

Introduction

The project manager in MPLAB v3.40 has been extended to support multiple files. Previously established projects from MPLAB v3.31 and earlier will be converted automatically by MPLAB v3.40 when they are opened. Converted projects cannot be re-opened from previous versions of MPLAB.

Highlights

In this tutorial you will learn these functions of MPLAB Projects:

- Overview of MPLAB Projects**
- Making a Project with One MPASM Source File**
- Compiling a Single MPASM Source File Without Creating a Project**
- Making a Project with Multiple MPASM Source Files with MPLINK**
- Making a Project with MPLAB-C17**
- Making a Project with Hi-Tech PIC C**

To perform these tasks, you will use the following features of MPLAB:

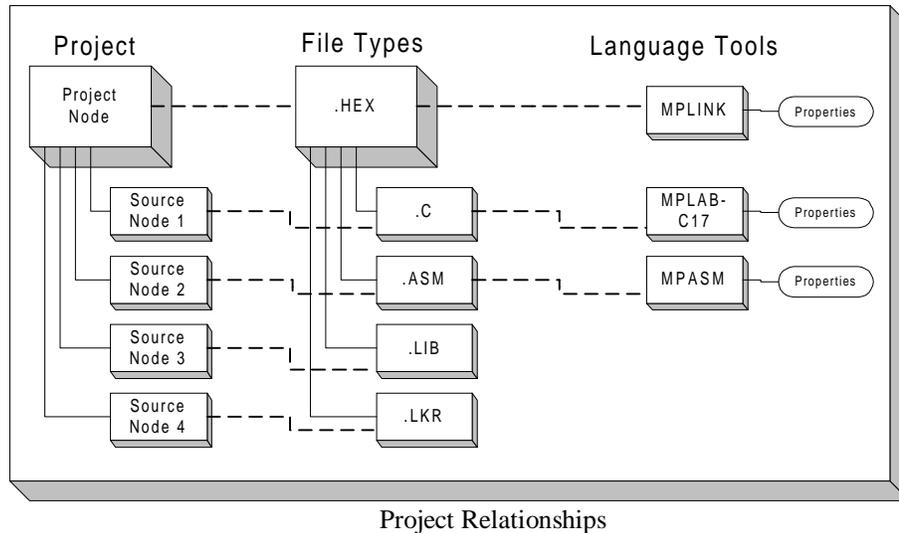
- Install Language Tool**
- New Project**
- Add Nodes to a Project**
- Set Project Node Properties**
- Make/Build Project**
- Project Window**

Glossary

<i>Project</i>	A Project in MPLAB is the group of files with links to the build tools to make an application..
<i>Build Tools</i>	Usually a compiler, assembler, or linker.
<i>Node</i>	A file of some type that is a component of a project.
<i>Project Node</i>	The “top” level of the project. All other files are used to build this.
<i>Source Node</i>	One of the component files that goes into building an application. This can be an assembly source file, a C language source file, a pre-compiled object file, a library, or a linker script.
<i>Make Project</i>	Conditionally compile, assemble, and link all the source nodes to build the application, as required by changes in the source code or project options.
<i>Build Project</i>	Unconditionally compile, assemble, and link all the source nodes to build the application.

Overview of MPLAB Projects

A Project in MPLAB is the group of files needed to build an application along with their associations to various build tools. A project is made up of a project node and one or more source nodes. The source nodes are typically MPASM source files, MPLAB-C17 source files, pre-compiled libraries and object files, and linker scripts. Usually the project is placed in the same directory as the main source files.



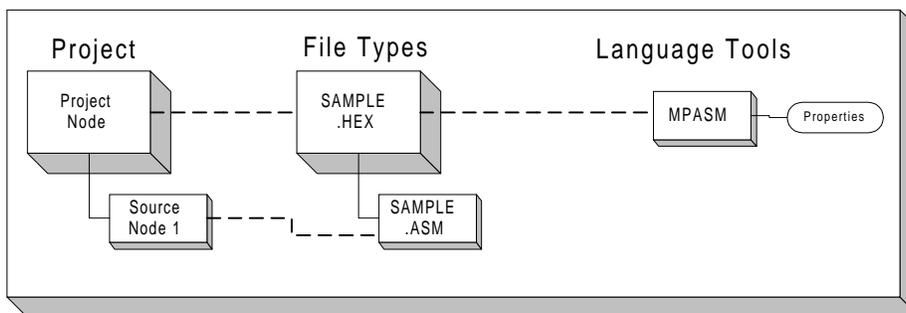
Associated with each node in a project is a language tool. When you create a project, you must assign a language tool unless such an assignment would be redundant. For instance, if you are using MPASM with a single source file (first tutorial below), you won't have to assign a language tool to the source file node, since the project node is already set up for MPASM. Similarly, precompiled libraries, precompiled object files, and linker scripts for MPLAB-C17 or MPASM will not need to be assigned a tool since these will always be built with MPLINK.

There are multiple options for each language tool that are usually set via command line switches when invoked from DOS. These options show up as Node Properties for each editable node. Such switches control the generation of .MAP files, choose the proper .HEX format, and enable/disable warning messages, among other things. They can be set differently for the various source nodes on a project, although, typically, they will be the same. The Node Properties dialog corresponds to the command line switches for the various tools and, when first viewed, this dialog shows the default values. Refer to the MPASM and MPLAB-C17 User's Guides for information on these command line switches.

Built into MPLAB's Project Manager is a MAKE facility. This tool looks at the date/time stamp for the source files that go into an application and figures out which components have changed and need to be re-compiled or assembled when the project is re-linked.

Making a Project with One MPASM Source File

To make a project that has only one MPASM source file, or that uses the previous method of projects (MPLAB v3.31 or earlier), wherein a single source file would `#include` other files, follow these steps.



Project Relationships For One MPASM Source File Tutorial

Set Development Mode

Select the proper development mode for the application. For this tutorial set Options>Development Mode to MPLAB-SIM simulator and select the 16F84 PICmicro.

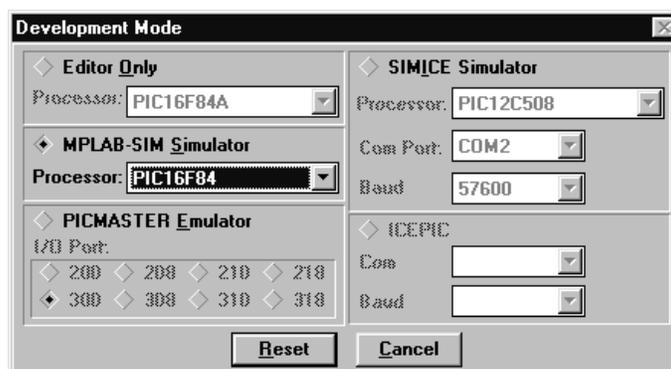


Figure 1

New Project

Select **Project>New Project**, select a directory for the new project, then type in its name. Use the MPLAB installation directory and name it `SAMPLE.PJT` for this tutorial.

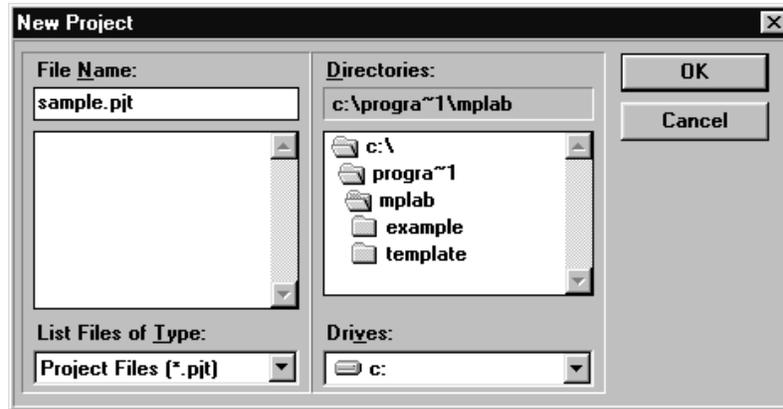


Figure 2

Project Dialog

After clicking on OK, you will see the Edit Project Dialog:

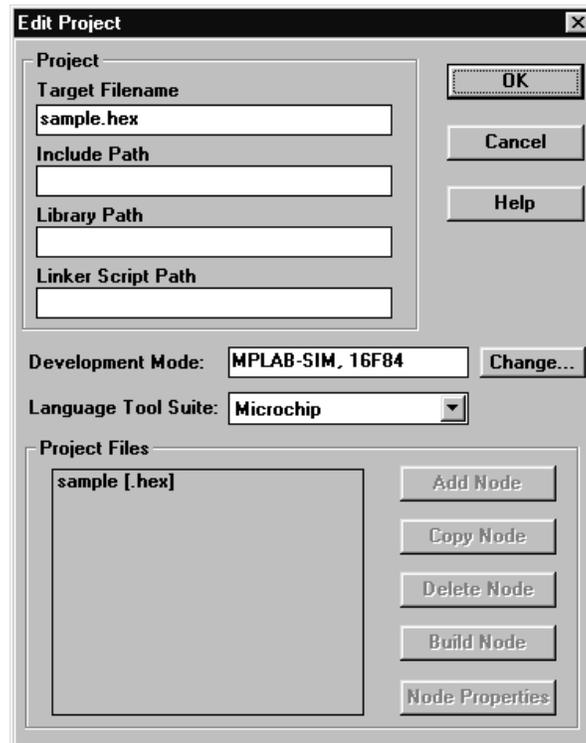


Figure 3

Set Node Properties

Click on the file name, SAMPLE.HEX, in the “Project Files” window, then select the “Node Properties” button.

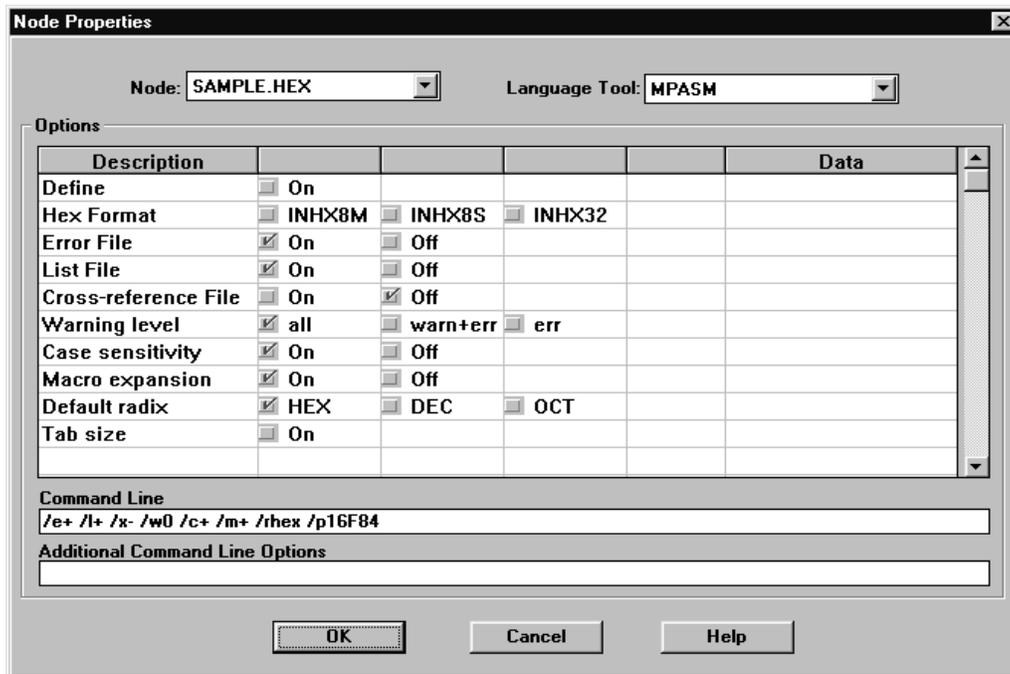


Figure 4

NOTE: The “Node Properties” dialog shows the command line switches for the tool, in this case MPASM. When you first open this dialog, the checked boxes represent the default values for the tool. For this tutorial, these do not need to be changed. Refer to the MPASM with MPLINK and MPLIB User’s Guide for more information on these command line switches.

Select OK to return to the Edit Project Dialog box.

Add Node

Select "Add Node" from the Edit Project Dialog. Use SAMPLE.ASM for this tutorial. This is the browse window that pops up from choosing “Add Node”.

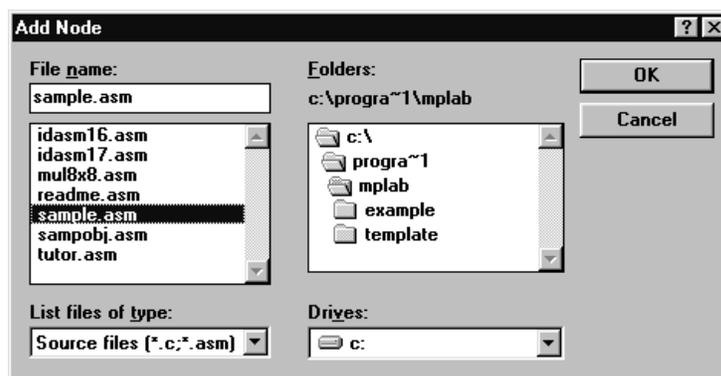


Figure 5

MPASM always makes a .HEX file with the same name as the source .ASM file. The Project Manager will create a SAMPLE.HEX file when the project is built.

Troubleshooting

If this did not work check these items:

Select Project>Install Language Tool... and check that MPASM is pointed to the MPASM.EXE in the MPLAB installation directory. Alternatively, MPASM can point to MPASMWIN.EXE, but the “Windowed” option should be selected.

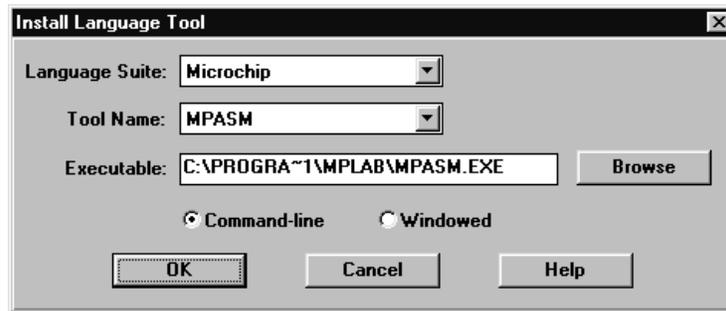


Figure 8

If you get a message from DOS saying that you have run out of environment space, use Windows Explorer to select the MPASM.EXE file in the MPLAB installation directory, and click on the right mouse button to bring up the Properties dialog:

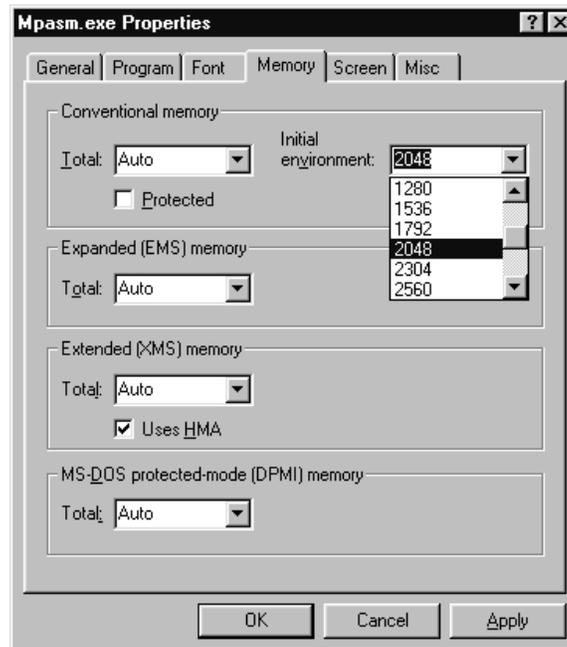


Figure 9

Increase the size of the Initial Environment. Usually a setting of 2048 will suffice, but if you have a lot of applications that set variables and add to your path statement in your AUTOEXEC.BAT file, you may need to make it larger.

Project Window

Open the Window>Project window to see that the target name is set properly to match the Node source name. They will have different file extensions, .ASM and .HEX, but both are named SAMPLE for this tutorial.

The Project window should look like this:

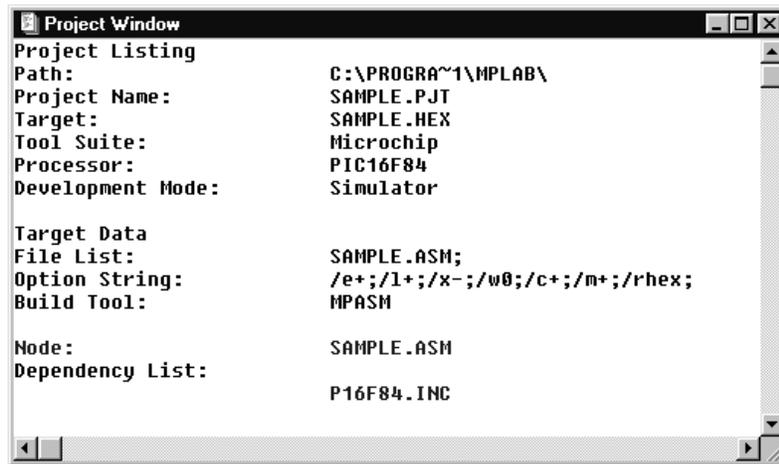


Figure 10

Summary of Setting Up MPASM Single File Projects

Here is a quick list of the steps to set up a new project as described above:

- Create new project with Project>NewProject
- Set project Node Properties to MPASM and select the desired build options
- Add Source file node

Compiling a Single MPASM Source File Without Creating a Project

It is possible to compile a single file without opening up a project. The disadvantage of this method is that although no initial project setup is needed, it requires that you specify options every time you compile the file. This example will use the same assembly-language file used in the last example.

You must first close any open projects. To do this, select Project>Close Project.

Set Development Mode

Select the proper development mode for the application. For this tutorial, set Options>Development Mode to MPLAB-SIM simulator and select the 16F84 PICmicro.

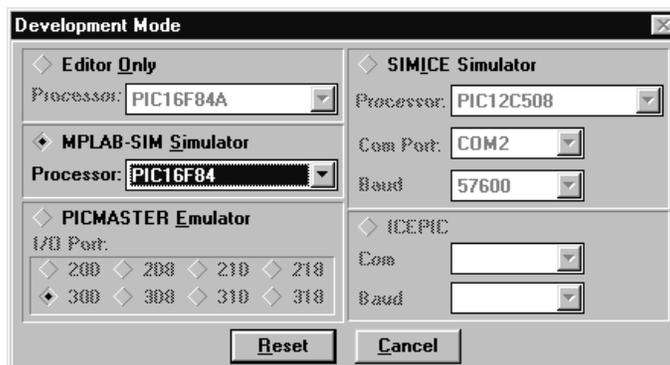


Figure 11

Open Source File

Open the source file that you wish to assemble. For this tutorial, use `SAMPLE.ASM` from the MPLAB installation directory.

```
c:\progra~1\mplab\sample.asm
;*****
;                SAMPLE.ASM
;                8x8 Software Multiplier for 16Cxxx Family
;*****
;
; The 16 bit result is stored in 2 bytes
;
; Before calling the subroutine " mpy ", the multiplier should
; be loaded in location " mulplr ", and the multiplicand in
; " mulcnd ". The 16 bit result is stored in locations
; H_byte & L_byte.
;*****
;
; LIST    p=16F84 ; PIC16F844 is the target processor
;
; #include "P16F84.INC" ; Include header file
;
; cblock 0x10 ; Temporary storage
;     mulcnd ; 8 bit multiplicand
;     mulplr ; 8 bit multiplier, this register will be set to zero after multiply
;     H_byte ; High byte of the 16 bit result
;     L_byte ; Low byte of the 16 bit result
;     count ; loop counter
; endc
;
; org    0
```

Figure 12

Compile Source File

Select **Project>Build Node** from the menu to compile SAMPLE.ASM using MPASM. MPLAB brings up an Invoke Build Tool Dialog that looks like this:

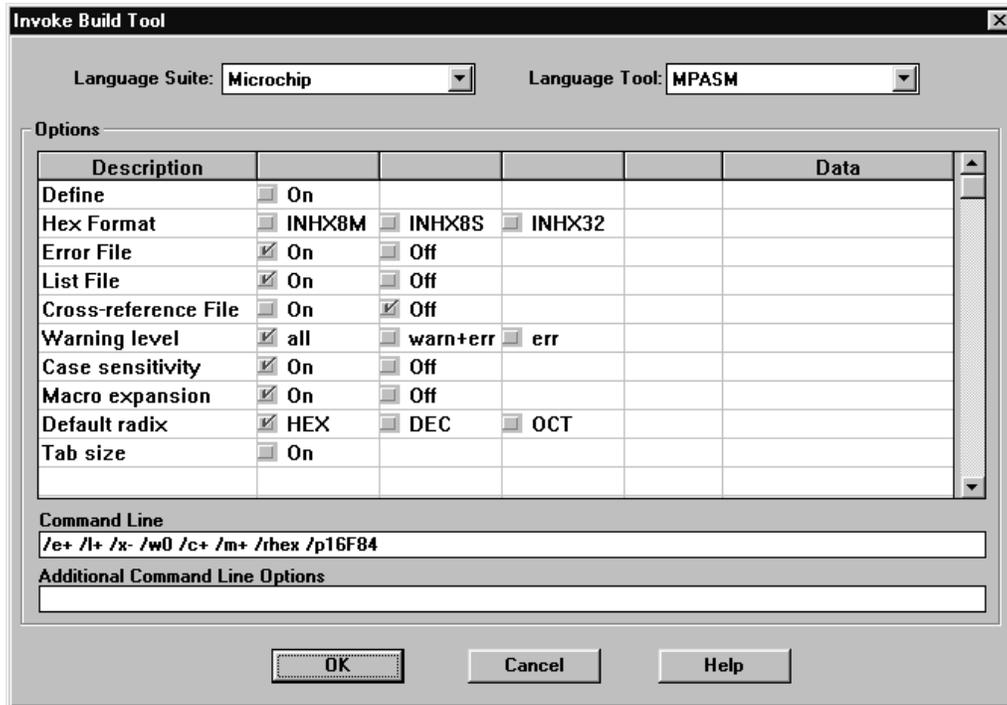


Figure 13

Verify that MPASM is selected, and set the tool options to match those shown above. Press “OK” in the Invoke Build Tool Dialog to start the build process. A Build Results window is generated that shows the command line sent to the assembler and the build output. It should look like this:

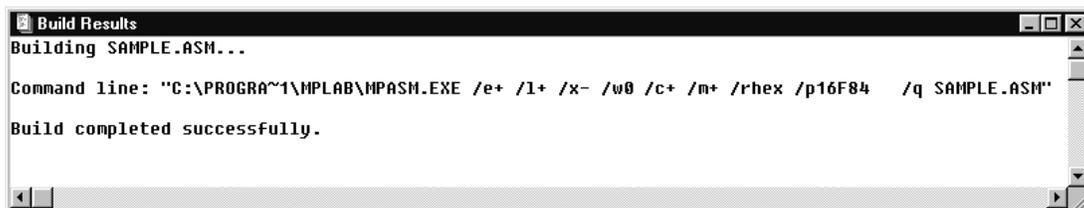
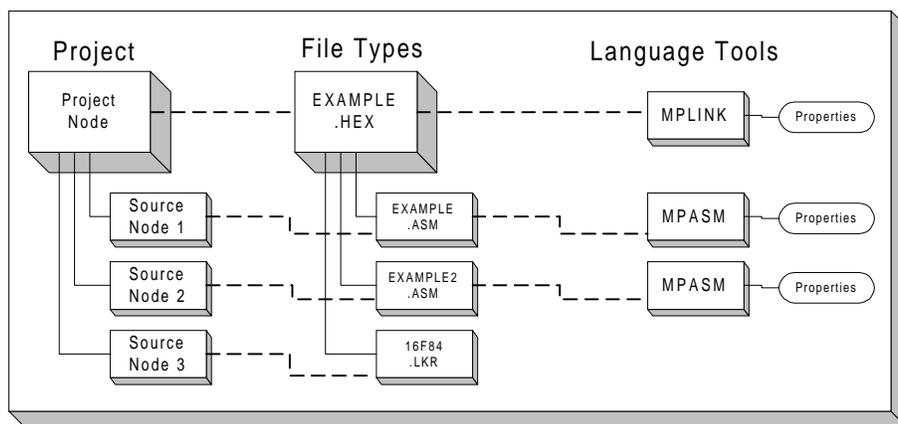


Figure 14

Making a Project with Multiple MPASM Source Files with MPLINK

To use MPLINK to link two or more MPASM object files, follow these steps. If you followed through the previous section, select **Project>Close Project**.



Project Relationships For Multiple MPASM Source File Tutorial

Set Development Mode

Set **Options>Development Mode** to MPLAB-SIM simulator and select the 16F84 PICmicro for this example.

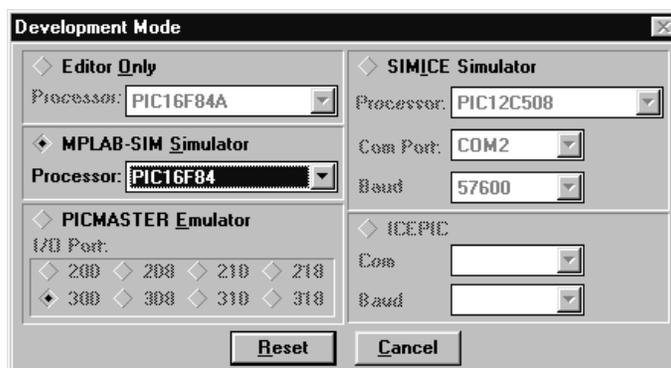


Figure 17

New Project

Select **Project>New Project**, browse to select a directory for a new project, then type in its name. Use the `\MPLAB\EXAMPLE` directory for this tutorial and name it `EXAMPLE.PJT`.

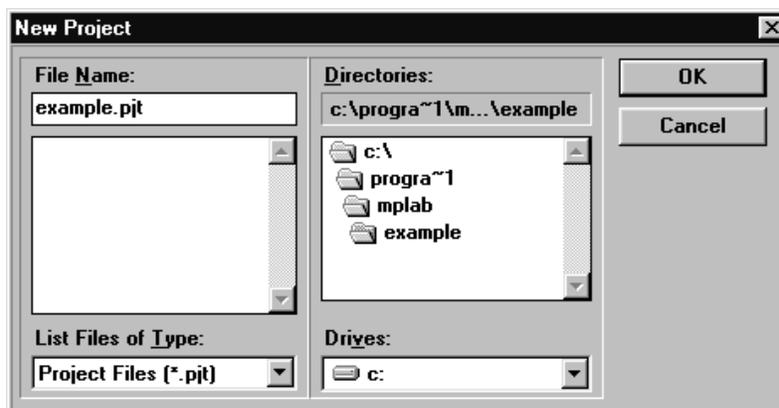


Figure 18

Set Project Node Properties

Select the name of the project in the "Project Files" dialog of the Edit Project Dialog and press "Node Properties" to bring up this dialog. Set the language tool to "MPLINK."

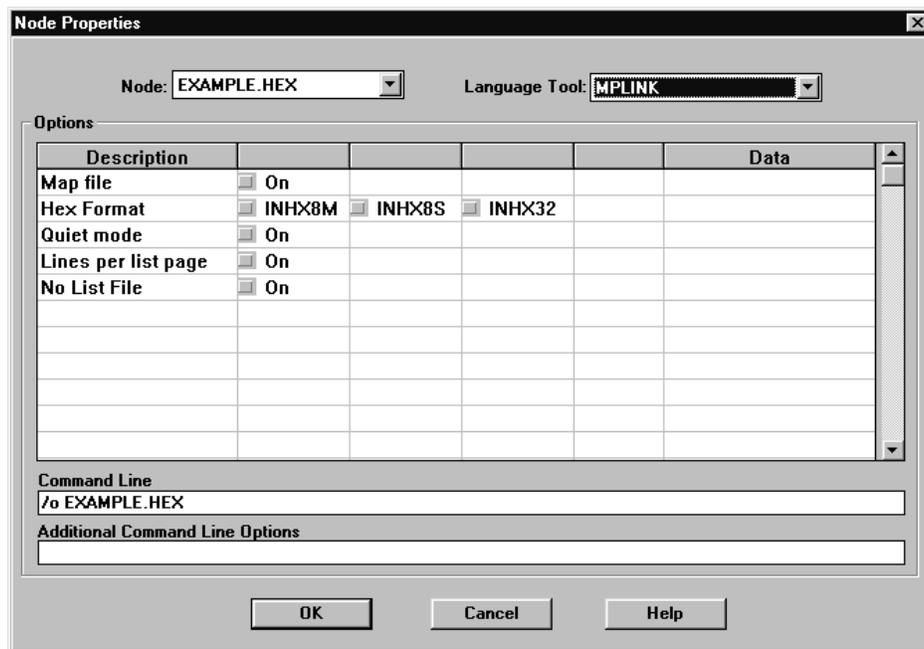


Figure 19

NOTE: The Node Properties dialog shows the command line switches for the tool, in this case MPLINK. When you first open this dialog, the checked boxes represent the default values for the tool. For this tutorial, these do not need to be changed. Refer to the MPASM with MPLINK and MPLIB User's Guide for more information on these command line switches.

Select OK to return to the Edit Project Dialog box.

Add First Source File Node

Select "Add Node" from the Edit Project Dialog. Use EXAMPLE.ASM in the \MPLAB\EXAMPLE directory for this tutorial.

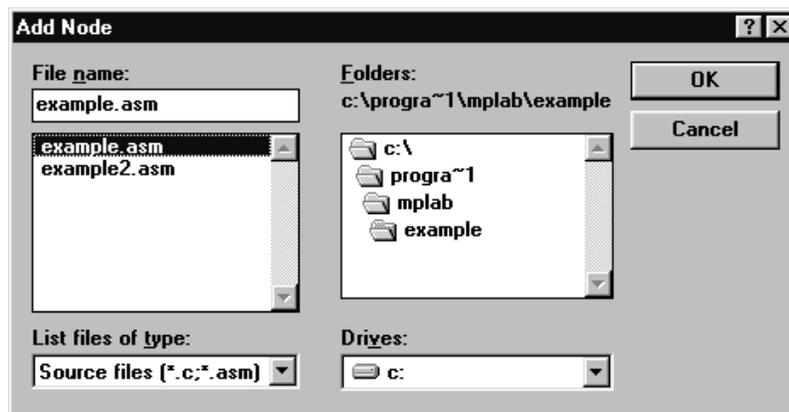


Figure 20

NOTE: You may select more than one file at a time from this dialog using the standard Windows methods of selecting one file and then selecting another while holding down the control key (selects only those two files) or the shift key (selects both files and all those in between).

Select example.asm from the list of project files in the Edit Project dialog, and press "Node Properties."

Verify that the language tool is set to MPASM.

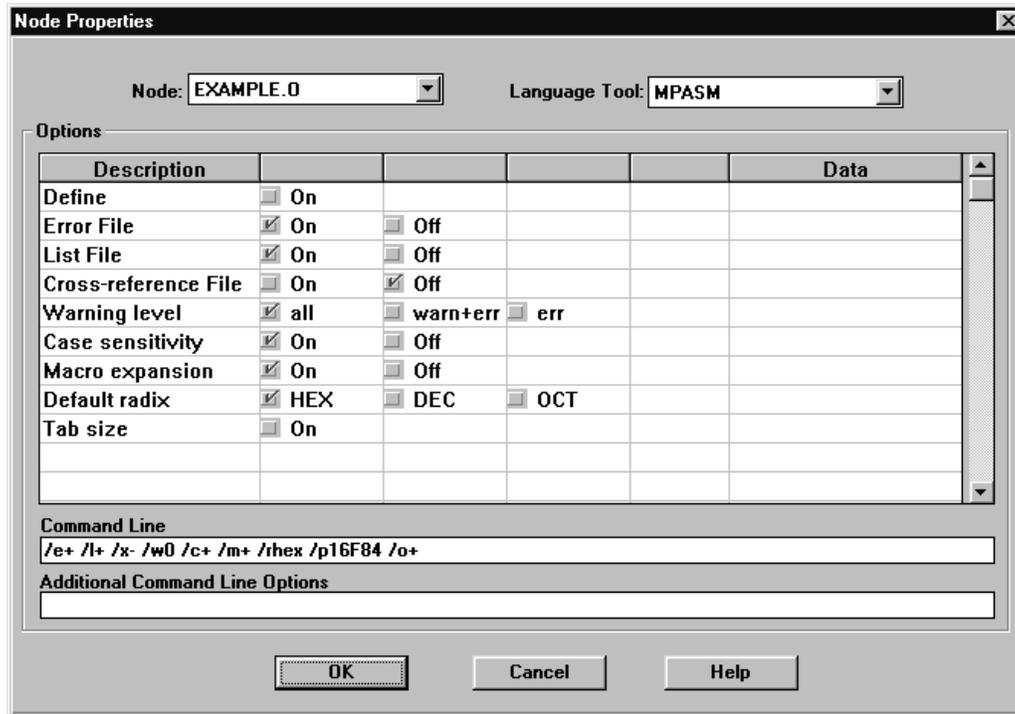


Figure 21

NOTE: The Node Properties dialog shows the command line switches for the tool, in this case MPASM. Refer to the MPASM with MPLINK and MPLIB User's Guide for more information on these command line switches.

Select OK to return to the Edit Project Dialog box.

Adding Additional Source Files

Follow the previous two steps to add the rest of the source files to the project. For this tutorial select EXAMPLE2.ASM from the \MPLAB\EXAMPLE directory. You can also use “Copy Node” to enter subsequent files with the same Node Options as the first source file. Make sure the Node Options are set properly on each file.

NOTE: To use “Copy Node” select one of the source node files listed in the Project Files box. Then select the “Copy Node” feature. In the “Add Node” dialog box, select one or more source files. Once the files are selected choose the OK button. This step will set up the node properties for this selected files the same as the referenced node used. This is especially useful for adding multiple source files with identical node properties.

Select Linker Script

Select a linker script using the “Add Node” button and the method described above. A linker script is a file that MPLINK uses to define the memory architecture of each PICmicro. Standard linker scripts come with MPLINK and are in the MPLAB installation directory. For this tutorial select 16F84.LKR from the \MPLAB\EXAMPLE directory. Node options can not be set for a linker script.

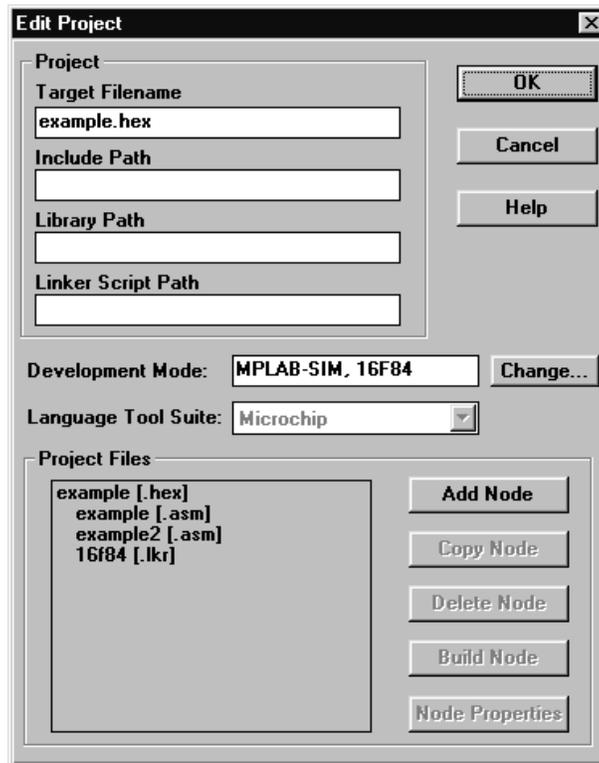


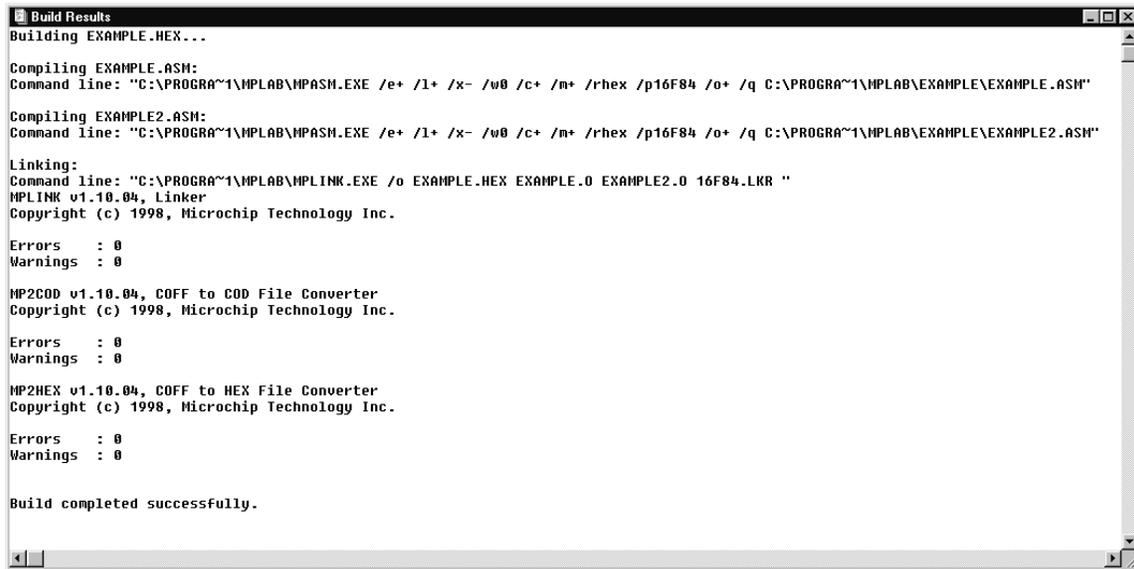
Figure 22

Press "OK" on the Edit Project Dialog.

NOTE In this simple example, no entries were made in the three “Path” boxes. As your application becomes more complex, you may need to enter the directories of your Include Files, Library Files and Linker Script Files in the appropriate box if they are not in the same directory as the project.

Make Project

Select **Project>Make Project** from the menu to compile the application using MPASM and MPLINK. A Build Results window is created that shows the command lines sent to each tool. It should look like this:



```
Build Results
Building EXAMPLE.HEX...

Compiling EXAMPLE.ASH:
Command line: "C:\PROGRAM~1\MPLAB\MPASM.EXE /e+ /l+ /x- /w0 /c+ /m+ /rhex /p16F84 /o+ /q C:\PROGRAM~1\MPLAB\EXAMPLE\EXAMPLE.ASH"

Compiling EXAMPLE2.ASH:
Command line: "C:\PROGRAM~1\MPLAB\MPASM.EXE /e+ /l+ /x- /w0 /c+ /m+ /rhex /p16F84 /o+ /q C:\PROGRAM~1\MPLAB\EXAMPLE\EXAMPLE2.ASH"

Linking:
Command line: "C:\PROGRAM~1\MPLAB\MPLINK.EXE /o EXAMPLE.HEX EXAMPLE.O EXAMPLE2.O 16F84.LKR "
MPLINK v1.10.04, Linker
Copyright (c) 1998, Microchip Technology Inc.

Errors : 0
Warnings : 0

MP2C0D v1.10.04, COFF to COD File Converter
Copyright (c) 1998, Microchip Technology Inc.

Errors : 0
Warnings : 0

MP2HEX v1.10.04, COFF to HEX File Converter
Copyright (c) 1998, Microchip Technology Inc.

Errors : 0
Warnings : 0

Build completed successfully.
```

Figure 23

Troubleshooting

If this did not work, check these items:

If MPLAB reports the message, “Time-out”, select OK to continue. Depending on the speed of your PC and the size of your project, you may wish to configure the length of time MPLAB will wait before reporting a timeout message. This value is set by selecting **Options>Environment Setup...** and adjusting the timeout settings in that dialog box.

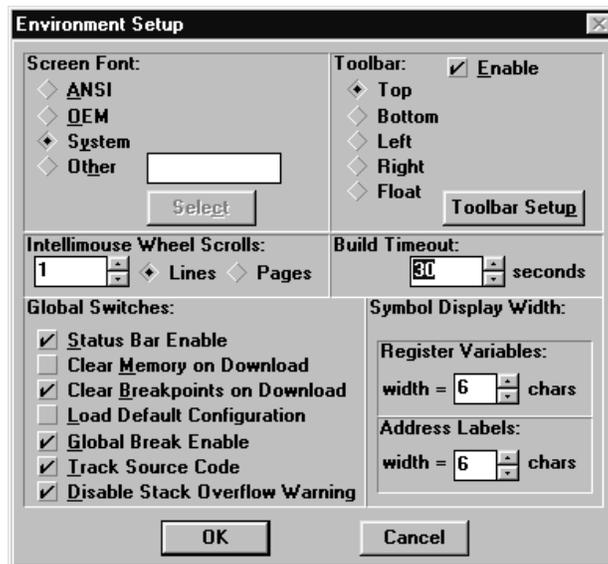


Figure 24

Select **Project>Install Language Tool...** and check that MPASM and MPLINK are pointed to the MPASM.EXE and MPLINK.EXE executables in the MPLAB installation directory.

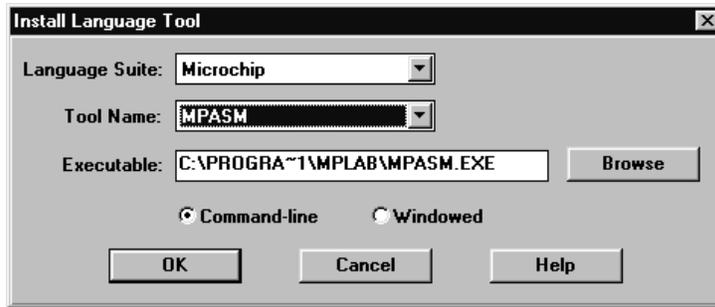


Figure 25

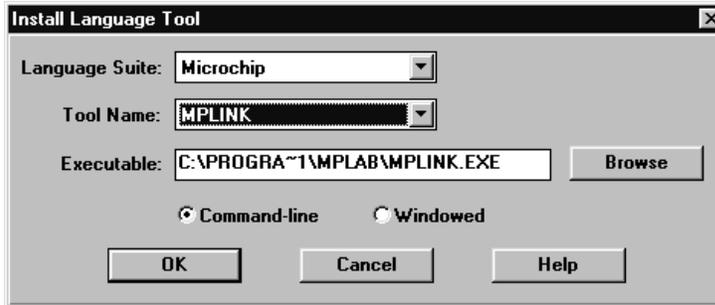


Figure 26

If you get a message from DOS saying that you have run out of environment space, use Windows Explorer to select the MPASM.EXE file in the MPLAB installation directory, and click on the right mouse button to bring up the Properties dialog:

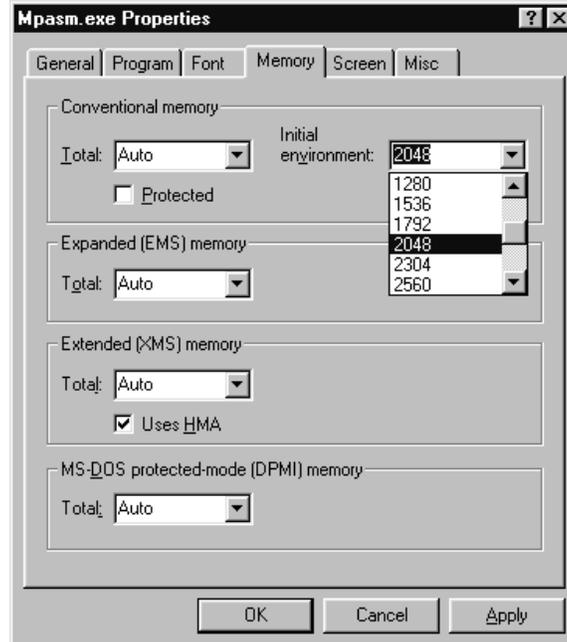


Figure 27

Increase the size of the Initial Environment. Usually a setting of 2048 will suffice, but if you have a lot of applications that set variables and add to your path statement in your AUTOEXEC.BAT file, you may need to make it larger.

Project Window

Open the Window>Project window. It should look like this:

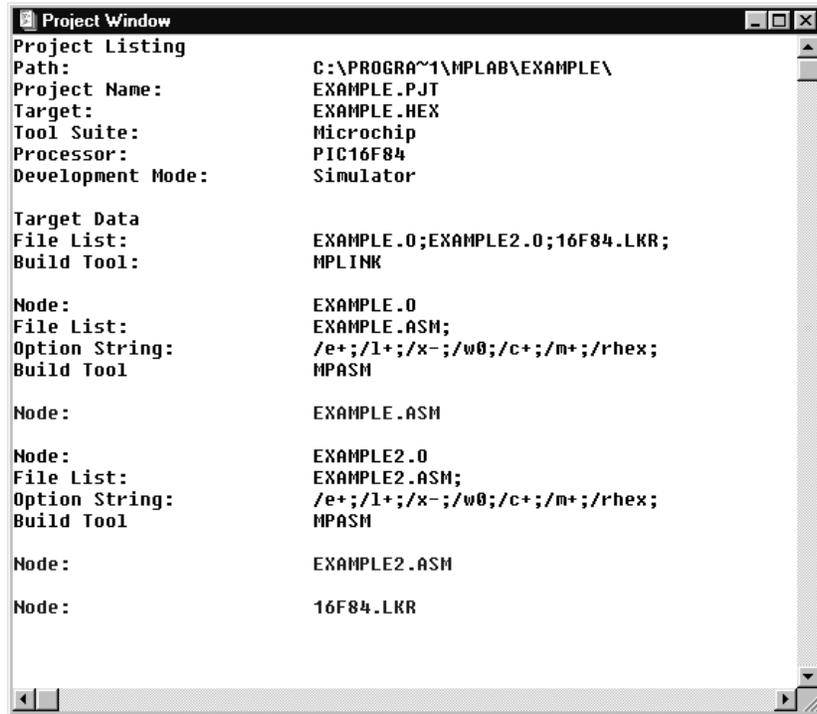


Figure 28

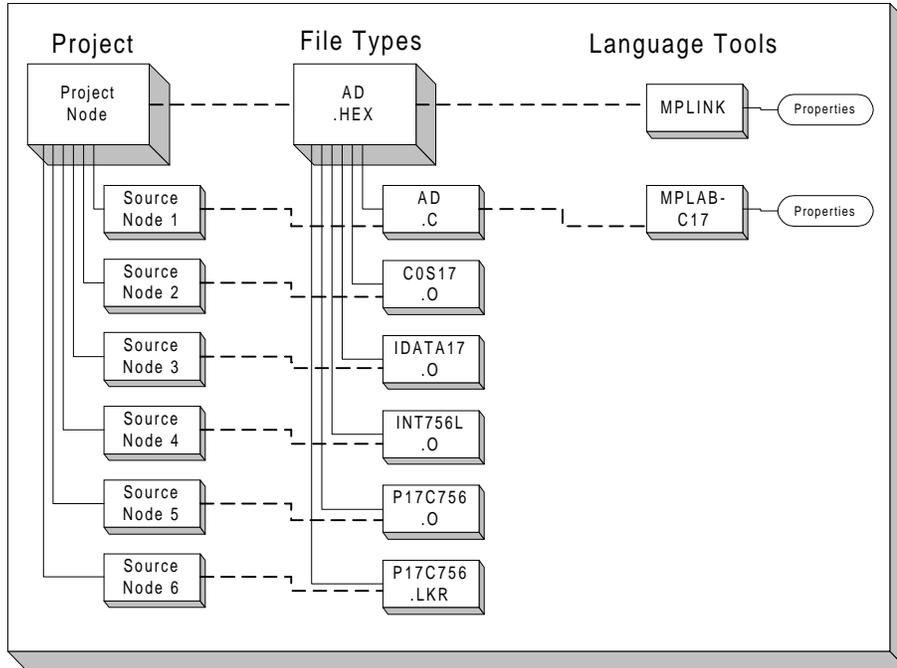
Summary of MPASM Multiple File Projects with MPLINK

Here is a quick list of the steps to set up a new project as described above:

- Create new project with Project>NewProject.
- Set project Node Properties to MPLINK.
- Add Source file nodes, and set node properties as needed.
- Add Linker Script file node.

Making a Project with MPLAB-C17

This tutorial will show you how to use MPLAB-C17 with projects in MPLAB to build applications. If you have followed along the previous tutorials, select **Project>Close Project**.



Project Relationships for MPLAB-C17 Tutorial

Set Development Mode

Set **Options>Development Mode** to MPLAB-SIM simulator and select the 17C756 PICmicro for this example.

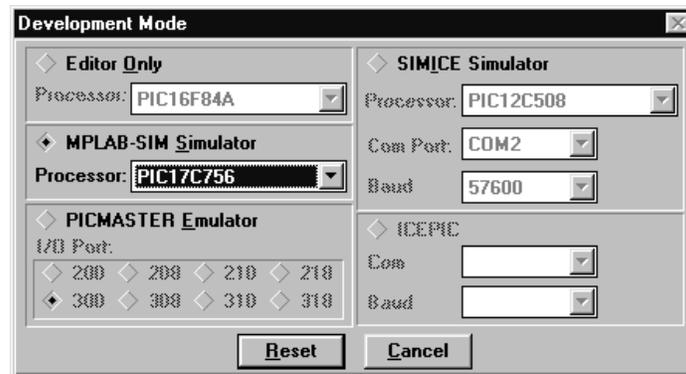


Figure 29

Install MPLAB-17 Language Tool

Make certain that MPLAB-C17 is installed correctly in MPLAB. The Project>Install Language Tool dialog should look like this (your executable path may be different):

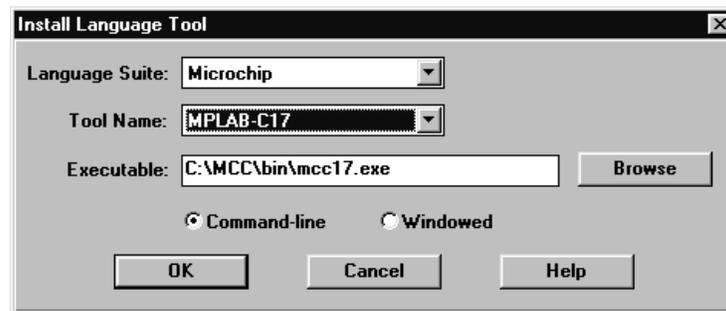


Figure 30

If the executable is not shown in the window, use the Browse button to point to MCC17 .EXE on your system.

New Project

Select Project>New Project and select a directory for a new project, then type in its name. Name it AD . PJT in the \MCC\EXAMPLES\AD directory.

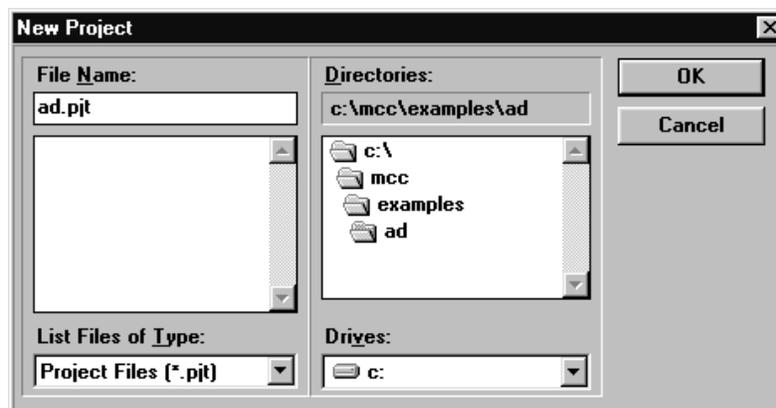
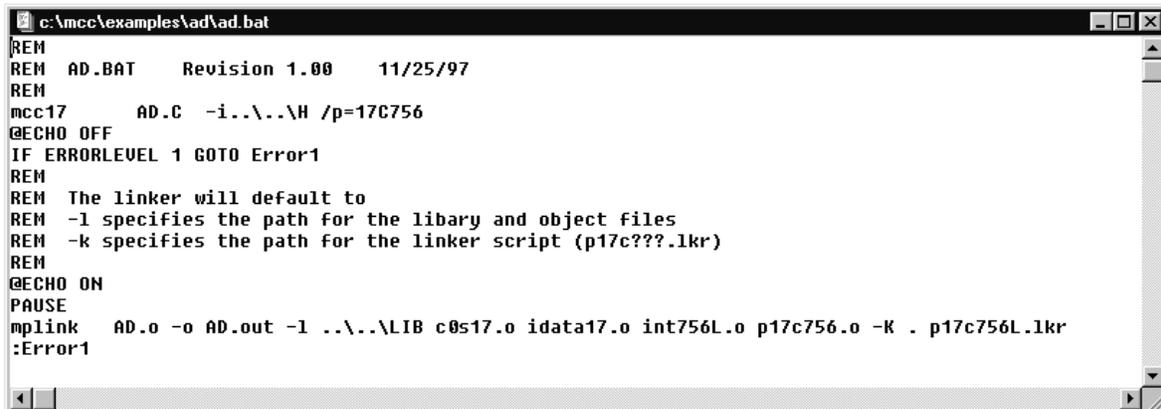


Figure 31

NOTE: The Node Properties dialog shows the command line switches for the tool, in this case MPLINK. When you first open this dialog, the checked boxes represent the default values for the tool. For this tutorial, these do not need to be changed. Refer to the MPASM with MPLINK and MPLIB User's Guide for more information on these command line switches.

Add First Source File

To determine which nodes to set up from this tutorial, look at AD.BAT. This is the batch file that can be used to compile this example from a DOS command line and is in the \MCC\EXAMPLES\AD directory. Use this data to add all required nodes. Here is a listing of the batch file:



```
c:\mcc\examples\ad\ad.bat
REM
REM AD.BAT  Revision 1.00  11/25/97
REM
mcc17      AD.C  -i..\..\H /p=17C756
@ECHO OFF
IF ERRORLEVEL 1 GOTO Error1
REM
REM The linker will default to
REM -l specifies the path for the library and object files
REM -k specifies the path for the linker script (p17c???.lkr)
REM
@ECHO ON
PAUSE
mplink AD.o -o AD.out -l ....\LIB c0s17.o idata17.o int756L.o p17c756.o -K . p17c756L.lkr
:Error1
```

Figure 34

The nodes required are AD.C - the main source file which must be compiled, the linker script P17C756L.LKR, and the following object files:

- C0S17.O - Start Up Code
- IDATA17.O - Code to Initialize Data
- INT756L.O - Interrupt Service Routines
- P17C756.O - PIC17C756 Register Definitions

You can return to setting up the project from the Project>Edit Project menu selection.

Select "Add Node" from the Edit Project Dialog. Add the source file, AD.C from the \MCC\EXAMPLES\AD directory.

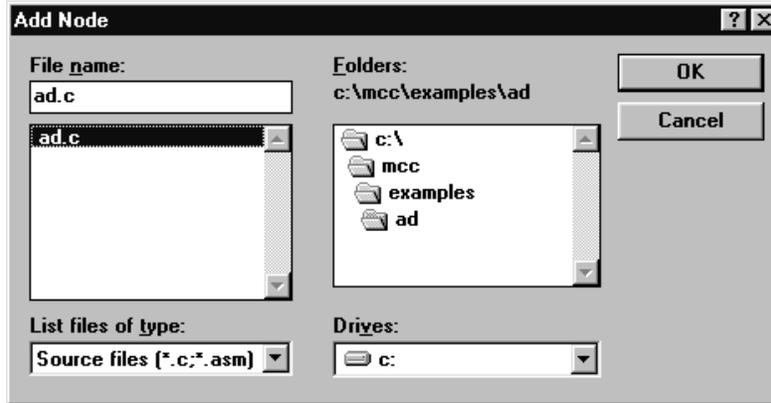


Figure 35

Select the file name in the Edit Project dialog and press "Node Properties." Verify that the language tool is set to MPLAB-C17.

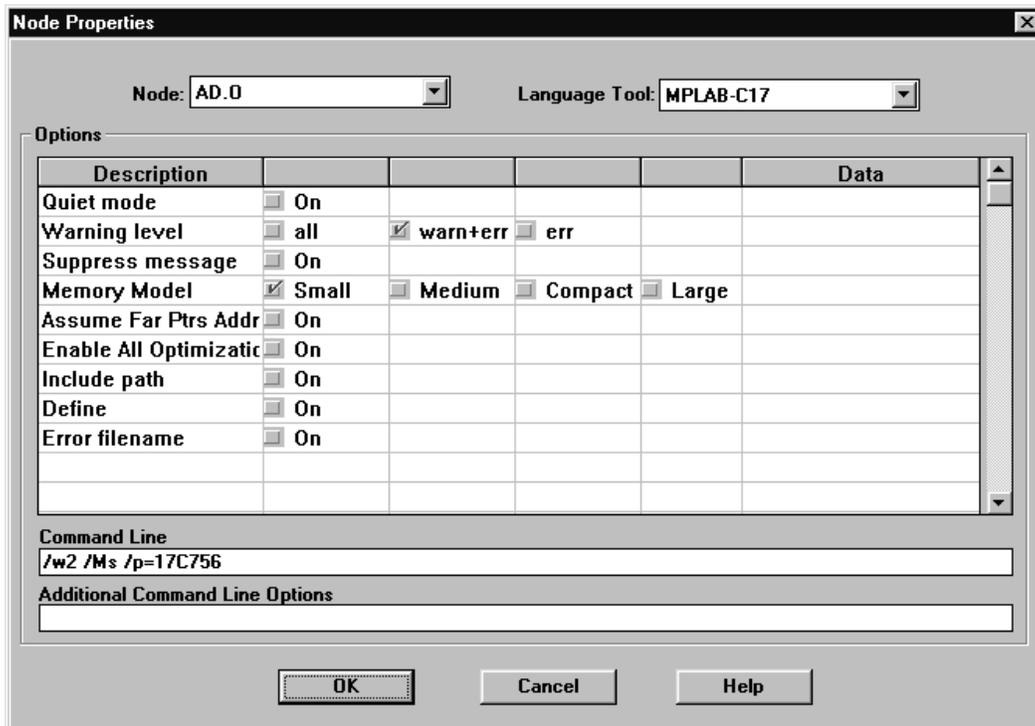


Figure 36

The "Object filename" is set to "AD.O" automatically. Nothing else needs to be changed in this dialog.

NOTE: The Node Properties dialog shows the command line switches for the tool, in this case MPLAB-C17. The checked boxes represent the default values for the tool. For this tutorial, none of the setting need to be changed from their default values. Refer to the MPLAB-C17 User's Guide for more information on these command line switches.

Adding Pre-Compiled Object Files

Use the “Add Node” button from the Edit Project dialog to add the precompiled object files from the MPLAB-C17 library in \MCC\LIB. Add all of the required object files: C0S17.O, IDATA17.O, INT756L.O, and P17C756.O. To select more than one file at a time, hold down the “Ctrl” key on your keyboard while selecting the files with your mouse. Options can not be set on precompiled object files.

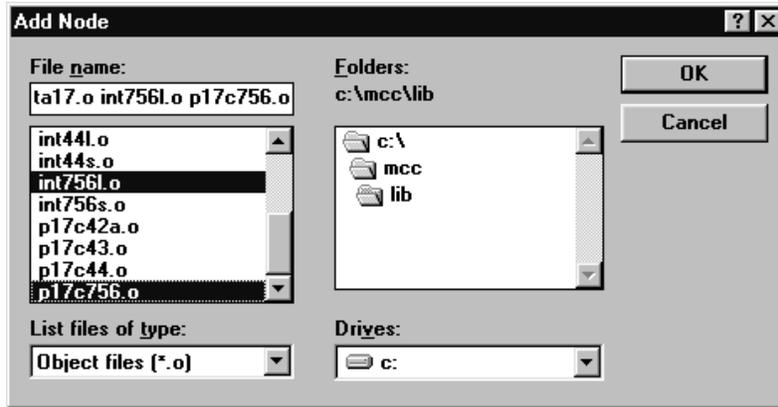


Figure 37

In the event you need to add a xxxx.lib (library) file, follow the same procedure as shown above.

Select Linker Script

Select a linker script and add it as a node. Use the linker script P17C756L.LKR in the \MCC\EXAMPLES\AD directory. Options can not be set on a linker script.

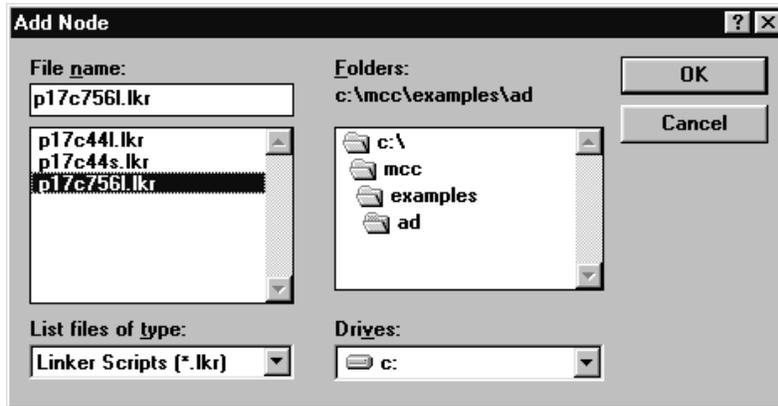


Figure 38

The Edit Project window should now look like this:

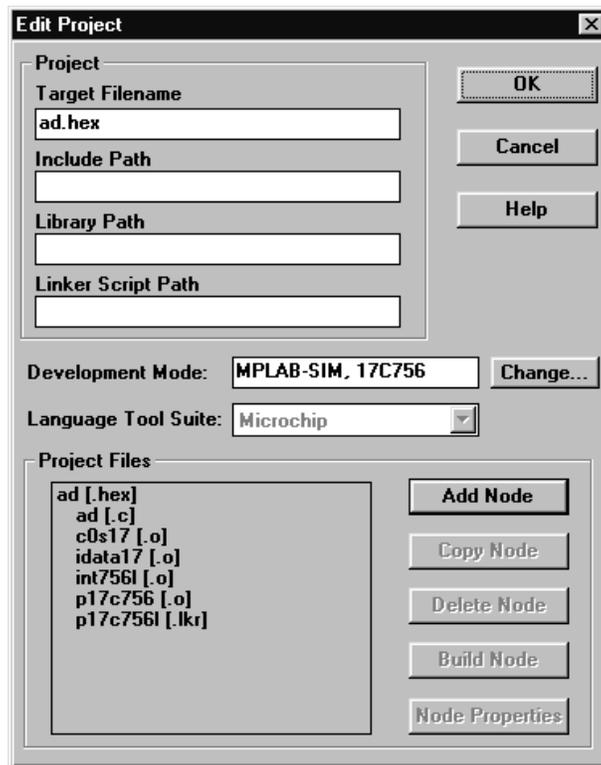


Figure 39

Press “OK” on the Edit Project Dialog.

Make Project

Select **Project > Make Project** from the menu to compile the application using MPLAB-C17 and MPLINK. A Build Results window is created that shows the command lines sent to each tool. It should look like this:

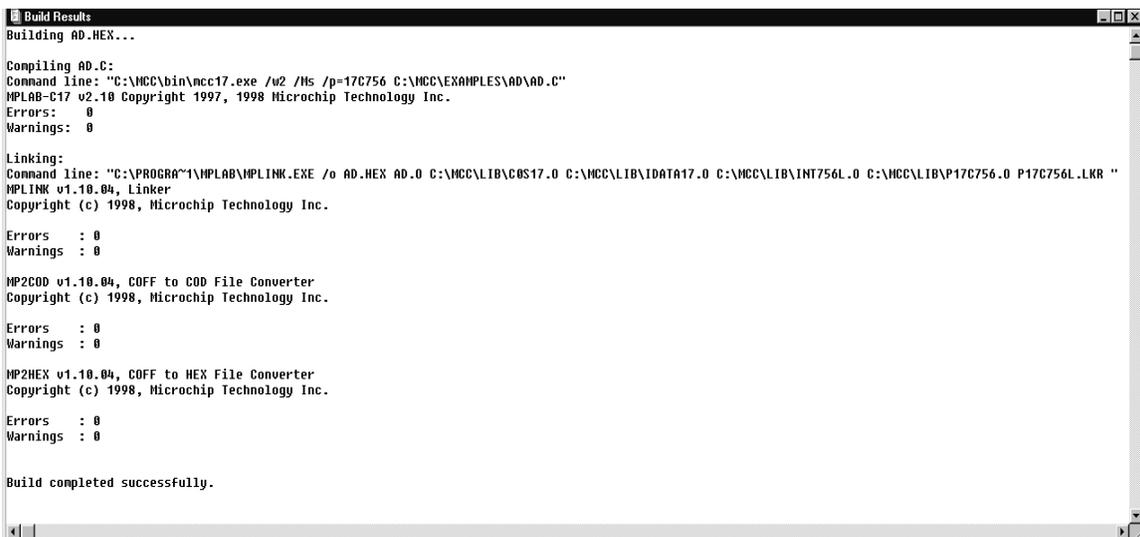


Figure 40

Troubleshooting

If this did not work, check these items:

Select Project>Install Language Tool... and check that MPLAB-C17 and MPLINK are pointed to the MCC17 . EXE and MPLINK . EXE executables.

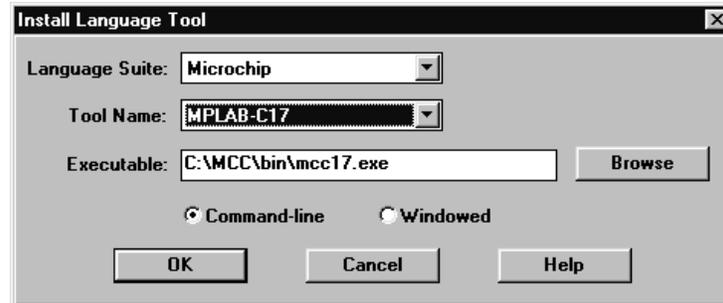


Figure 41

Project Window

Open the Window>Project window. It should look like this:

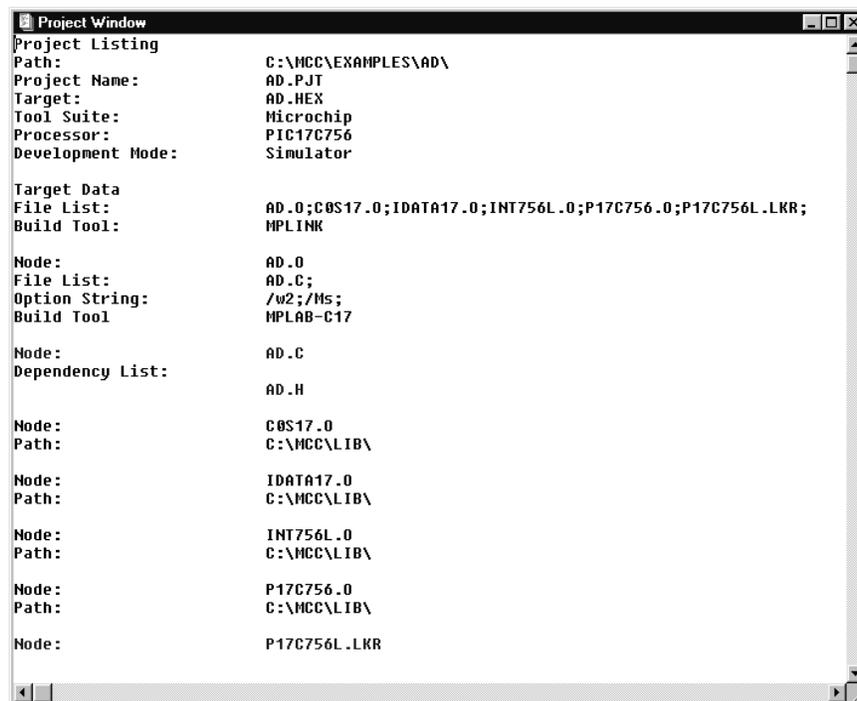


Figure 42

Summary of Setting Up MPLAB-C17 Projects

Here is a quick list of the steps to set up a new project as described above:

- Create new project with Project>NewProject
- Set project Node Properties to MPLINK
- Add Source files, setting language tool to MPLAB-C17 or MPASM
- Add Pre-Compiled Nodes (.O files and .LIB files)
- Add Linker Script node

Making a Project with Hi-Tech PIC C

This tutorial will show you how to use Hi-Tech's PIC C compiler with projects in MPLAB to build applications. If you have followed along the previous tutorials, select Project>Close Project.

Set Development Mode

Set Options>Development Mode to MPLAB-SIM simulator and select the 16C77 PICmicro for this example.

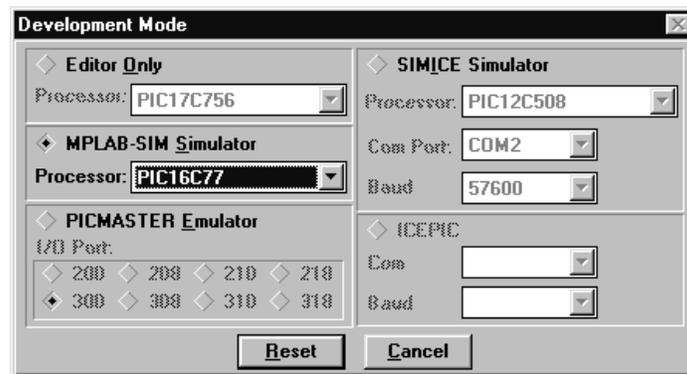


Figure 43

Install PIC C Language Tools

Make certain that PIC C is installed correctly in MPLAB. The Project>Install Language Tool dialog should be set to something similar to this for the HI-TECH tools (your executable path may be different):

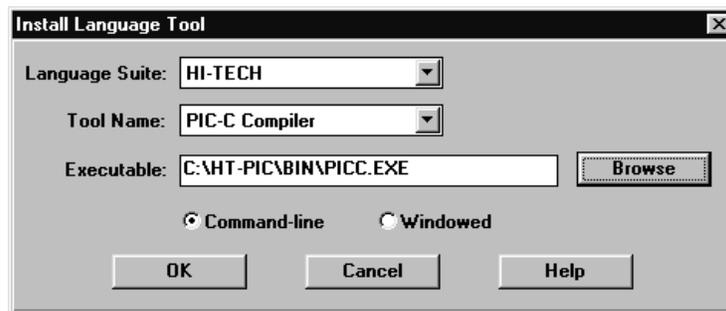


Figure 44

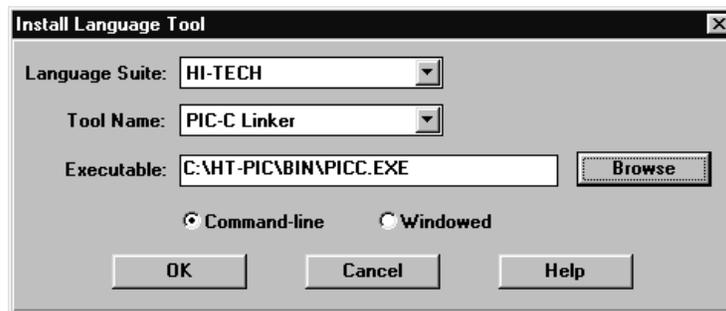


Figure 45

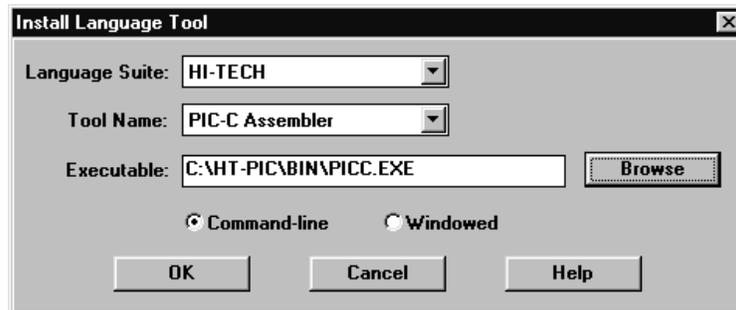


Figure 46

Use the Browse button to point to PICC.EXE on your system for the compiler, linker, and assembler.

New Project

Select **Project>New Project** and select a directory for a new project, then type in its name. Name it `SAMPLE.PJT` in the `\HT-PIC\SAMPLES` directory.

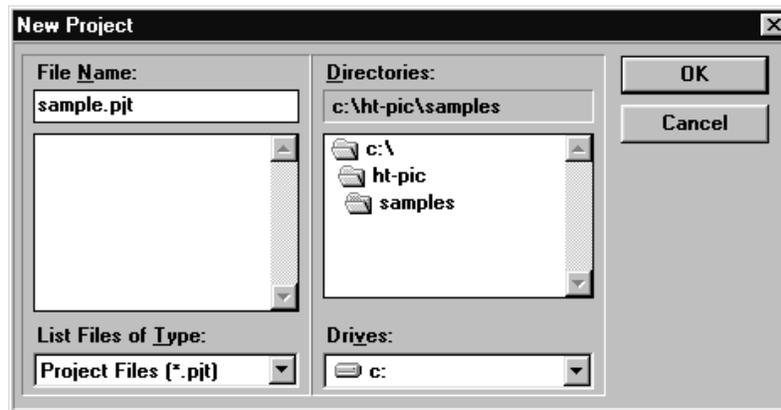


Figure 47

After setting the project name, press OK and the Edit Project dialog will be shown.

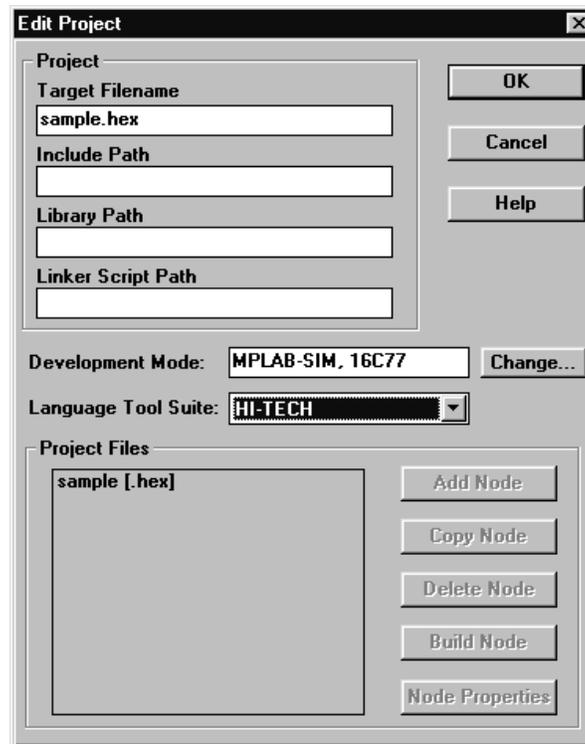


Figure 48

Make sure to set the Language Tool Suite to HI-TECH.

Set Project Options

Select the name of the project in the "Project Files" dialog of the Edit Project Dialog and press "Node Properties." Set the language tool to "PIC-C Linker" and check the "Generate Debug Info" check box. Type in "Microchip" in the Data column as shown below:

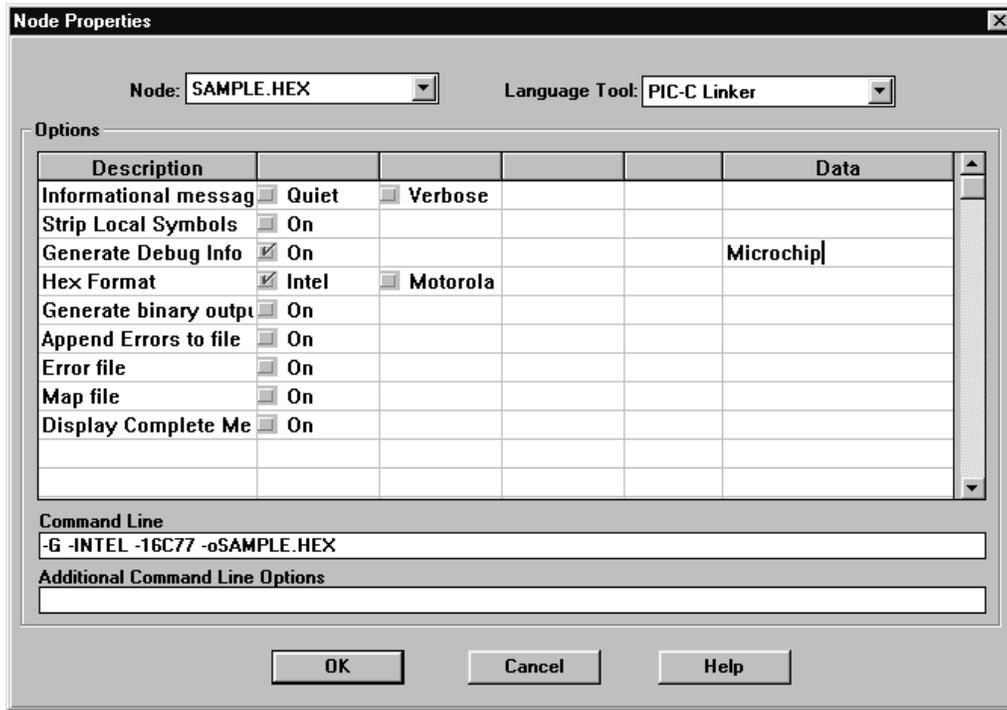


Figure 49

NOTE: The Node Properties dialog shows the command line switches for the tool, in this case PIC-C. When you first open this dialog, the checked boxes represent the default values for the tool. For this tutorial, only the debug info setting needs to be changed. Refer to the Hi-Tech documentation for more information on these command line switches.

Select "OK" in the Node Properties Dialog to return to the Edit Project Dialog.

Add First Source File

Select "Add Node" from the Edit Project Dialog. Add the source file, `SAMPLE.C` from the `\HT-PIC\SAMPLES` directory.

When the file name is shown and selected in the Add Node dialog, press "Node Properties."

Set up this dialog this way:

- Set the “Language Tool” to PIC-C Compiler
- Check the “Generate Debug Info” box
- Enter “Microchip” in the “Generate Debug Info” Data column.

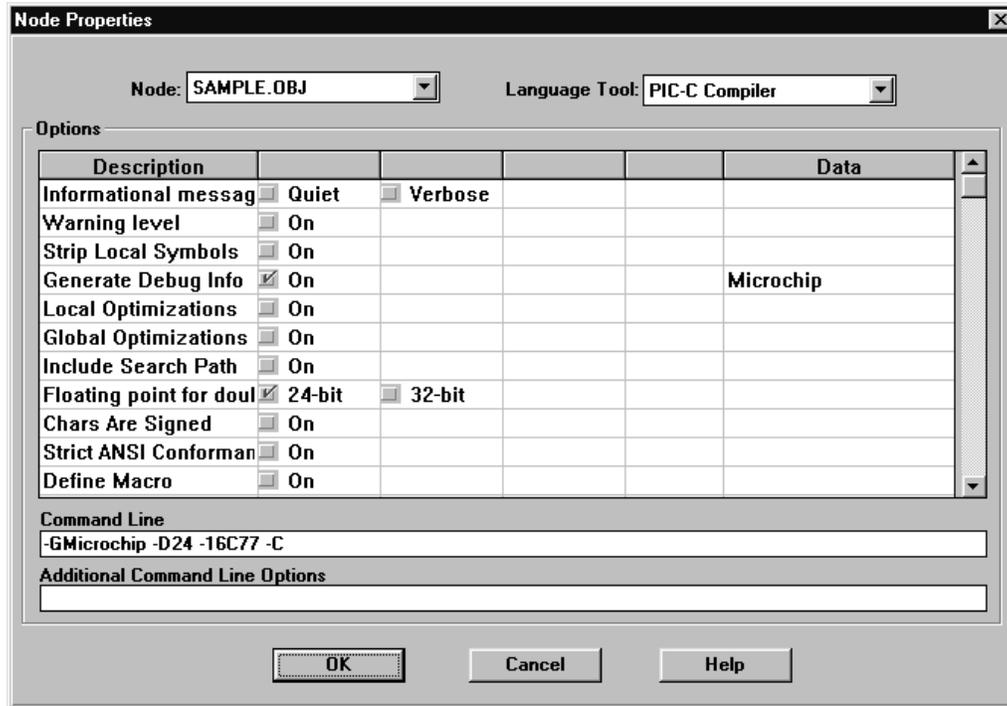


Figure 50

The “Object filename” is set to “SAMPLE . OBJ” automatically.

NOTE: The Node Properties dialog shows the command line switches for the tool, in this case PIC C. The checked boxes represent the default values for the tool. For this tutorial, only the debug info setting needs to be changed. Refer to the Hi-Tech PIC C documentation for more information on these command line switches.

Press OK, select the SAMPLE.C node and use the “Copy Node” button to add ADC.C, DELAY.C, and LCD.C with the same Node Properties as SAMPLE.C. When you are finished, the project should look like this:

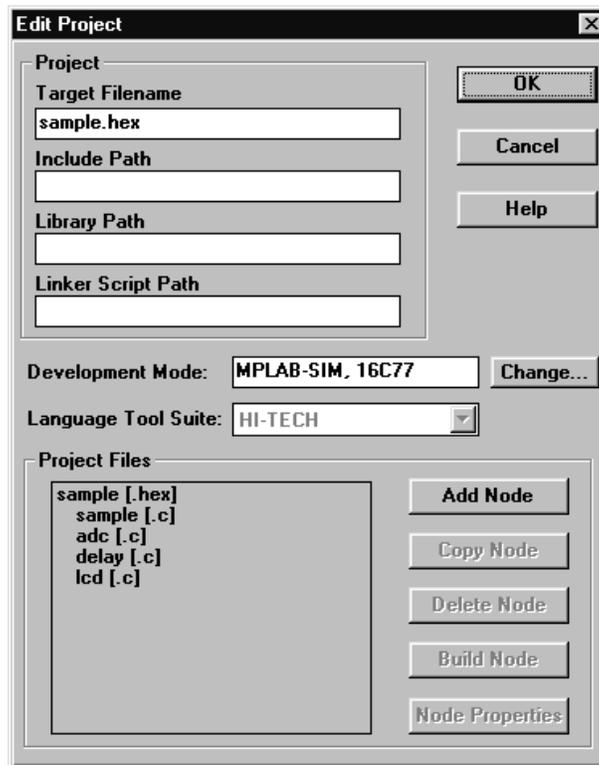


Figure 51

Make Project

Select **Project>Make Project** from the menu to compile the application using the Hi-Tech compiler and linker. A Build Results window is created that shows the command lines sent to each tool. It should look like this:

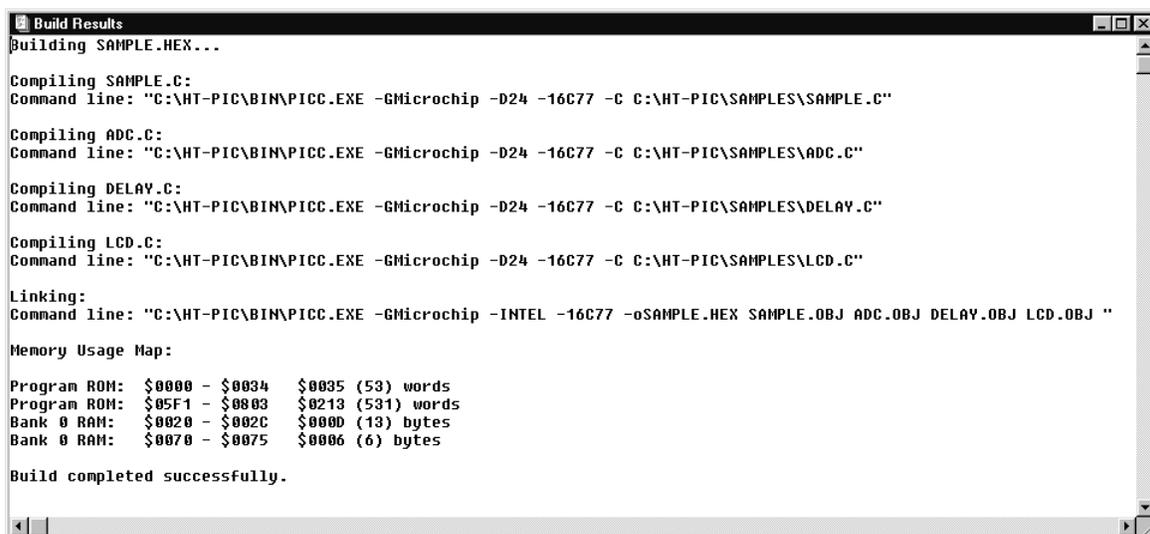


Figure 52

Troubleshooting

If this did not work, check these items:

Select Project>Install Language Tool... and check that PIC C Compiler and PIC C Linker are both pointing to the PICC.EXE executable.

Project Window

Open the Window>Project window. It should look like this:

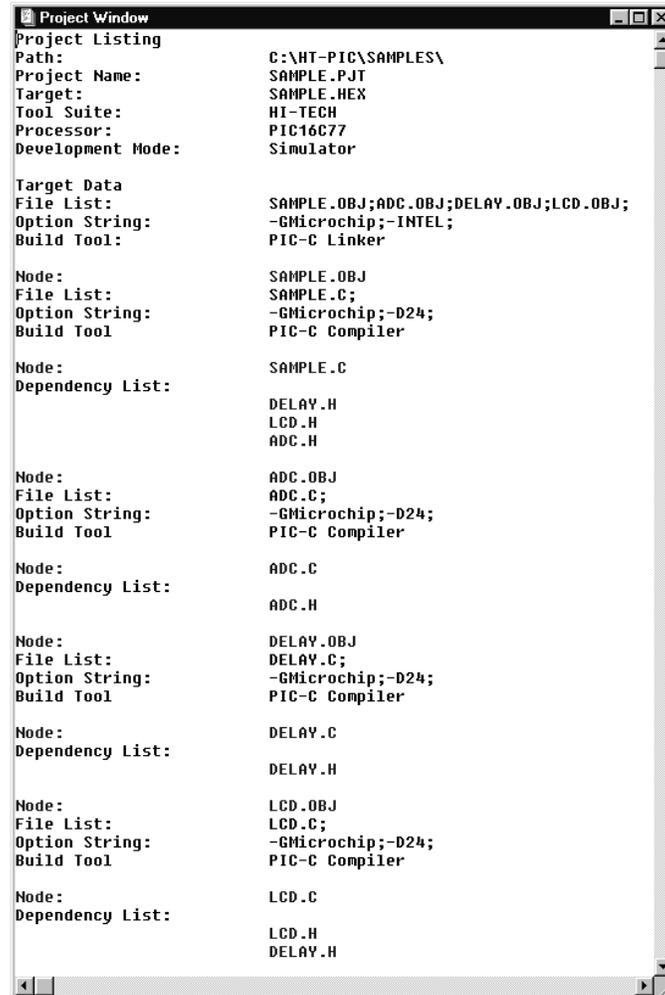


Figure 53

Summary of Setting Up PIC C Projects

Here is a quick list of the steps to set up a new project as described above:

- Set up Language Tools for PIC C Compiler, Linker, and Assembler
- You may need to set the "Include File Directory" to \HT-PIC\H (or where PIC C include files are installed on your system)
- Create new project with Project>NewProject
- Turn on "Generate Debug Info" for project node
- Set project Node Properties to PIC C Linker
- Add Source files, setting language tool to PIC C Compiler or Assembler
- Turn on "Generate Debug Info" for each source node
- Set "Generate Debug Info" Data to "Microchip" for each source node