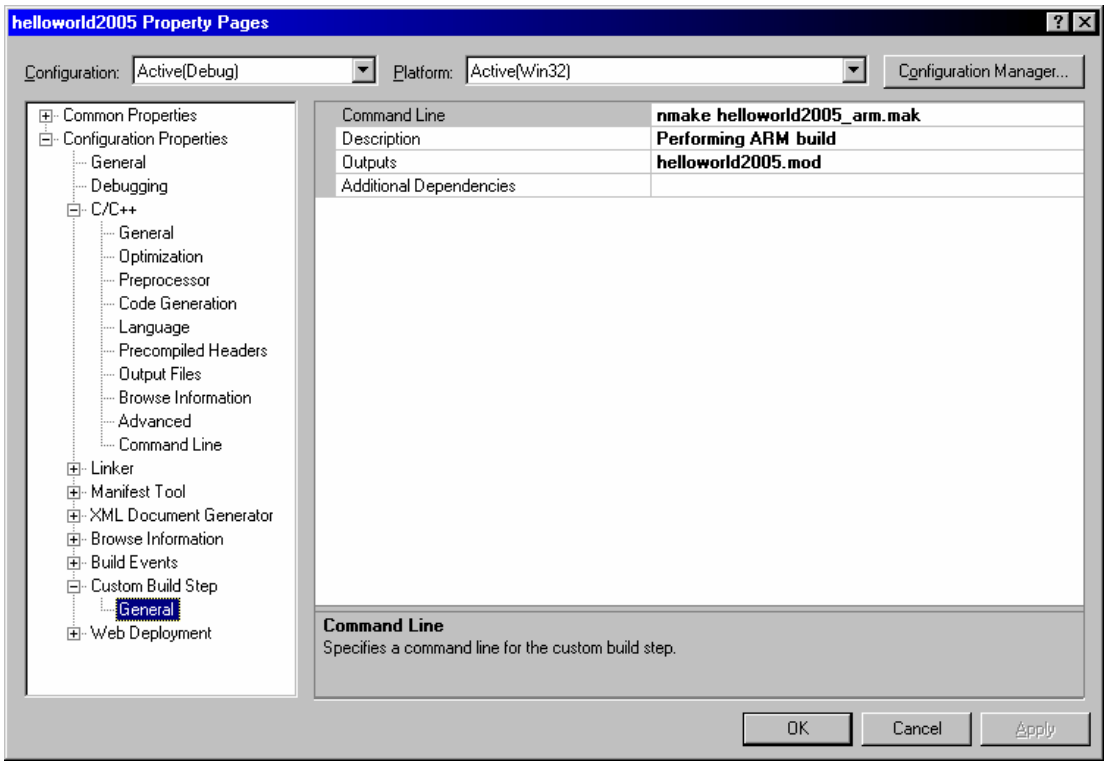
4. Set the Debugging Command to the appropriate BREW Simulator executable.



- *Optional*: You may wish to execute your ARM makefile as a Custom Build Step to automate the process of creating a MOD file. You can do this by setting the following parameters on the Custom Build page (settings are similar for GCC make):

    1. Command Line: **`nmake helloworld2005_arm.mak`**

    2. Outputs: **`helloworld2005.mod`**



5. You should now be ready to compile your application.

Creating the ARM makefile

In order to create the appropriate ARM makefile for your project, you should edit the included template makefiles. This section describes the steps involved in editing the ARMCC makefile. The steps involved to modify the GCC makefile are very similar.

1. Set the correct compilation target:

    - Code:

```
BREW_HOME       =$(BREWDIR)
ARM_HOME        =$(ARMHOME)
TARGET          =helloworld2005
OBJS            =AEEAPP~1.o AEEMOD~1.o helloworld2005.o
APP_INCLUDES    = -I "$(BREWDIR)\inc"
```

2. Add all necessary object files:

    - Code:

```
BREW_HOME       =$(BREWDIR)
```

```
ARM_HOME        =$(ARMHOME)
TARGET          =helloworld2005
OBJS            =AEEAPP~1.o AEEMOD~1.o helloworld2005.o
APP_INCLUDES    = -I "$(BREWDIR)\inc"
```

3. Modify the compilation rules to match your project directory:

   - Code:

```
RULE2 = \PROJECTS\HELLOW~1
{$(RULE2)}.c.o:
        @echo ----------------------------------------------------------------
        @echo OBJECT $(@F)
        $(ARMCC) $(CFLAGS) $(INC) $(OBJ_FILE)  $(RULE2)\$(SRC_FILE)
        @echo ----------------------------------------------------------------
```

4. Generate the dependency list by invoking armcc with the –m option. Append the generated list of dependencies to the template makefile.
   - **C:\apps\ADS12\bin\armcc –m –I "C:\Program Files\BREW 3.1.4\sdk\inc" helloworld2005.c**

Further Information

Please refer to the following links for further information about the ARMCC and GCC compilers:

- ARM Documentation - Software Development Tools
- GCC Literature