LAPIS SEMICONDUCTOR

IDEU8
User's Manual

Program Development Support Software

Issue Date: Oct. 2011

NOTICE

No copying or reproduction of this document, in part or in whole, is permitted without the consent of LAPIS Semiconductor Co., Ltd.

The content specified herein is subject to change for improvement without notice.

The content specified herein is for the purpose of introducing LAPIS Semiconductor's products (hereinafter "Products"). If you wish to use any such Product, please be sure to refer to the specifications, which can be obtained from LAPIS Semiconductor upon request.

Examples of application circuits, circuit constants and any other information contained herein illustrate the standard usage and operations of the Products. The peripheral conditions must be taken into account when designing circuits for mass production.

Great care was taken in ensuring the accuracy of the information specified in this document. However, should you incur any damage arising from any inaccuracy or misprint of such information, LAPIS Semiconductor shall bear no responsibility for such damage.

The technical information specified herein is intended only to show the typical functions of and examples of application circuits for the Products. LAPIS Semiconductor does not grant you, explicitly or implicitly, any license to use or exercise intellectual property or other rights held by LAPIS Semiconductor and other parties. LAPIS Semiconductor shall bear no responsibility whatsoever for any dispute arising from the use of such technical information.

The Products specified in this document are intended to be used with general-use electronic equipment or devices (such as audio visual equipment, office-automation equipment, communication devices, electronic appliances and amusement devices).

The Products specified in this document are not designed to be radiation tolerant.

While LAPIS Semiconductor always makes efforts to enhance the quality and reliability of its Products, a Product may fail or malfunction for a variety of reasons.

Please be sure to implement in your equipment using the Products safety measures to guard against the possibility of physical injury, fire or any other damage caused in the event of the failure of any Product, such as derating, redundancy, fire control and fail-safe designs. LAPIS Semiconductor shall bear no responsibility whatsoever for your use of any Product outside of the prescribed scope or not in accordance with the instruction manual.

The Products are not designed or manufactured to be used with any equipment, device or system which requires an extremely high level of reliability the failure or malfunction of which may result in a direct threat to human life or create a risk of human injury (such as a medical instrument, transportation equipment, aerospace machinery, nuclear-reactor controller, fuel-controller or other safety device). LAPIS Semiconductor shall bear no responsibility in any way for use of any of the Products for the above special purposes. If a Product is intended to be used for any such special purpose, please contact a ROHM sales representative before purchasing.

If you intend to export or ship overseas any Product or technology specified herein that may be controlled under the Foreign Exchange and the Foreign Trade Law, you will be required to obtain a license or permit under the Law.

Windows is a registered trademark of Microsoft Corporation (USA) in USA and other countries, and other product names and company names are trademarks or registered trademarks.

Copyright 2008 - 2011 LAPIS Semiconductor Co., Ltd.

Contents

1. Introduction 1

1.1 About This Product 1
1.1.1 Overview 1
1.1.2 Features 1
1.2 Manual Organization 2
1.3 Notation 3

2. Installing IDEU8 4

2.1 IDEU8 Components and System Requirements 4
2.2 Installing IDEU8 4

3. Tutorial 5

3.1 Creating New Project 5
3.1.1 Setting Up Project 5
3.1.2 Adding to Project 6
3.2 Specifying Command Line Options 7
3.2.1 Compiler and Assembler Options 7
3.2.2 Target Options 7
3.3 Build 7
3.4 Jumping to Error Locations 8
3.5 Debugging Programs 8

4. Project Management 10

4.1 Working with Projects 10
4.1.1 Creating Project 10
4.1.2 Saving Project 10
4.1.3 Reading in Project 11
4.2 Project Window 11
4.2.1 File Tree 11
4.2.2 Context Menu 12
4.3 File Management 14
4.3.1 Adding File 14
4.3.2 Deleting a File 15
4.3.3 File Option Sets 15
4.3.4 Link Order 15
4.4 Specifying Compiler and Assembler Options 16
4.4.1 *General* Tab 17
4.4.2 *Directory* Tab 18
4.4.3 *Optimization* Tab 20
4.4.4 *Code Generation* Tab 21
4.4.5 *Warnings* Tab 22
4.4.6 *Listing* Tab 24
4.4.7 *Memory settings* Tab 26
4.4.8 *Macro* Tab 27

4.5 Specifying Target Options 27
4.5.1 *General* Tab 28
4.5.2 *Segment* Tab 29
4.5.3 *Memory settings* Tab 31
4.6 Defining Option Sets 33

5. Building Programs 34

5.1 Overview 34
5.2 Build 34
5.3 Rebuild 34
5.4 Build Selected File 34
5.5 Canceling Build Commands 34
5.6 Deleting Object Files 35
5.7 Loading DTU8 Debugger 35
5.7.1 Modifying Program 35
5.7.2 Link Between IDEU8 and Debugger 35
5.8 *Output* Window 35
5.8.1 Copying Messages to Windows Clipboard 35
5.8.2 Opening the corresponding file 35
5.8.3 Saving Messages 36
5.8.4 Clearing Messages from Window 36
5.8.5 Dock 36

6. Other Functions 37

6.1 Creating Text Files 37
6.2 *Editor* Window 38
6.2.1 *Edit* Menu 38
6.3 Specifying System Paths 40
6.4 Other Environment Settings 41
6.5 Printing Files 43
6.6 On-Line Help 43
6.7 Displaying Version Information 44

Index 45

1. Introduction

1.1 About This Product

1.1.1 Overview

The U8 integrated development environment (hereinafter abbreviated to IDEU8) provides the programmer with a powerful software environment for developing microcontroller application programs for 8-bit RISC processors based on the nX-U8 core.

IDEU8 integrates the compiler, MACU8 assembler, DTU8 debugger, and other software packages into a single development support environment and adds project management, a programming editor, and other functionality boosting development efficiency. It also eliminates the need to manually create make and batch files.

This single integrated environment covers all program development steps from typing the source code in the built-in editor through to compiling and linking. It even provides an integrated debugger for debugging the resulting program. What were separate development support software tools are now a single package.

Note: The only component not yet integrated is the LIBU8 librarian from the MACU8 assembler package.[1]

1.1.2 Features

■ **Windows Support**

IDEU8 permits program development in the modern graphical user interface (GUI) environments Windows. It runs as a 32-bit application, for maximum throughput.

■ **Project Management**

IDEU8 provides overall management of source code file dependencies, compiler and assembler command line options, and other project particulars.

The *Project* window provides maintenance facilities for adding source code files, include files, and library files to a project and for deleting them. It graphically displays these files in a treelike hierarchical layout.

This window also provides similar facilities for adding document files containing background information for the project.

[1] Excluding the LIBU8 librarian in the MACU8 Assembler Package.

operation with easy-to-use dialog boxes.

It also supports the saving of different groups of such settings for use as file- and target-specific option sets.

■ Built-In Editor

When the compiler or assembler issues an error message, IDEU8 provides single-key access for editing the corresponding source code line in its built-in editor.

The environment also supports tag jumps to the lines triggering build errors.

■ Extensive Built-In Help

The extensive help available includes the quick help function displaying a brief note on the status line describing the currently selected menu command plus on-screen text descriptions of functions, usage, and other topics.

1.2 Manual Organization

For further details on the original development support packages used in the IDEU8 environment, refer to the following manuals.

■ CCU8 Compiler Package

- CCU8 Compiler User's Manual
- CCU8 Compiler Programming Guide
- RTLU8 Run-Time Library User's Manual

■ MACU8 Assembler Package

- MACU8 Assembler Package User's Manual

■ DTU8 Debugger

- DTU8 Debugger User's Manual

This manual consists of the following Chapters.

■ Chapter 1 Introduction

This is the Chapter that you are now reading.

■ Chapter 2 Installing IDEU8

This Chapter describes the system requirements and installation procedure.

■ Chapter 3 Tutorial

This Chapter describes the operational flow for loading IDEU8, opening a project, setting compiler options, and debugging the program.

■ **Chapter 4 Project Management**

This Chapter first describes the *Project* window in detail. It then gives the procedures for adding and deleting source code files and for saving and reading in project (.PID) files.

■ **Chapter 5 Building Programs**

This Chapter describes the build menu commands and the *Output* window.

■ **Chapter 6 Other Functions**

This Chapter describes the *Editor* window, specifying paths for the component executables, using the on-line help, and other miscellaneous topics.

■ **Index**

This is an alphabetical index to this manual. This is particularly useful for checking items for configuring the individual build tools.

1.3 Notation

This manual uses bold face for key tops and indicates key combinations with plus signs. The notation *Shift*+*F1*, for example, means to hold down the *Shift* key and press the *F1* key.

There are no icons or other notational items that require explanation.

2. Installing IDEU8

2.1 IDEU8 Components and System Requirements

The following are the IDEU8 components and the system requirements for using IDEU8.

■ Operation Environment

- 32-bit version of Windows XP, Windows Vista or Windows 7, and 64-bit version of Windows Vista or Windows 7.
- Hardware environment appropriate for the above operating system
- Graphic adapter and display with at least SVGA (800 × 600) resolution
- At least 20 megabytes of hard disk space free
- CD-ROM drive
- Mouse

2.2 Installing IDEU8

The IDEU8 setup program includes not only the IDEU8 executable and related support files, but also the CCU8 compiler for building object files, the DTU8 debugger for debugging them, and all other software necessary between these two extremes. As a result, this one program therefore installs the entire development environment.

■ Installing from Setup CD

1. Insert the nX-U8 Development Tools setup CD in the CD-ROM drive.
2. A setup utility (SETUP.EXE) starts automatically. If the setup utility doesn't start, start setup.exe on root directory of setup CD.
3. Follow the instructions on the screen.

The setup program displays a message box when installation is complete. Press the *OK* button to exit.

■ Uninstalling

1. Open the "Add/remove applications" control panel.
2. Select IDEU8 from the list.
3. Press the *Remove* button to uninstall IDEU8 and related software.

3. Tutorial

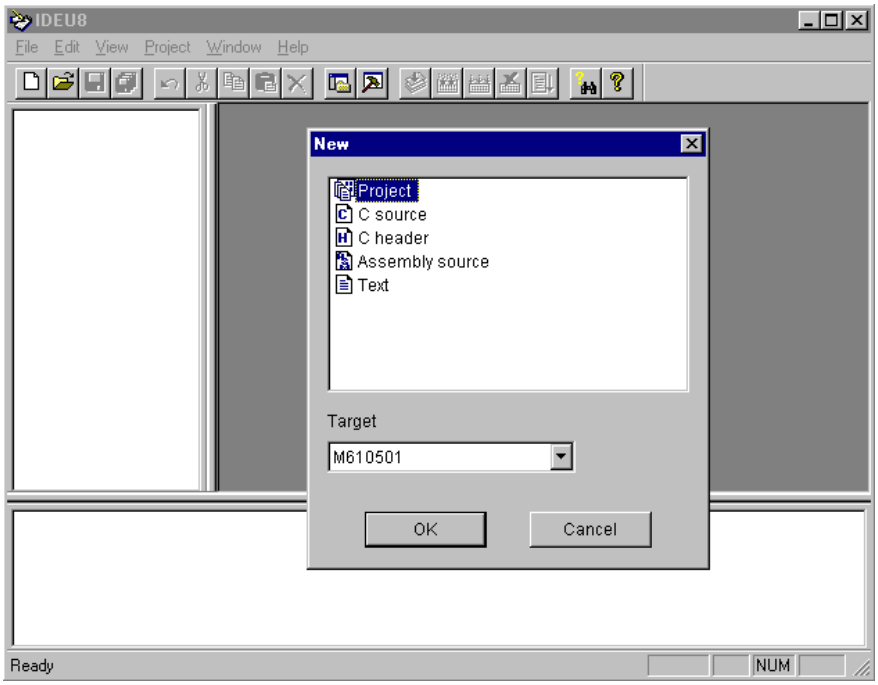
3.1 Creating New Project

Before you can use IDEU8, you must create an IDEU8 project file. This file contains the following information necessary for building (the entire sequence of compiling, assembling, and linking files) an executable program.

- List of source code and library files necessary for building the executable program
- Command line options for the build tools used to construct the executable program
- Database storing the dependency relationships between source code files and include files during builds

3.1.1 Setting Up Project

Choosing the *File* menu *New*... menu command displays the following dialog box.



Select Project from the list at the top of the dialog box and choose the target microcontroller model from the Target MCU list box at the bottom.

Pressing the OK button opens the Create New Project dialog box for specifying the project (.PID) file.

The base name specified here (everything except the extension.PID) for the project file becomes the base name for the final target files, the executable program resulting from the build. Naming the project file Skeleton.pid, for example, produces final target files

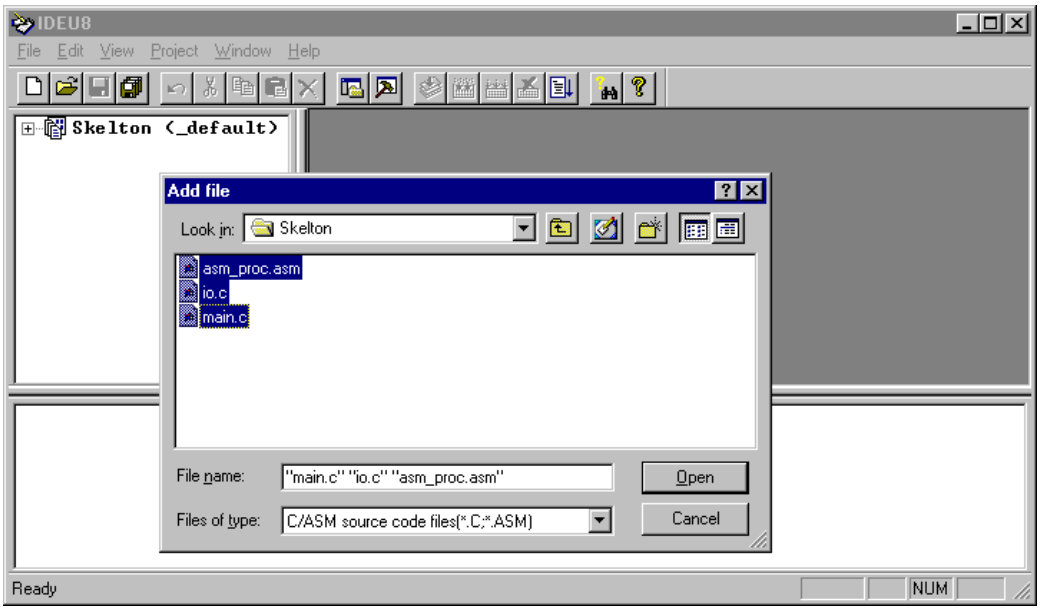
Skeleton.abs, Skeleton.hex, etc.

Note that File menu *Open…* menu command is available for opening an existing IDEU8 project file.

3.1.2 Adding to Project

To create a new source code file using the built-in text editor, choose the File menu *New...* menu command and specify the file type. Choosing C source code, for example, opens an editor window configured for editing a new .C file.

This new editor file and any existing source code files must be added to the IDEU8 project file for building the program. Selecting the Project menu *Add file...* menu command displays the following dialog box.



IDEU8 judges source code files, library files, and other files necessary for building the program by their file extensions.

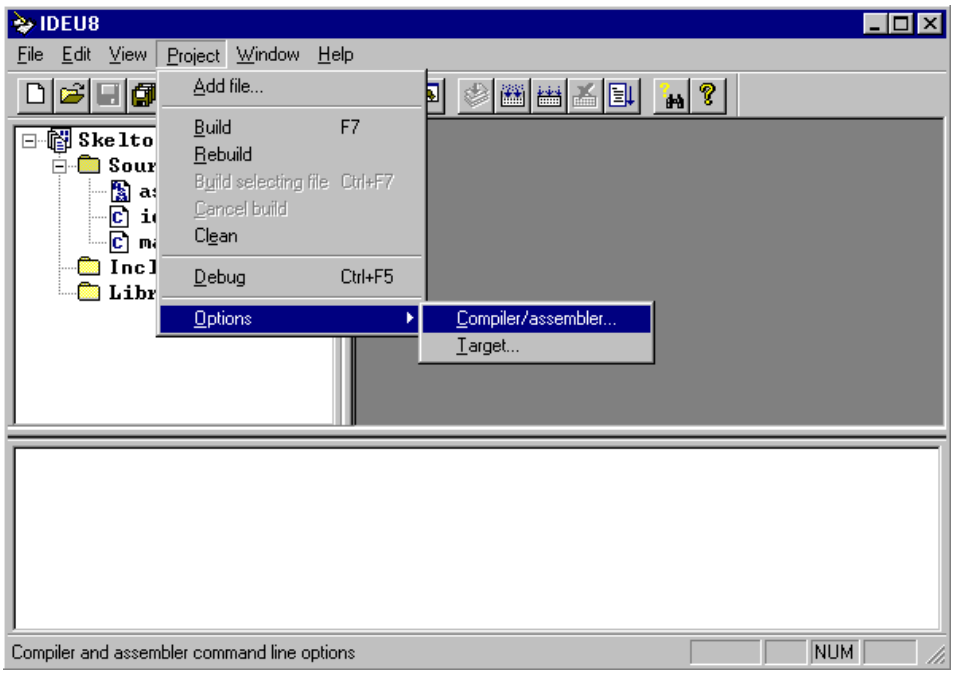
- .C C source code file, input for the compiler
- .ASM Assembly language source code file, input for the assembler
- .OBJ Object file, input for the linker
- .LIB Library object file, input for the linker

The Project window's Source folder is for C and assembly language source code files; the Library folder, for .OBJ and .LIB object files. IDEU8 automatically maintains the Include folder, adding files as the compiler and assembler detect files included in the source code files.

Files with extensions other than the four given above go in the root folder for the project. Candidates include text files documenting the project and data files for other applications.

3.2 Specifying Command Line Options

The CCU8 compiler, RASU8 assembler, RLU8 linker, and other build tools provide a wide variety of command line options for tailoring operation to meet the needs of the user application program under development. These must be specified before building the application. IDEU8 provides dialog boxes for quickly and easily specifying these command line options.



3.2.1 Compiler and Assembler Options

The Project menu Options|*Compiler/assembler...* menu command dialog box provides build tool option settings for generating intermediate object files with the compiler and assembler.

For further details, see Section 4.5 "Compiler/Assembler Option Settings."

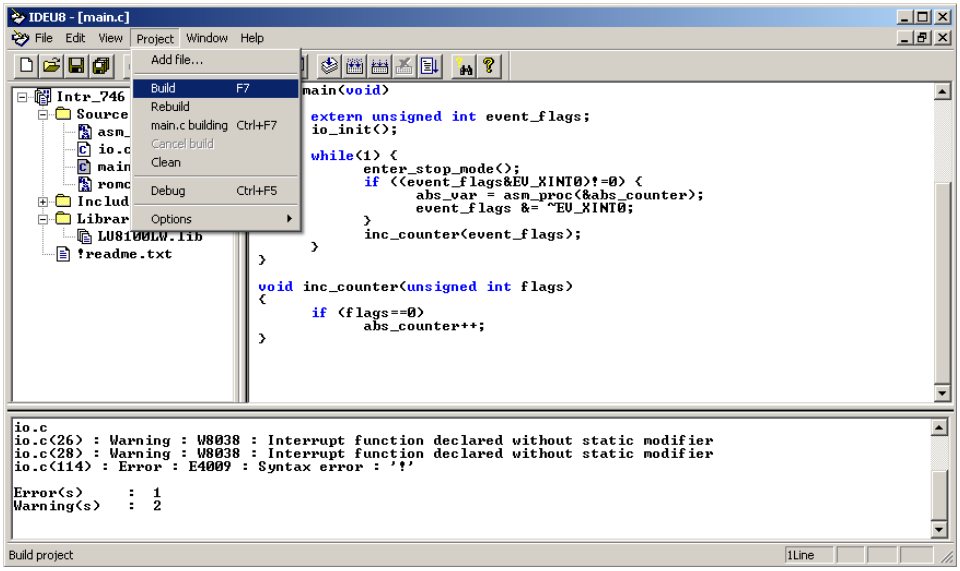
3.2.2 Target Options

The Project menu Options|*Target...* menu command dialog box provides build tool option settings for generating the target file with the linker and object converter.

For further details, see Section 4.6 "Specifying Target Options."

3.3 Build

The Project menu *Build* menu command builds the final executable program, processing the project's C source code files, assembly language source code files, and library files as necessary.



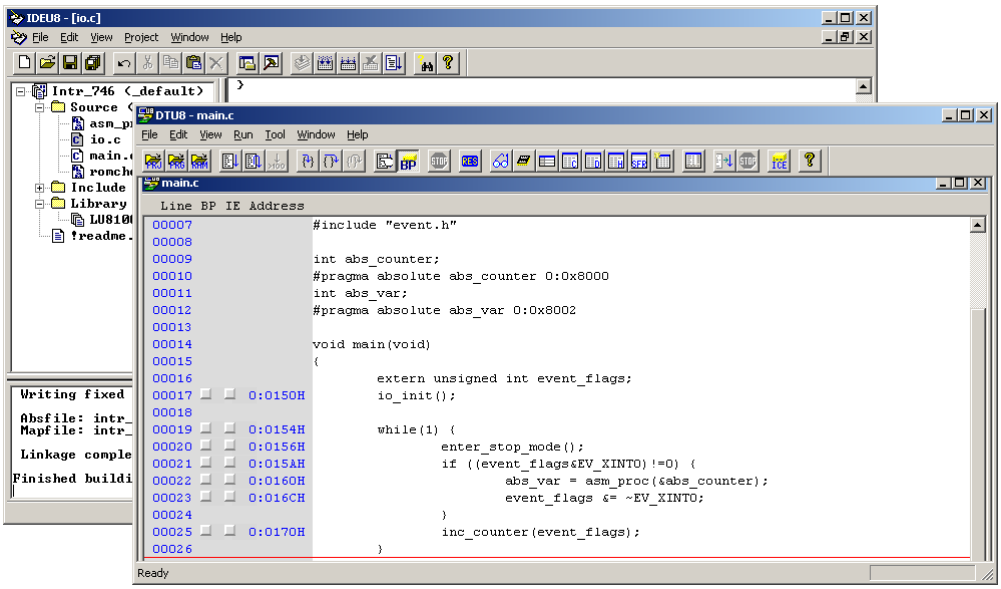
The basic step is simply specifying the Project menu *Build* menu command. The first time, this menu command compiles and assembles all source code files and then links the resulting object files to create the final target file. Subsequent uses first check file time stamps to see whether they have changed since the preceding build and then limit processing to the minimum required to incorporate interim updates into the final target file. The command compiles and assembles only those source code files that have been modified or include header files that have been modified and then links the object files.

3.4 Jumping to Error Locations

The Output window displays build progress, including any error and warning messages from the compiler and assembler. Double-clicking on an error or warning message in this window (or moving the cursor there and pressing the Enter key) opens the corresponding source code file in an Editor window with the cursor on the corresponding line.

3.5 Debugging Programs

The Project menu *Debug* menu command runs the debugger from within the IDEU8 environment. This menu command first builds an absolute object (.ABS) file and passes it to the debugger. The debugger automatically connects to the emulator, downloads the program code, and takes the other steps required to start debugging.



When all build operations on the source code files are complete, yielding a final target (ABS) file ready for downloading to the in-circuit emulator, you are ready to debug the program. To start, choose the *Project* menu *Debug* menu command.

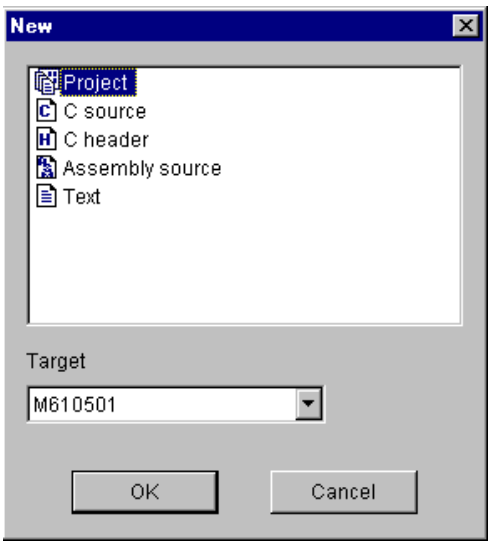
4. Project Management

4.1 Working with Projects

IDEU8 starts off with no project file open.

4.1.1 Creating Project

When IDEU8 loads, the first thing to do is create a project. Choosing the *File* menu New... menu command displays the following dialog box.



Selecting “Project” from the list, specifying the target microcontroller, and pressing the *OK* button creates a new project. The project name then becomes the base name for the final target (ABS) file, map file, etc.

Here “target microcontroller” is the model identification for use with the CCU8 compiler's /T command line option.

Creating a new project enables the *Project* menu *Add file...* and *Project* window context menu *Add to project...* menu commands for adding source code files to the project.

4.1.2 Saving Project

IDEU8 saves projects to disk as project (.PID) files. The following methods are available for saving the current project.

- Choose the *File* menu *Save project* menu command.
- Select the root folder for the project in the Project window, choose the *File menu Save* menu command.
- Select the root folder for the project in the Project window, choose the *File menu Save as...* menu command, and, if desired, specify a new name for the project file.

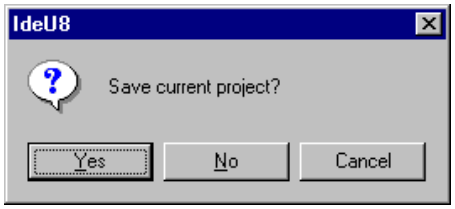
Note: Changing the project name also changes the target file name, so the next build

requires at least the linking step to create that new target file.[2]

4.1.3 Reading in Project

To read in a saved project (.PID) file and thus reproduce the IDEU8 state for that project, choose the *File* menu *Open...* menu command and scroll the *File type* list box in the dialog box that appears to Project files (*.PID).

IDEU8 can open only one project at a time. If there is one already open, attempting to open another produces the following confirmation message if there have been changes in that project's files.



4.2 Project Window

Choosing the View menu *Project* menu command or pressing the Alt+0 (zero) key combination opens the Project window. If there is no IDEU8 project open, this window is empty.

Creating a new project or reading in a project (.PID) file then displays four folders in a hierarchical layout in the *Project* window.

4.2.1 File Tree

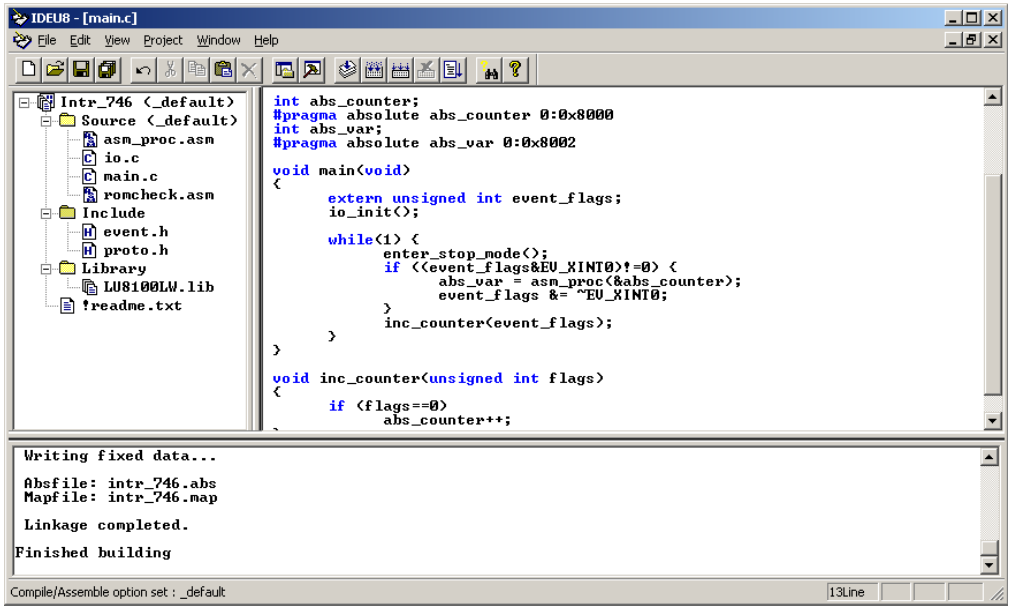
The *Project* window displays the files in a project in a treelike hierarchy with a Project folder at the top and three folders—Source, Include, and Library—under that.

Folder name	Extension	Explanation
Project Folder	(not shown)	This gives the project and target names.
Source Folder	.C and .ASM	These are the input files for the compiler and assembler for builds.
Include Folder	(any)	These are files that IDEU8 detects the source code files referencing during builds.
Library Folder	.OBJ and .LIB	These files are linked in during builds.

File with extensions other than .C, .ASM, .OBJ, or .LIB go in the root folder for the project.

Clicking on the + button to the left of a folder expands the display to include any files in that folder and changes the plus sign inside the button to a minus sign.

[2] If the name of a project is changed, the name of the target file is also changed. Therefore, in the next build, such link processing as will change the old names to the new ones is at least performed.



IDEU8 automatically sorts the files in the Source folder into dictionary order and process them in this order, not the order in which they were added.

The Include folder contains include files and other files that IDEU8 detects the source code files referencing during builds. It sorts them into dictionary order. There is a toggle controlling display of files in the system include directory, specified with the File menu Environment settings|System path menu command. See Section 6.4 "Miscellaneous Functions" for the setting procedure.

For further details on the Library folder display, see Section 4.3.4 "Link Order." The Project window has a context menu accessible with the following standard Windows operations.

4.2.2 Context Menu

The Project window has a context menu accessible with the following standard Windows operations.

- Right-clicking on a file or folder in the Project window
- Selecting a file or folder in the Project window and pressing the Shift+F10

■ Build

This menu command performs a build operation on the currently selected source code file.

If a source code file is selected, this menu command covers only that file. If the Source folder itself is selected, this menu command builds the final target file.

■ Open

This menu command opens the selected file in an editor window if that file is in the Source or Include folder or has the extension .TXT or .LOG. Otherwise, it launches the application that Windows has assigned to the file extension.[3]

[3] A file with an extension .TXT or .LOG always opens in an IDEU8 editor window.

■ Add to project...

This menu command adds the specified file to the project. This is the same as the Project menu *Add file...* menu command.

■ Delete from project

This menu command deletes the currently selected file from the project.

■ Settings...

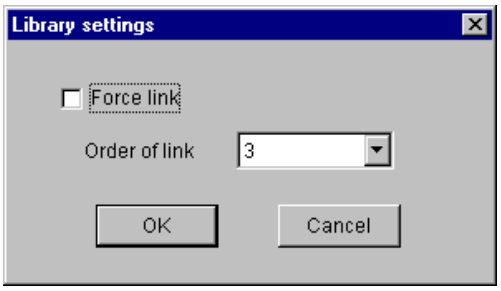
If the Source folder is selected, this menu command selects the set of default compiler/assembler options for all source code files.

If a source code file in the Source folder is selected, this menu command selects the set of compiler/assembler options specific to that file.

If the Project folder is selected, this menu command selects the set of target options.

For further details, see Section 4.6 "Specifying Target Options."

If a .LIB file in the Library folder is selected, this menu command displays the dialog box for specifying library file link control. For further details on using this dialog box, see Section 4.3.4 "Link Order."



■ Dock

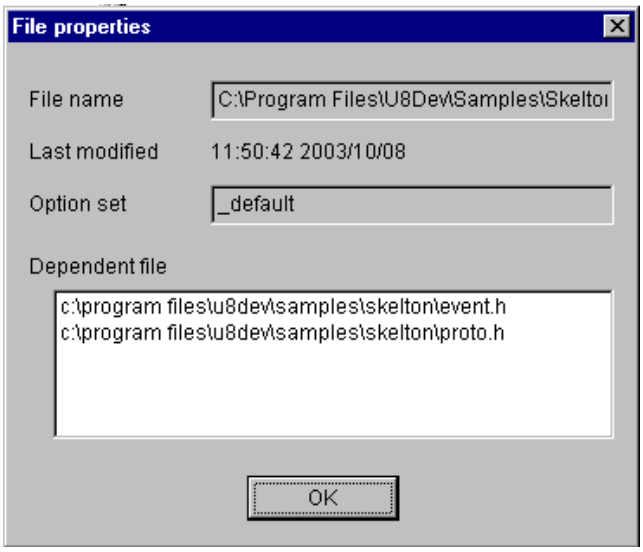
This menu command toggles docking of the Project window at the left side of the IDEU8 main window.

■ Hide

This menu command hides the Project window, but does not close the project file.

■ Properties...

This menu command displays the following dialog box for displaying file properties.



The File name field gives the full absolute path to the selected file or, if the Project folder is selected, the project (.PID) file information.

The Date last modified field gives the file time stamp, the date and time of the last update for the selected file.

The Option set name field gives the option set assigned to the selected file or project file.

The Dependent file field lists files referenced during builds. For a source code file in the Source folder, these represent included files. For the Project folder itself, the list consists of all object and library files for linking.

4.3 File Management

IDEU8 saves the following information in project files: target microcontroller, source code files in the *Project* window, CCU8 compiler and RASU8 assembler options, window states, and environment settings. The name for the project file is the one specified when it was created.

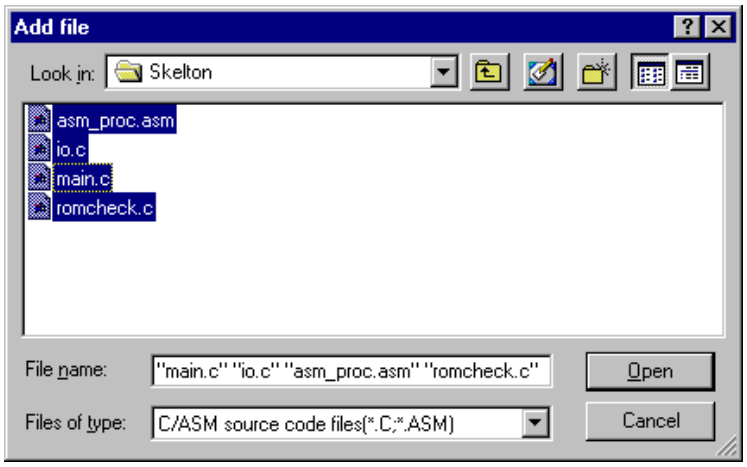
Setting up separate project files for each product under development allows switching at the project level.

4.3.1 Adding File

The following methods are available for adding source code files, object files, library files, and document files to a project.

- Project menu *Add file...* menu command
- Project window context menu *Add to project* menu command
- Selecting the Project window and pressing the Ins key

These menu commands display the following dialog box for selecting the file to add.



The Project menu *Add file...* menu command supports the simultaneous selection of multiple files within the same folder. The file extension determines the destination folder in the Project window: Source, Library, or root.

4.3.2 Deleting a File

To delete a file from the project, select it in the *Project* window and choose the context menu *Delete from project* menu command.[4]

4.3.3 File Option Sets

IDEU8 supports option sets at two levels: the target option set for the entire project and compiler and assembler command line option sets for the individual source code files.

Selecting the project name in the *Project* window and choosing the context menu *Settings...* menu command specifies the target option set. The name of the specified option set appears next to the project name in the Project window.

Selecting the Source folder name in the Project window and choosing the context menu Settings… menu command specifies the default option set for the CCU8 compiler and RASU8 assembler. The name of the specified option set appears next to the Source folder name in the Project window.

Otherwise, the command specifies the option set for the selected source code file. If the specified option set is other than the default, its name appears next to the file name in the Project window.

To view the option set for a file, select it and choose the context menu *Properties...* menu command.

4.3.4 Link Order

IDEU8 specifies the following files in the object_files[5] field of the RLU8 linker:

- Object (.OBJ) files generated by compiling and assembling source code files in the Source folder Project window
- Object (.OBJ) files in the Project window Library folder

[4] This menu command merely deletes the file from the project. It does not erase the actual disk file.

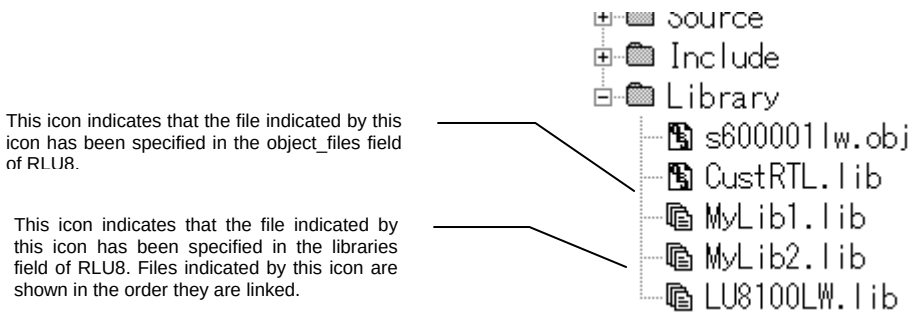
[5] The object_files field of the RLU8 linker is the field listing the files for the linker to link. For further details, refer to the RLU8 description in the MACU8 User's Manual.

- Library (.LIB) files in the Project window Library folder marked with the Forced linking check box on the context menu *Settings…* menu command dialog box
- Additional files marked with the Add to object field on the Project menu Options|*Target...* menu command dialog box

IDEU8 specifies the following files in the libraries[6] field of the RLU8 linker:

- The remaining Library (.LIB) files in the Project window Library folder
- Additional files marked with the Add to library check box on the Project menu Options|*Target...* menu command dialog box

The context menu *Settings…* menu command dialog box specifies the order for library files in the Project window Library folder. The Add to library check box on the Project menu Options|*Target...* menu command dialog box then determines whether the library file is searched.[7]



4.4 Specifying Compiler and Assembler Options

Before building the final target (ABS) file for the program, you must first specify command line options to pass to the CCU8 compiler and RASU8 assembler with the *Project* menu *Options* → *Compile/assemble...* menu command sequence.

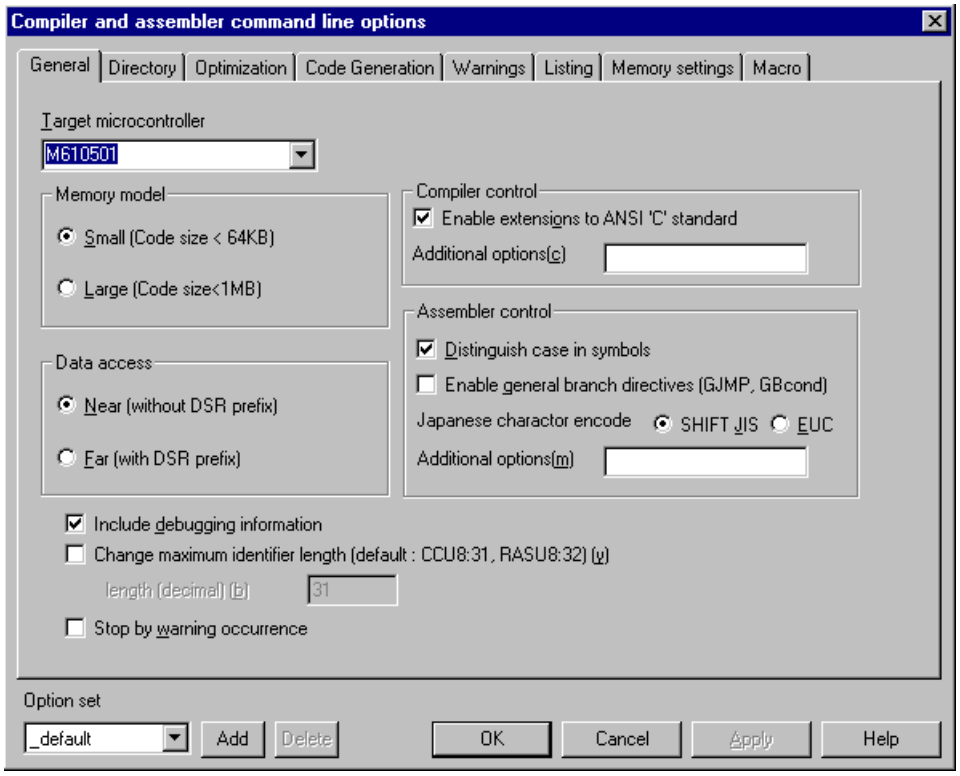
The dialog box that appears controls key build tool options affecting the creation of intermediate object files.

[6] The libraries field of the RLU8 linker is the field listing the library files for the linker to search for unresolved external references. For further details, refer to the RLU8 description in the MACU8 User's Manual.

[7] Note that library files are searched in the order specified in the libraries field of the RLU8 linker.

4.4.1 *General* Tab

Clicking the *General* tab in the compiler and assembler command line options dialog box switches to the following contents.



■ **Target microcontroller**

Selecting the microcontroller name of the list specifies the /T command line option to the Ccu8 compiler.

■ **Memory Model**

These radio buttons specify the memory model for the target microcontroller.

Selecting *SMALL* specifies the /MS command line option to the CCU8 compiler and RASU8 assembler, limiting the code memory space available to 64 kilobytes.

Selecting *LARGE* specifies the /ML command line option to the CCU8 compiler and RASU8 assembler, limiting the code memory space available to 1 megabytes.

■ **Data access**

This option is the default data access option.

Selecting the *Near (without DSR prefix)* radio button specifies the /near command line option to the CCU8 compiler and the /DN command line option to the RASU8 assembler.

Selecting the *Prohibition of __far description/Deterrence of evacuation of DSR in an interruption function* check box specifies the /nofar command line option to the CCU8 compiler.

Selecting the *Far (with DSR prefix)* radio button specifies the /far command line option to the CCU8 compiler and the /DF command line option to the RASU8 assembler

■ **Compiler Control**

No selecting the *Enable extensions to ANSI 'C' standard* check box specifies the /Za command line option to the CCU8 compiler.

The *Additional options* input field is for adding CCU8 compiler options not supported by IDEU8.

■ **Assembler Control**

Selecting the *Distinguish case in symbols* check box specifies the /CD command line option to the RASU8 assembler.

Selecting the *Enable general branch directives (GJMP, Gbcond)* check box specifies the /G command line option to the RASU8 assembler.

If your assembly source codes includes the Japanese character, you have to select the *Japanese character encode* radio button. Selecting the *EUC* radio button specifies /KE command line option to the RASU8 assembler.

The *Additional options* input field is for adding RASU8 assembler options not supported by IDEU8.

■ **Include debugging information**

Selecting the *Include debugging information* check box specifies the following :

Compiling the source code of a C language : specifies the /SD command line option to the CCU8 compiler and the RASU8 assembler

Assembling the source code of an assembly language : specifies the /D command line option to the RASU8 assembler

■ **Change maximum identifier length**

Selecting the *Change maximum identifier length* check box specifies the /SL command line option to the CCU8 compiler and the RASU8 assembler.

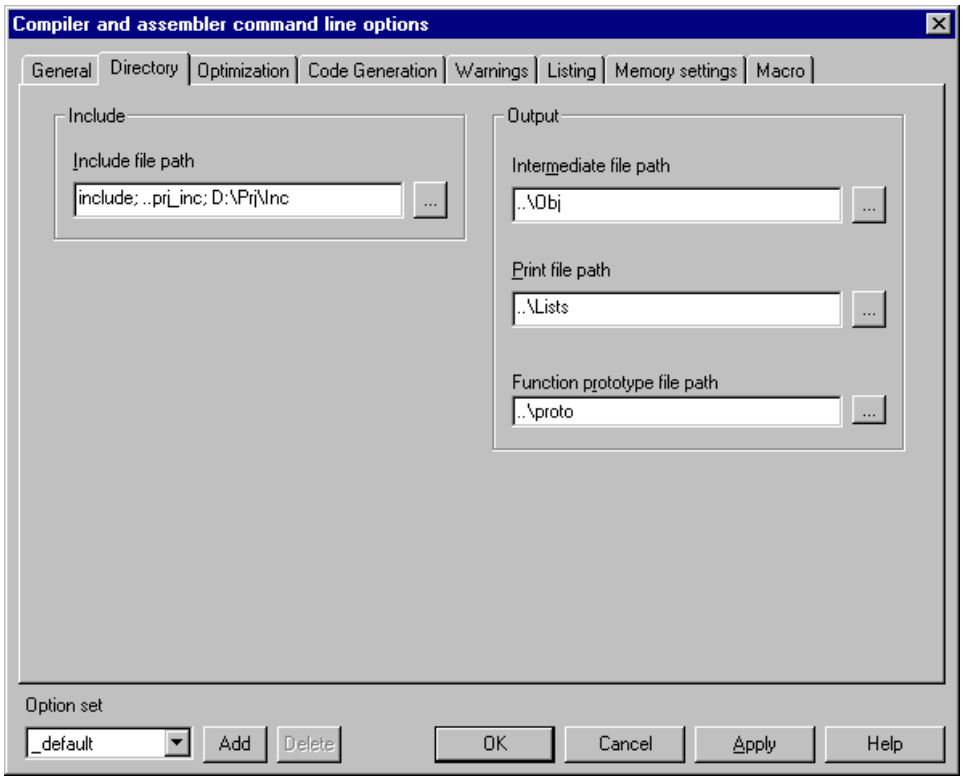
■ **Break on warning message**

Selecting this check box stops the build on the first warning level message from the compiler or assembler.

4.4.2 *Directory* Tab

Clicking the *Directory* tab in the compiler and assembler command line options dialog box switches to the following contents.[8]

[8] In all IDEU8 directory specifications, the directory containing the project (.PID) file of IDEU8 is interpreted as the current directory.



■ Include

This input box specifies the directories to search for include files specified in #include statements in C source code files.[9]

IDEU8 passes the contents as is to the CCU8 compiler and the RASU8 assembler with their /I command line options.

Separate multiple directories with semicolons (;) or commas (,).

■ Output

The Output fields specify the output directories for files generated by the compiler and assembler.

The first input box, *Intermediate file path*, specifies the destination for intermediate files from the CCU8 compiler and relocatable object files from the RASU8 assembler. IDEU8 passes the contents as is to the CCU8 compiler with the /Fa command line option and to the RASU8 assembler with the /O command line option.

The *Print file path* field specifies the output directory for the following files, selected on the Listing tab.

- Error (.LER) file from the compiler
- Call tree (.CAL) file from the compiler
- Preprocessor phase output listing (.I) file from the compiler
- Print (.PRN) file from the assembler

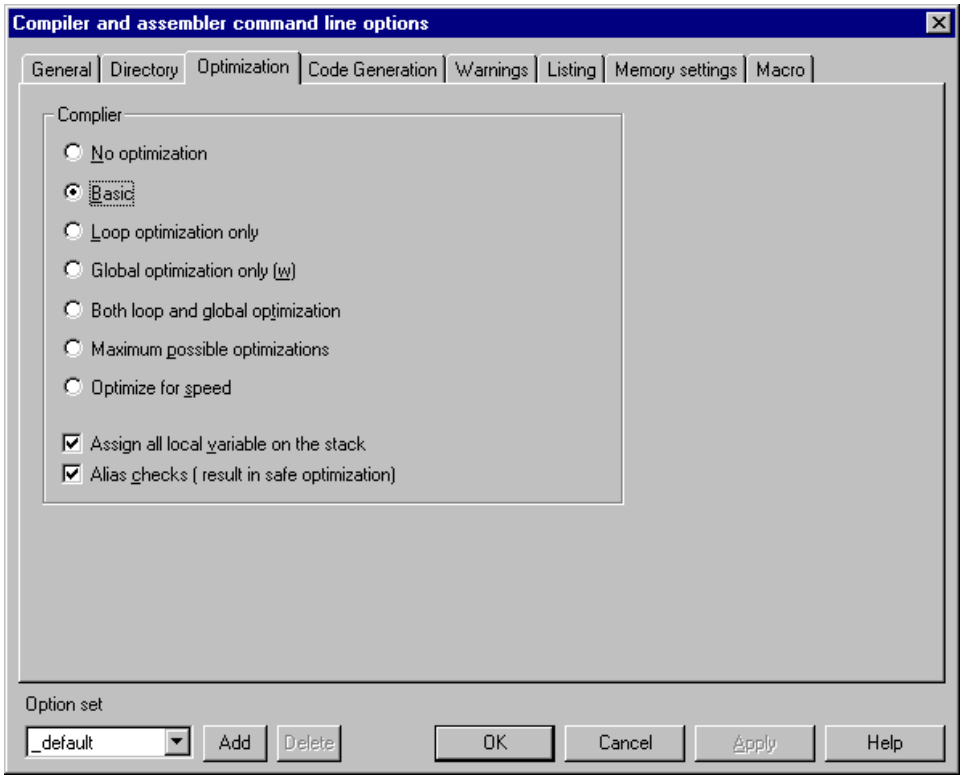
[9] The File menu Environment settings|System path menu command also specifies a path to search for include files for the compiler.

The *Function prototype list path* field specifies the output directory for the following files.

- Function prototype list (.PRO) file from the compiler
- EXTRN declaration (.EXT) file from the assembler

4.4.3 *Optimization* Tab

Clicking the *Optimization* tab in the compiler and assembler command line options dialog box switches to the following contents.



■ **Compiler**

This portion controls the use of the CCU8 compiler command line options for optimization.

Selecting the *No optimization* radio button specifies the /Od command line option to the CCU8 compiler, disabling all optimization.

Selecting the *Basic* radio button specifies no option to the CCU8 compiler for optimization, default optimization.

Selecting the *Loop optimization only* radio button specifies the /Ol command line option to the CCU8 compiler

Selecting the *Global optimizations only* radio button specifies the /Og command line option to the CCU8 compiler.

Selecting the *Both loop and global optimizations* radio button specifies both /Ol and /Og command line options to the CCU8 compiler.

Selecting the *Maximum possible optimizations* radio button specifies the /Om command line option to the CCU8 compiler, applying all optimizations possible.

Selecting the *Optimize for speed* radio button specifies the /Ot command line option to the

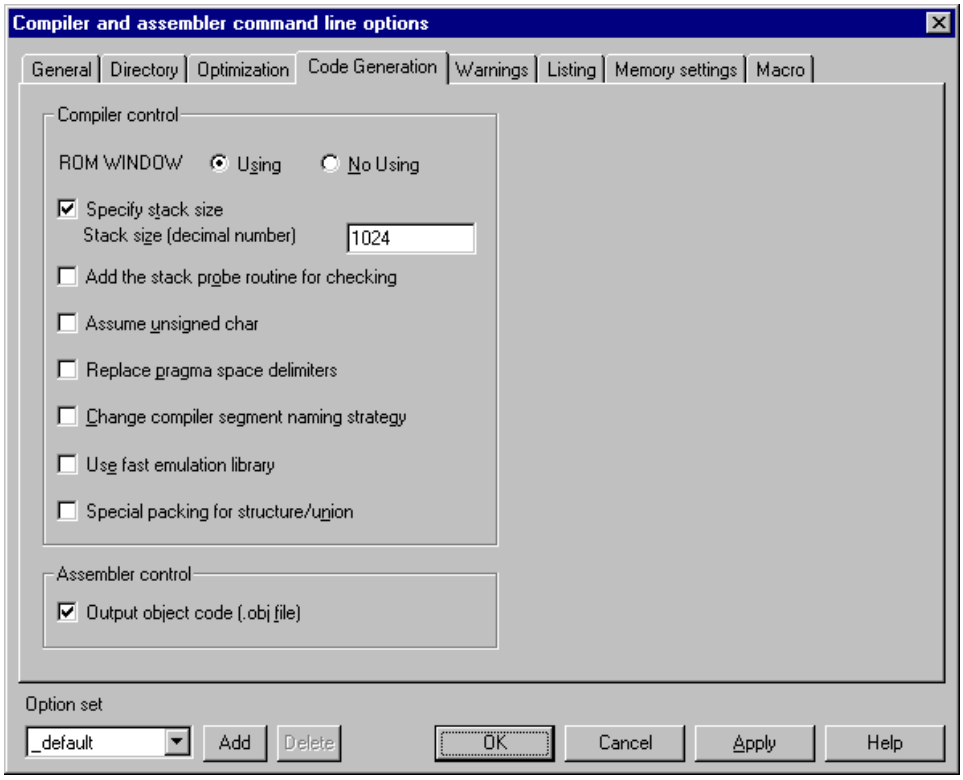
CCU8 compiler.

Selecting the *Allocate all local variables not declared register to the stack* check box specifies the /Zs option to the compiler.

Selecting the *Alias checks (result in safe optimization)* check box specifies the /Oa command line option to the CCU8 compiler.

4.4.4 *Code Generation* Tab

Clicking the *Code Generation* tab in the compiler and assembler command line options dialog box switches to the following contents.



■ **Compiler**

This portion controls the use of the CCU8 compiler command line options for code generation.

Selecting the *No Using* ration button of the *ROM WINDOW* specifies the /NOWIN to the CCU8 compiler, without using ROM WINDOW area.

Selecting the *Specify stack size* check box specifies the /SS command line option to the CCU8 compiler and enables the input box for specifying the stack size.

Selecting the *Add the stack probe routine for checking* check box specifies the /ST command line option to the CCU8 compiler, monitoring the stack overflow.

Selecting the *Assume unsigned char* check box specifies the /J command line option to the CCU8 compiler, using type unsigned char for variables of type char in source code files.

Selecting the *Replace pragma space delimiters with commas* check box specifies the /PF command line option to the CCU8 compiler.

Selecting the *Changing compiler segment naming strategy* check box specifies the /SYS command line option to the CCU8, compiling system files.

Selecting the *Use fast emulation library* check box specifies the /Ff command line option to the CCU8, compiling system files, reducing the execution time for float type.

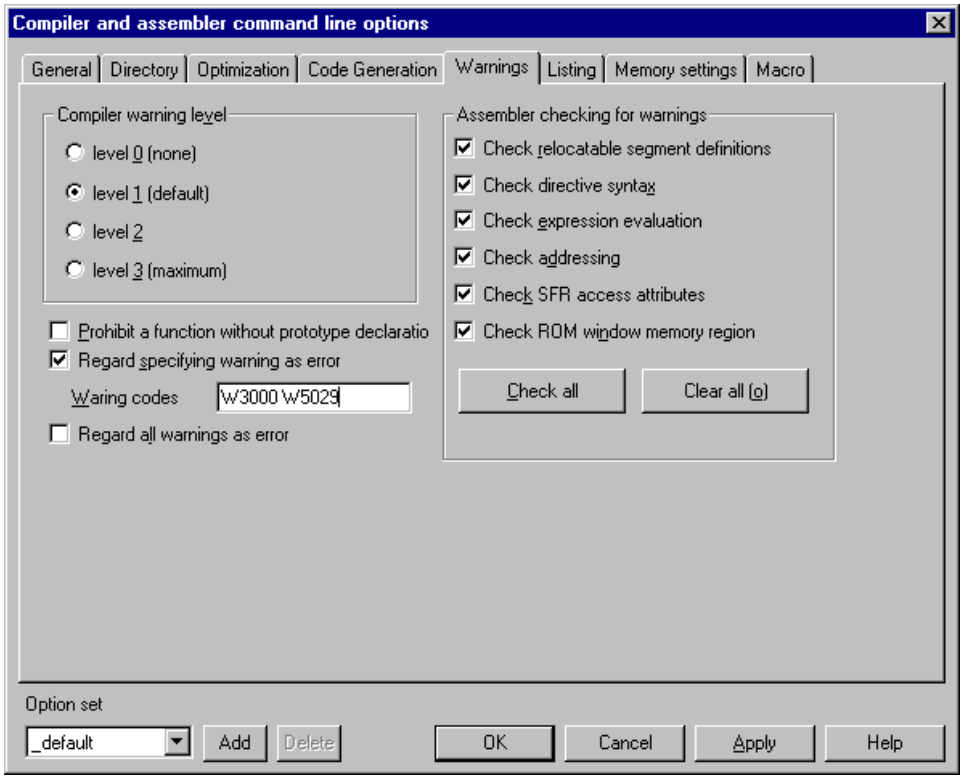
Selecting the *Special packing for structure/union* check box specifies the /Zp command line option to the CCU8.

■ Assembler control

Not selecting the *Output object code (.obj file)* radio button specifies the /NO command line option to the RASU8 assembler, disabling object file output.

4.4.5 *Warnings* Tab

Clicking the *Warnings* tab in the compiler and assembler command line options dialog box switches to the following contents.



■ Compiler warning level

Selecting theses radio button specified with the /W command line option to the CCU8 compiler over the range from 0 to 3.

■ Assembler checking for warnings

These check boxes control RASU8 assembler checking for warnings.

Selecting the *Check relocatable segment definitions* check box specifies the /WR command line option to the RASU8 assembler, enabling checking for problems related to missing ROM window memory region specifications, stack segment size, etc.

Selecting the *Check directive syntax* check box specifies the /WP command line option to the RASU8 assembler, enabling checking of listing control directives, MODEL directives, etc.

Selecting the *Check expression evaluation* check box specifies the /WE command line option to the RASU8 assembler, enabling syntax checking of address expressions, numerical expressions, and operands.

Selecting the *Check addressing* check box specifies the /WA command line option to the RASU8 assembler, enabling checking of usage types, address spaces, and NEAR branch instructions.

Selecting the *Check SFR access attributes* check box specifies the /WS command line option to the RASU8 assembler, enabling checking of SFR access restrictions.

Selecting the *Check ROM window memory region* check box specifies the /WT command line option to the RASU8 assembler, enabling checking for problems arising from missing ROM window memory region specifications.

■ Prohibit a function without prototype declaration

Selecting this check box specifies the /Wc W5031 option to the compiler. It instructs the compiler to issue error against a function call without its prototype declaration.

■ Regard specifying warning as error

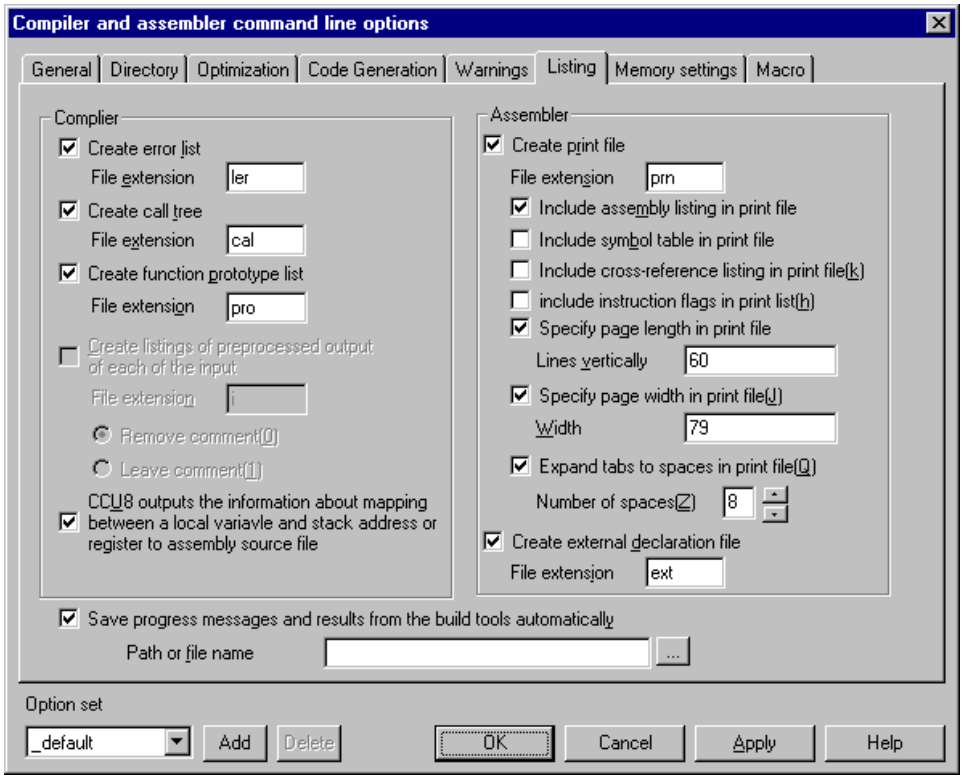
Selecting this check box specifies the /Wc option to the compiler along with the warning numbers specified in the accompanying *Warning codes* input box. Separate multiple warning numbers with commas or spaces.

■ Regard all warnings as error

Selecting this check box specifies the /Wa option to the compiler and grays out the warning level 0 radio button, *Regard specifying warning as error* check box, and *Prohibit a function without prototype declaration* check box.

4.4.6 *Listing* Tab

Clicking the *Listing* tab in the compiler and assembler command line options dialog box switches to the following contents.



■ **Compiler**

This portion controls the use of the CCU8 compiler command line options for listing files.

Selecting the *Create error list* check box specifies the /LE command line option to the CCU8 compiler, generating an error list file. Specify the file extension for this file in the accompanying input box.[10]

Selecting the *Create call tree* check box specifies the /CT command line option to the CCU8 compiler, generating a call tree file. Specify the file extension for this file in the accompanying input box.[11]

Selecting the *Create function prototype list* check box specifies the /Zg command line option to the CCU8 compiler, generating a prototype list file. Specify the file extension for this file in the accompanying input box.[12]

Checking the *Create listins of preprocessed output of each of the input* and selecting the *Remove comment* radio button specifies the /PC command line option to the CCU8 compiler, stripping comments from these results;

Checking the *Create listins of preprocessed output of each of the input* and selecting the *Leave comment* radio button specifies the /LP command line option to the CCU8 compiler, stripping comments from these results;

[10] The default extension for an error file for CCU8 is .LER.

[11] The default extension for a call tree file for IDEU8 is .CAL.

[12] The default extension for a function prototype list file for IDEU8 is .PRO.

Note that selecting a check box for any of the first three outputs (error list, call tree, and function prototype list) grays out the fourth (Save Preprocessor phase output listing and skip compile). By the same token, selecting the fourth check box grays out the first three.

Checking the *CCU8 outputs the information about mapping between a local variable and stack address or register to assembly source file* check box specifies the /Lv command line option to the CCU8 compiler.

■ Assembler

This portion controls the use of the RASU8 assembler command line options for listing files.

Selecting the *Create print file* check box specifies the /PR command line option to the RASU8 assembler.

Selecting the *Include assembly listing in print file* check box specifies the /L command line option to the RASU8 assembler. This check box is only valid when the *Create print file* check box is selected.

Selecting the *Include symbol table in print file* check box specifies the /S command line option to the RASU8 assembler. This check box is only valid when the *Create print file* check box is selected.

Selecting the *Include cross-reference listing in print file* check box specifies the /R command line option to the RASU8 assembler. This check box is only valid when the *Create print file* check box is selected.

Selecting the *Include flag changes in print file* check box specifies the /SF command line option to the RASU8 assembler. This check box is only valid when the *Create print file* check box is selected.

Selecting the *Specify page length in print file* check box sends the /PL command line option to the RASU8 assembler together with the specified number. This check box is only valid when the *Create print file* check box is selected.

Selecting the *Specify page width in print file* check box sends the /PW command line option to the RASU8 assembler together with the specified number. This check box is only valid when the *Create print file* check box is selected.

Selecting the *Expand tabs to spaces* check box sends the /T command line option to the RASU8 assembler together with the specified number. This check box is only valid when the *Create print file* check box is selected.

Selecting the *Create assembly error file* check box specifies the /E command line option to the RASU8 assembler.

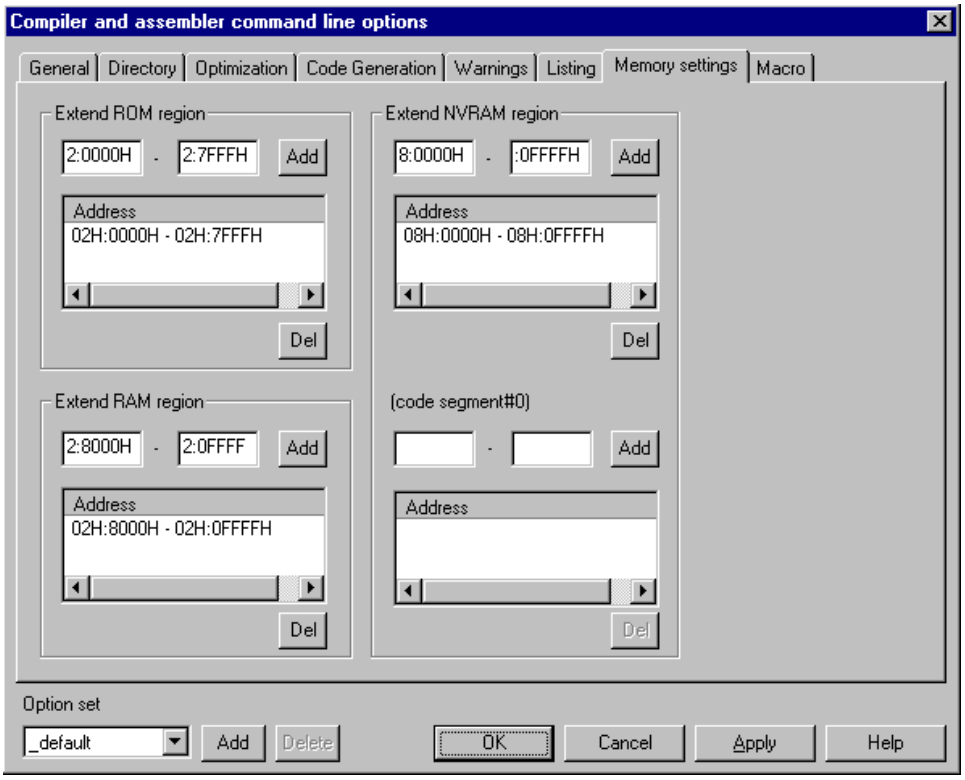
Selecting the *Create external declaration file* check box specifies the /X command line option to the RASU8 assembler.

■ Save progress messages and results from the build tools automatically

Selecting this check box saves a separate build log for each source code file translated by the CCU8 compiler or the RASU8 assembler in the output directory specified in the accompanying *Path or file name* input box. Leaving the input box empty saves these files in the same directory as the project (.PID) file. The file names are the base name for the source code file plus the extension .LOG.

4.4.7 *Memory settings* Tab

Clicking the *Memory settings* tab in the compiler and assembler command line options dialog box switches to the following contents.



These command line options specify external memory regions on a user system.

Each portion allows adding plural region.

■ **Extend ROM region**

Adding the *Extend ROM region* specifies the /BROM command line option to the RASU8 assembler, for adding external ROM region.

■ **Extend RAM region**

Adding the *Extend RAM region* specifies the /BRAM command line option to the RASU8 assembler, for adding external RAM region.

■ **Extend NVRAM region**

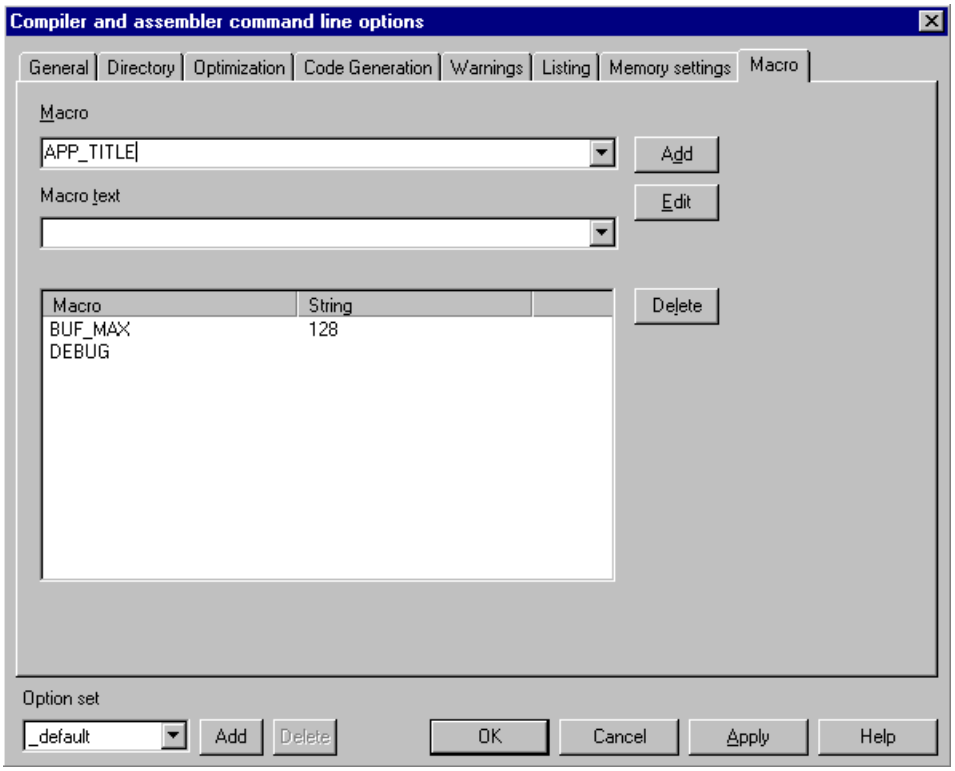
Adding the *Extend NVRAM region* specifies the /BNVRAM command line option to the RASU8 assembler, for adding external nonvolatile memory region.

■ **Extend NVRAM region (code segment #0)**

Adding the *Extend NVRAM region (code segment#0)* specifies the /BNVRAMP command line option to the RASU8 assembler, for adding external nonvolatile memory region to physical segment address #0 in the code memory space.

4.4.8 *Macro* Tab

Clicking the *Macro* tab in the compiler and assembler command line options dialog box switches to the following contents.



■ **Macro**

This input box specifies the macro symbol for the /D command line option of a CCU8 compiler and the /DEF command line option of a RASU8 assembler.

■ **Macro Text**

This input box specifies the character string to replace occurrences of the specified macro symbol in the source code files.

■ **Add Button**

Entering text for the macro symbol and body and pressing this button adds the macro definition to the list box.

■ **Delete Button**

Selecting a macro definition from the list box and pressing this button deletes that macro definition from the list box.

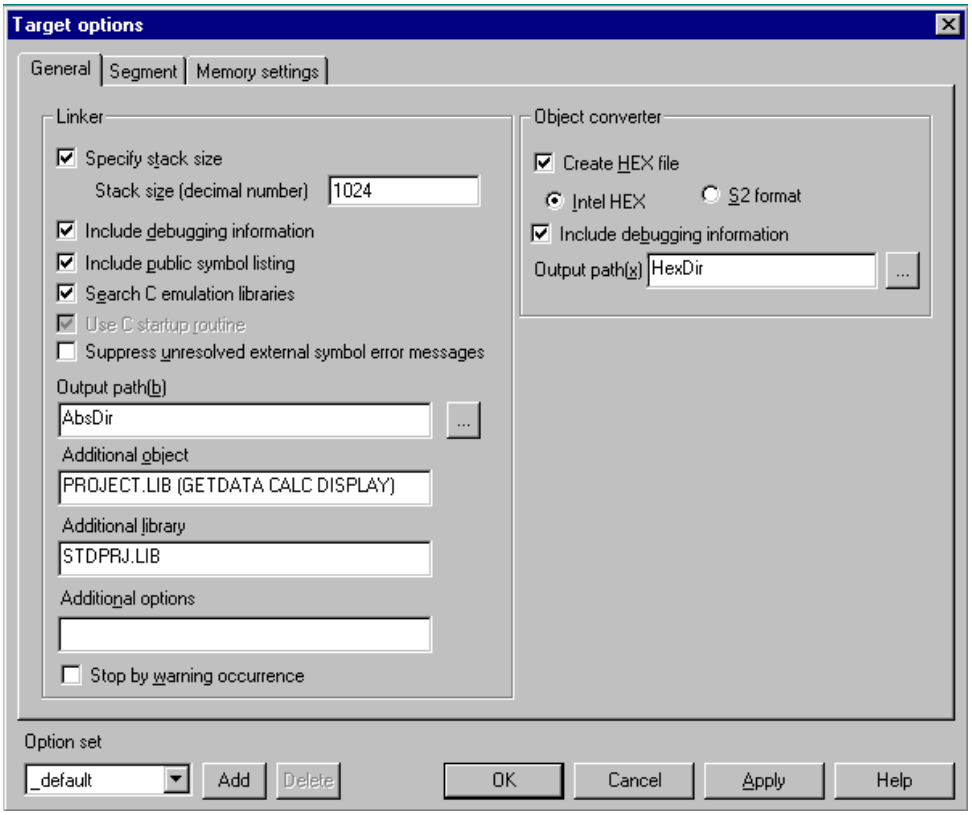
4.5 Specifying Target Options

The target command line options are the build tool options affecting the creation of the final target (ABS) file for the program by the linker and object converter. To specify them, choose the *Project* menu *Options* → *Target*... menu command sequence.

The target option dialog box has four tabs: *General*, *Memory*, *Segments*, and *List*.

4.5.1 *General* Tab

Clicking the *General* tab in the target option dialog box switches to the following contents.



■ **Linker**

This portion controls the use of RLU8 linker command line options.

Selecting the *Specify stack size* check box specifies the /STACK command line option to the RLU8 linker , for changing the stack segment size.

Selecting the *Include debugging information* check box specifies the /D and /SD command line option to the RLU8 linker, for debugging using DTU8 debugger.

Selecting the *Include public symbol listing* check box specifies the /S command line option to the RLU8 linker, for listing all public symbols.

Selecting the *Search C emulation libraries* check box specifies the /CC command line option to the RLU8 linker.

Deselecting this check box grays out the *Use C start-up routine* check box, making it necessary to add the compiler command line options /COMB($$init_info $$init_info_end) /COMB($$content_of_init $$end_of_init).

Selecting the *Suppress unresolved external symbol error messages* check box specifies the /EXC command line option to the RLU8 linker, suppressing such messages if the unresolved external symbol is unused.

The *Output path* input box specifies the destination for an absolute object files (.abs) from the RLU8 linker.

The text in the *Additional object* field goes as is into the RLU8 linker object_files field.

This field is for specifying object (.OBJ) files that cannot be added to the Project window--by absolute path, for example, or because they are in library files.

The text in the *Additional library* field input field goes as is into the RLU8 linker libraries field. This field is for specifying library files that cannot be added to the Project window--by system library search path or absolute path, for example.

The *Additional options* input field is for adding RLU8 linker options not supported by IDEU8.

Selecting the *Stop by warning occurrence* check box stops the link process on the first warning level message from the linker. Here the Project menu Debug menu command does not load the debugger.

■ Object Converter

This portion controls the use of OHU8 object converter command line options.

Selecting the *Create HEX file* check box automatically runs the OHU8 object converter after the linker.

Selecting the *Intel HEX* radio button specifies the /H command line option to the OHU8 object converter, outputting the Intel HEX output format.

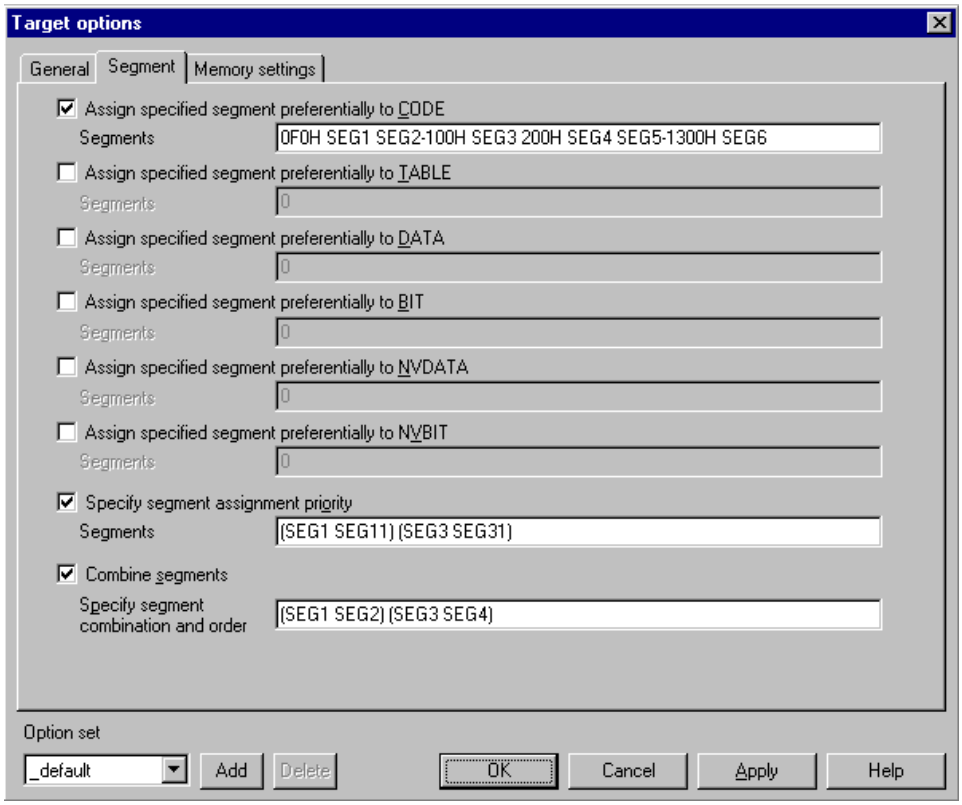
Selecting the *S format* radio button specifies the /S command line option to the OHU8 object converter, outputting the Motorola S2 output format.

Selecting the *Include debugging information* check box specifies the /D command line option to the OHU8 object converter, adding symbol tables to the output.

The *Output path* input box specifies the destination for a Hex files from the OHU8 object converter.

4.5.2 *Segment* Tab

Clicking the *Segment* tab in the target option dialog box switches to the following contents.



■ Segment Assignment

Selecting the *Assign specified segment preferentially to CODE* check box sends the /CODE command line option to the RLU8 linker together with the segment names and byte addresses specified in the accompanying input box.

Selecting the *Assign specified segment preferentially to TABLE* check box sends the /TABLE command line option to the RLU8 linker together with the segment names and byte addresses specified in the accompanying input box.

Selecting the *Assign specified segment preferentially to DATA* check box sends the /DATA command line option to the RLU8 linker together with the segment names and byte addresses specified in the accompanying input box.

Selecting the *Assign specified segment preferentially to BIT* check box sends the /BIT command line option to the RLU8 linker together with the segment names and bit addresses specified in the accompanying input box.

Selecting the *Assign specified segment preferentially to NVDATA* check box sends the /NVDATA command line option to the RLU8 linker together with the segment names and byte addresses specified in the accompanying input box.

Selecting the *Assign specified segment preferentially to NVBIT* check box sends the /NVBIT command line option to the RLU8 linker together with the segment names and bit addresses specified in the accompanying input box.

Selecting the *Specify segment assignment priority* check box sends the /ORDER command line option to the RLU8 linker together with the segment names specified in the accompanying input box, specifying the segment processing order.

Selecting the *Combine segments* check box sends the /COMB command line option to the RLU8 linker together with the segment names specified in the accompanying input box, specifying the memory space assignment order. Note that segments must be of type CODE

or TABLE. This specification for the input box is allowed the plural combinations using parentheses.

For further details on parameter syntax for specifying segment combination and order fields, refer to the MACU8 Assembler Package User's Manual.

Too much information to be input to the segment combination and order fields makes it harder to confirm the settings in the dialog box. In that case, using a response file for the RLU8 linker is recommended. As an example, create a response file as shown below. Here, a file named "sample.res" is used.

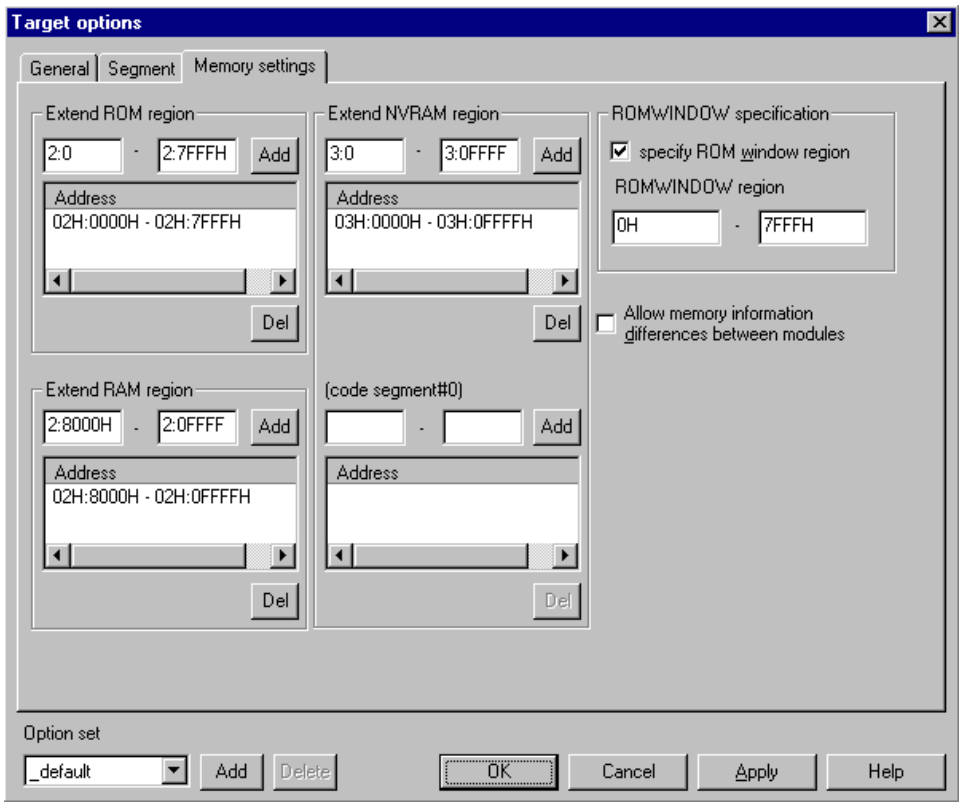
```
//Response file (sample.res)
/CODE(0F0H SEG1 SEG2-100H SEG3 200H SEG4 SEG5-1300H SEG6)
/ORDER(SEG1 SEG11)
/ORDER(SEG3 SEG31)
/COMB(SEG1 SEG2)
/COMB(SEG3 SEG4)
```

Place the created file into the same folder as the project file and specify the file in the *Additional options* field on the *General* tab in the target option dialog box by prefixing '@' to the response file name, as shown below.

Specifying this way will pass an option defined in a response file to the RLU8 linker.

4.5.3 *Memory settings* Tab

Clicking the *Memory settings* tab in the target option dialog box switches to the following contents.



These command line options specify external memory regions on a user system. Each portion is allowed to add plural region.

■ Extend ROM region

Adding the Extend ROM region specifies the /ROM command line option to the RLU8 linker, for adding external ROM region.

■ Extend RAM region

Adding the Extend RAM region specifies the /RAM command line option to the RLU8 linker, for adding external RAM region.

■ Extend NVRAM region

Adding the Extend NVRAM region specifies the /NVRAM command line option to the RLU8 linker, for adding external nonvolatile memory region.

■ Extend NVRAM region (code segment #0)

Adding the Extend NVRAM region (code segment#0) specifies the /NVRAMP command line option to the RLU8 linker, for adding external nonvolatile memory region to physical segment address #0 in the code memory space.

■ ROM WINDOW specification

This portion specifies the ROM WINDOW region to the RLU8 linker.

Selecting the *Specify ROM window region* check box sends the /ROMWIN command line option to the RLU8 linker together with the specifications from the list box.

■ **Allow memory information differences between modules**

Selecting this check box specifies the /PDIF command line option to the RLUE8 linker.

4.6 Defining Option Sets

IDEU8 allows the programmer to specify multiple sets of option settings, select a file in the *Project* window, and specify an option set for that file with the context menu *Settings*... menu command.

Note, however, that the *Project* window actually differentiates two types of option sets: compiler and assembler command line options for specification at the all of project or the individual source code file level and target options for specification at the root level, the project name.

■ **Example 1** ■

Using separate option sets for C and assembly language source code files

Use the *Compile/assemble options* dialog box to create two option sets: “C settings” for C source code and “ASM settings” for assembly language source code. For each C source code file in the project, specify the former; for each assembly language source code file.

■ **Example 2** ■

Using separate option sets for debugging and for release

Use the *Compile/assemble options* dialog box to create two option sets: “Debugging settings” invoking macro definitions generating debugging code and turning optimization off and “Release settings” using settings normal for shipping versions. Use the former for work in progress and switch source code files to the latter as you finish debugging them.

5. Building Programs

5.1 Overview

IDEU8 first invokes the CCU8 compiler to convert any C source code files in the project to intermediate files containing assembly language source code. It then invokes the RASU8 assembler to convert these intermediate files and any assembly language source code files in the project to relocatable object files. Finally, it invokes the RLU8 linker to merge these relocatable object files into the final target (ABS) file for the program. Also available is the object converter for converting to the Intel hex or Motorola S2 format, if a hex format file is necessary.

Note that for a normal build operation, IDEU8 skips regeneration if the time stamp on an object file indicates that it is newer than the corresponding source code files and any files included therein.

5.2 Build

The *Build* menu command starts by determining whether each object file going into the final target (ABS) file for the program needs updating. If the time stamp for a source code or include file necessary for constructing the intermediate object file is newer than that of the object file, IDEU8 performs a build operation on that source code file. Reconstructing an intermediate object file leads to reconstructing the final target (ABS) file for the program. This approach minimizes the number of build tool invocations.

IDEU8 updates its database of the source code files and include files going into each object file during the above compile and assemble operations.

5.3 Rebuild

The *Rebuild* menu command starts by deleting any object files corresponding to source code files in the project. It then invokes the compiler and assembler to rebuild all object files from scratch. This action forces reprocessing of all source code files.

5.4 Build Selected File

The Project menu Build selected file name menu command compiles and assembles only the file specified in the Project window. It does not check the time stamp.

5.5 Canceling Build Commands

Choosing the *Project* menu *Cancel build* menu command cancels the current *Build* or

Rebuild menu command.[13]

5.6 Deleting Object Files

To delete the final target (ABS) file together with all intermediate object files generated, choose the *Project* menu *Clean* menu command. The next *Build* menu command thus becomes identical to the *Rebuild* menu command, applying build operations to all source code files.

5.7 Loading DTU8 Debugger

The *Project* menu *Debug* menu command updates the final target file and loads it into the DTU8 debugger.

For further details on using the DTU8 debugger, refer to the DTU8 Debugger User's Manual.

5.7.1 Modifying Program

To modify the program while using the DTU8 debugger, edit the corresponding source code files in the IDEU8 *Editor* window, update the object file with the IDEU8, and continue debugging with the DTU8 debugger.

5.7.2 Link Between IDEU8 and Debugger

Switching project files in IDEU8 or opening a different program file in the debugger breaks the link between IDEU8 and the debugger. Either exit the debugger and reload it from IDEU8 or switch it to the absolute object file generated by the currently open IDEU8 project.

5.8 *Output* Window

The *Output* window displays progress messages and results from the compiler, assembler, and linker.

The *View* menu *Output* menu command is a toggle, alternately opening and closing this window.

5.8.1 Copying Messages to Windows Clipboard

To copy the currently selected text to the Windows clipboard choose the *Output* window context menu *Copy* menu command.

5.8.2 Opening the corresponding file

Moving the cursor to a compiler or assembler error message in the *Output* window and choosing the context menu *Jump to error line* menu command—or simply double-click on an error or warning message in this window—or move the cursor there and press the Enter key—opens the corresponding file in the *Editor* window and automatically scrolls to the

[13] The *Cancel build* menu command simply cancels the next calling of each build tool (CCU8, RASU8, RLU8, OHU8): processing by each tool is completed.

source code line triggering the error.

5.8.3 Saving Messages

To save the contents of the *Output* window to a log file choose the *Output* window context menu *Save messages*... menu command.

5.8.4 Clearing Messages from Window

To delete all text and clear the *Output* window, choose the context menu *Clear* menu command.

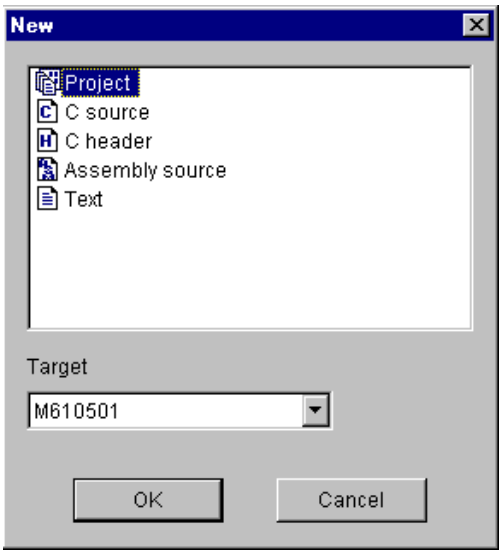
5.8.5 Dock

The context menu *Docking view* menu command toggles docking of the Output window at the bottom of the IDEU8 main window.

6. Other Functions

6.1 Creating Text Files

To create a C source code file, header file, or other text file in the IDEU8 environment, choose the *File* menu *New*... menu command to display the following dialog box.



■ New C Source Code File

Double-clicking “C source” from the list in the *New* dialog box creates an empty C source code file in the *Editor* window.

■ New C Header File

Double-clicking “C header” from the list in the *New* dialog box creates an empty C header file in the *Editor* window.

■ New Assembly Language Source Code File

Double-clicking “assembly source” from the list in the *New* dialog box creates an empty assembly language source code file in the *Editor* window.

■ New Text File

Double-clicking “text” from the list in the *New* dialog box creates an empty text file in the *Editor* window.

Note that the window type selected with this dialog box is the same as that specified with the File menu *New*... menu command.

6.2 *Editor* Window

The *Editor* window is for viewing and editing source code files and header files. The following methods are available for opening a source code file in the *Editor* window to edit the program.

■ Opening from the menu

Choose the *File* menu *Open*... menu command, specify the file type in the dialog box that appears, select a file from the resulting list, and press the *Open* button.

■ Opening from the *Project* window

Select the desired file from the treelike hierarchy in the *Project* window and choose the context menu *Open*... menu command.

■ Opening from the *Output* window

If there is a compile or other error during a build operation on a source code file, an error message and the error location appear in the *Output* window. Moving the cursor to a compiler or assembler error message in the *Output* window and choosing the context menu *Jump to error line* menu command—or pressing the Enter key—opens the corresponding file in the *Editor* window and automatically scrolls to the source code line triggering the error.

6.2.1 *Edit* Menu

The *Edit* menu provides menu commands for editing the file in the *Editor* window.

■ *Undo*

Choosing the *Edit* menu *Undo* menu command reverses the effects of the last text edit operation.

■ *Cut*

This menu command moves the currently selected text to the Windows clipboard.

■ *Copy*

This menu command copies the currently selected text to the Windows clipboard.

■ *Paste*

This menu command inserts the contents of the Windows clipboard at the current cursor position.

■ *Delete*

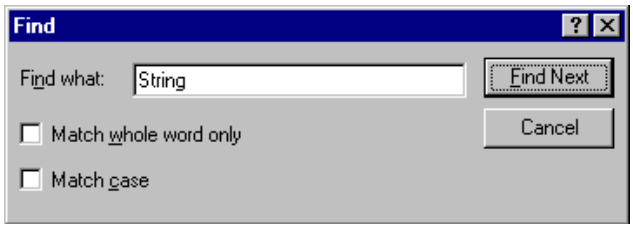
This menu command deletes the currently selected text. Unlike the *Cut* menu command, it does not copy the text to the Windows clipboard.

■ *Select all*

This menu command selects all text in the file.

■ ***Find...***

This menu command displays the following dialog box.



Enter the desired text in the *Text to find* input box.

Selecting the *Words only* check box limits the text search to complete words—skipping "romwindow" and "from" if the text search text is "rom," for example. Selecting the *Case sensitive* check box expands the text search to distinguish between upper and lower case—skipping "ABC" and "Abc" if the text search text is "abc," for example.

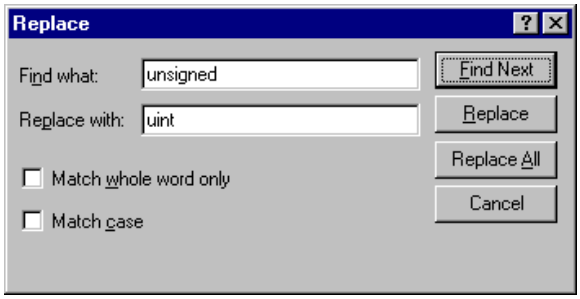
Pressing the *Find next* button repeats the text search using the current dialog box search condition specifications, finding the next occurrence—if one exists.

■ ***Find Next***

This menu command continues the text search using the search condition specifications from the last *Find*... menu command.

■ ***Replace...***

This menu command displays the following dialog box.



Enter the search text in the *Text to find* input box and the replacement text in the *Replacement* input box.

Selecting the *Case sensitive* check box expands the text search to distinguish between upper and lower case—skipping "ABC" and "Abc" if the text search text is "abc," for example.

Pressing the *Find next* button repeats the text search using the current dialog box search condition specifications, finding the next occurrence—if one exists.

Pressing the *Replace next* button replaces the found text and searches for the next occurrence.

Pressing the *Replace all* button replaces all occurrences without any prompting.

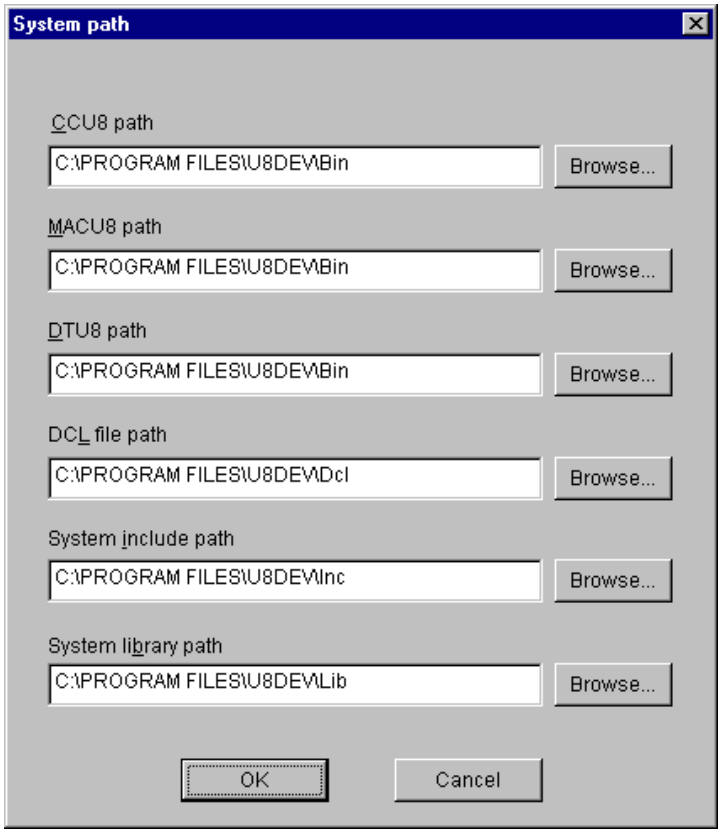
■ ***Jump* to Line Number...**

This menu command displays the dialog box for moving the cursor to the specified line in the file.

6.3 Specifying System Paths

Choosing the *File* menu *Environment settings* → *System Path...* menu command sequence displays the following dialog box for specifying build tool paths.[14]

Note: The nX-U8 development tool setup program automatically specifies these system path settings, but they may be modified as necessary.



■ Specifying CCU8 Compiler path

Enter the directory holding the executable for the CCU8 compiler in the *CCU8 path* input box.

■ Specifying MACU8 Assembler Package path

Enter the directory holding the executable for the MACU8 assembler package in the *MACU8 path* input box.

■ Specifying DTU8 Debugger Path

Enter the directory holding the executable for the DTU8 debugger in the *DTU8 path* input box.

■ Specifying DCL file path

This path is specified to the DCL environment variable, invoking the RASU8 assembler.

---

[14] If IDEU8 was installed using the nX-U8 Development Tools setup program, the installer has specified the system paths. Change them or add new paths as required.

■ **Specifying System include path**

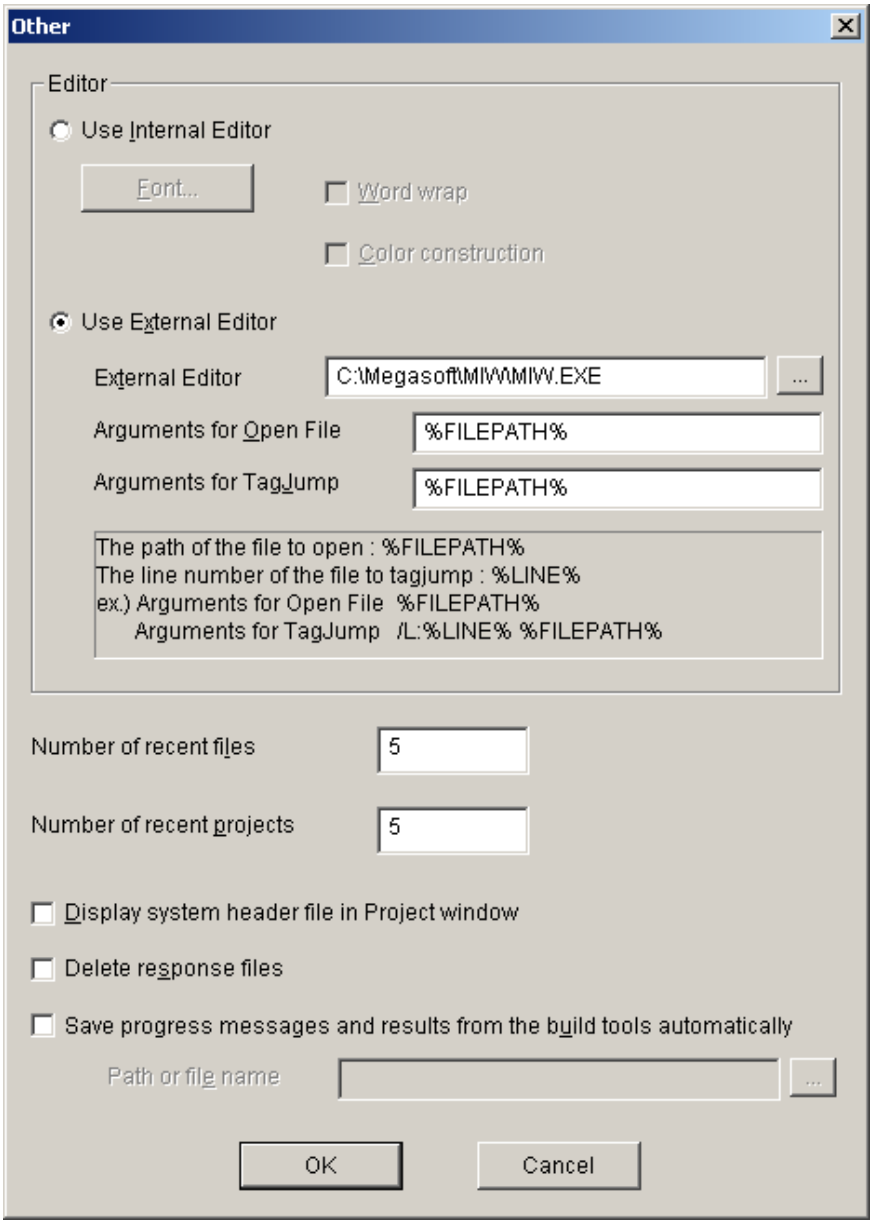
This path is specified to the INCLU8 environment variable, invoking the CCU8 compiler. Separate multiple directories with semicolons (;) or commas (,).

■ **Specifying System library path**

This path is specified to the LIBU8 environment variable, invoking the RLU8 linker. Separate multiple directories with semicolons (;) or commas (,).

6.4 Other Environment Settings

Choosing the *File* menu *Environment settings* → *Other*... menu command sequence displays the following dialog box



The *Editor* field specifies whether to *Use Internal Editor* or *Use External Editor*.

Specifying *Use Internal Editor* enables the following items:

■ ***Font*... button**

Pressing this button displays the Font dialog box for specifying the display font in the *Editor w*indow.

■ ***Word wrap* check box**

Selecting this check box specifies to wrap text to the editor window size.

■ ***Color construction* check box**

Selecting this check box specifies to highlight keywords for C and assembly language in an editor window.[15]

Specifying *Use External Editor* enables the following items:

■ ***External Editor entry* field**

Enter the file path to the external editor in this filed.

■ ***Arguments for Open File* entry field**

Enter in this field the command line option string (file path and arbitrary option) for opening a file of the registered external editor.

Specify a %FILEPATH% keyword (all in capital letters) for the open target file path in the command line option. If any blank character is included (like *Program Files*) in the file path, enclose %FILEPATH% by double quotation marks (") to specify it.

To open a source file or header file by IDEU8, replace %FILEPATH% with the file path that has been specified for open and then activate the external editor.

■ ***Arguments for TagJump* entry field**

Enter in this field the option string to be passed to the external editor (file path, line number, and arbitrary option) when making a tag jump (opening the specified file to jump to the specified line) using the registered external editor.

Specify a %FILEPATH% keyword (all in capital letters) for the open target file path in the command line option and %LINE% keyword (all in capital letters) for the jump destination line number. If any blank character is included (like *Program Files*) in the file path, enclose %FILEPATH% by double quotation marks (") to specify it.

To make a tag jump by IDEU8, replace %FILEPATH% with the tag jump target file path and %LINE% with the jump destination line number and then activate the external editor.

Enter the *Number of recent files* input box for the number of history for choosing the *File* menu *Files* menu command sequence.

Enter the *Number of recent projects* input box for the number of history for choosing the *File* menu *Projects* menu command sequence.

[15] Syntax highlighting can adversely affect editor response for large files. Disable it as necessary or as desired.

Selecting the *Display system header file in Project window* specifies to display system header files existing the *System include path* in the *Include* folder of the *Project* window.

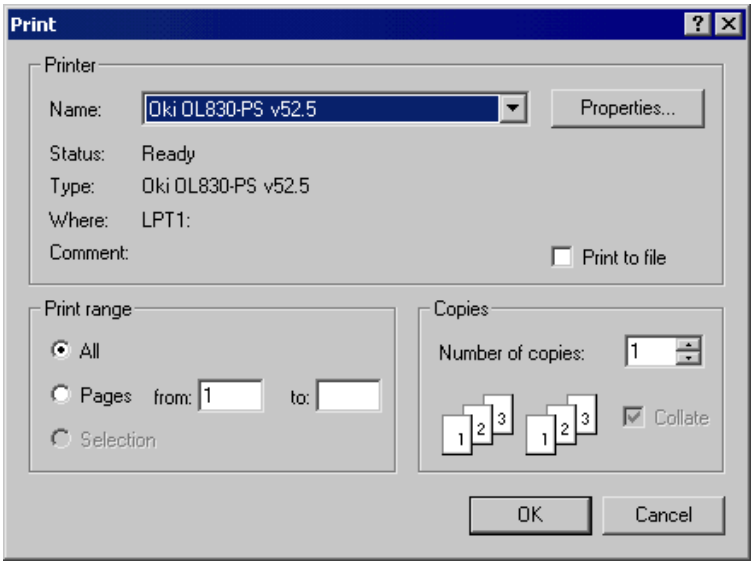
Selecting the *Delete response files* specifies to delete temporary files created by IDEU8 for building process.

Selecting the *Save progress messages and results from the build tools automatically* check box saves the Output window contents to a build log (text) for each build processing. Note that subsequent builds overwrite the build log file.

Specifying a directory path name in the *Path or file name* input box saves the build log file as the file name formed from the base name for the project file plus the extension .LOG into the directory path. Specifying a file name saves the log file as that name.

6.5 Printing Files

Choosing the *File* menu *Print...* menu command displays the following dialog box for printing the source code file currently displayed in the *Editor* window.

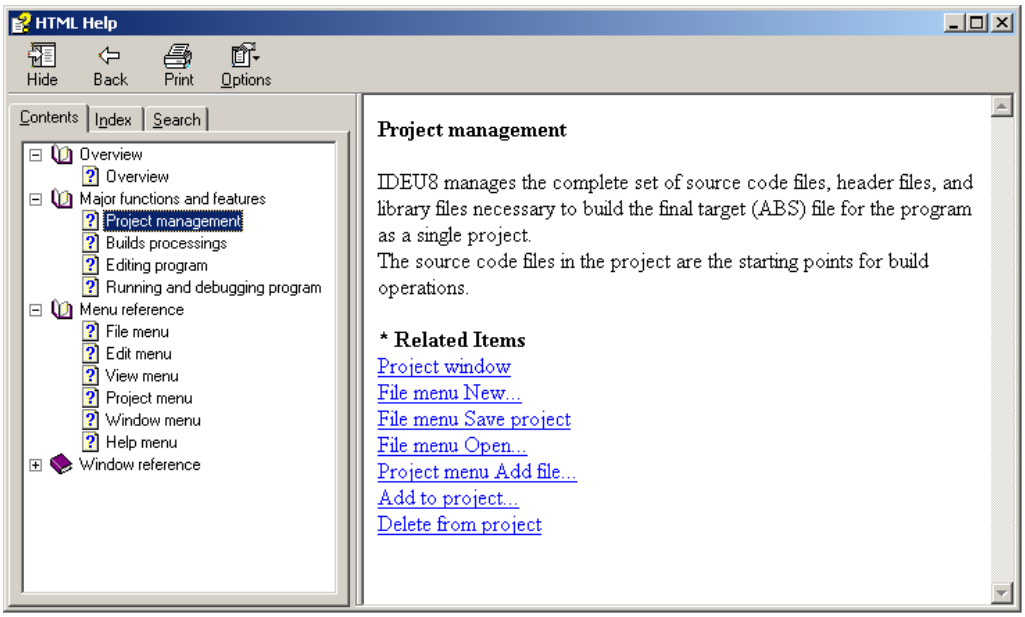


Specify such settings as the page range and number of copies and press the *OK* button.

6.6 On-Line Help

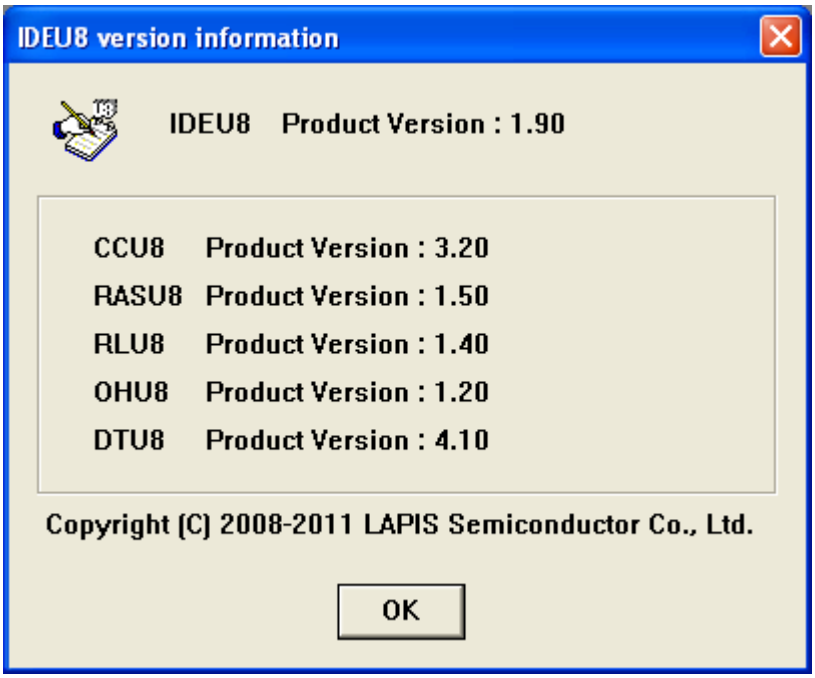
IDEU8 provides on-line help in HTML help format.

Choosing the *Help* menu *Help contents* menu command or pressing the *F1* key displays the IDEU8 *Help* window.



6.7 Displaying Version Information

Choosing the *Help* menu *About IDEU8*... menu command displays the following dialog box, which gives version numbers for IDEU8 and the build tools.



Index

A

assembler option 16

B

build 34
cancelling ~ 34
Build 7
build selected file 34

C

CCU8 compiler
/CT command line option 24
/D command line option 27
/Fa command line option 19
/far command line option 18
/Ff command line option 22
/I command line option 19
/J command line option 21
/LE command line option 24
/ML command line option 17
/MS command line option 17
/near command line option 17
/NOWIN command line option 21
/Oa command line option 21
/Od command line option 20
/Og command line option 20
/Ol command line option 20
/Om command line option 21
/Ot command line option 21
/PC command line option 24
/PF command line option 22
/SD command line option 18
/SL command line option 18
/SS command line option 21
/ST command line option 21
/SYS command line option 22
/T command line option 17
/W command line option 22
/Wa command line option 23
/Wc command line option 23
/Wc W5031 command line option 23
/Za command line option 18
/Zg command line option 24
/Zp command line option 22
LP command line option 24
Lv command line option 25
Clean 35
clipboard 35, 38
code memory space 17
command line option
specifying ~ 7
compiler option 16

D

DCL environment variable 41
Debug 8

E

Editor window 35, 38

F

file
deleting ~ 15
Edit menu 38
File menu
Environment settings 40, 41
Files 43
New... ..... 5, 10, 37
Open... 11
Print... 44
Projects 43
Save project 10
Font 42

H

Help menu
About IDEU8... 45
Help contents 44

I

INCLU8 environment variable 41
installing ~ 4
intermediate object file 16

L

LIBU8 environment variable 41
Listing Tab
Compile/assemble 24

M

Macro Tab
Compile/assemble 27

N

notation 3

O

object file
deleting ~ 35
OHU8 object converter
/D command line option 29
/H command line option 29
/S command line option 29
on-line help 44
optimization
for speed 21
option set 33
Options 7
Compile/assemble 16
Output window 35

P

program
modifying ~ 35
Project
creating ~ 10
creating new ~ 5
reading in ~ 11
saving ~ 10
Project menu
Add file 10
Cancel build 34
Debug 9, 35
Project window 11
Include folder 43
Project Window
Context Menu 12

R

RASU8 assembler
/BNVRAM command line option 26
/BNVRAMP command line option 26
/BRAM command line option 26
/BROM command line option 26
/CD command line option 18
/D command line option 18
/DEF command line option 27
/DF command line option 18
/DN command line option 17
/E command line option 25
/G command line option 18
/I command line option 19
/KE command line option 18
/L command line option 25
/ML command line option 17
/MS command line option 17
/NO command line option 22
/O command line option 19
/PL command line option 25

/PR command line option 25
/PW command line option 25
/R command line option 25
/S command line option 25
/SD command line option 18
/SF command line option 25
/SL command line option 18
/T command line option 25
/WA command line option 23
/WE command line option 23
/WP command line option 23
/WR command line option 22
/WS command line option 23
/WT command line option 23
/X command line option 25
rebuild 34
response 43
RLU8 linker
/STACK command line option 28
RLU8 linker
/D command line option 28
RLU8 linker
/SD command line option 28
RLU8 linker
/S command line option 28
RLU8 linker
/CC command line option 28
RLU8 linker
/EXC command line option 28
RLU8 linker
/CODE command line option 30
RLU8 linker
/TABLE command line option 30
RLU8 linker
/DATA command line option 30
RLU8 linker
/BIT command line option 30
RLU8 linker
/NVDATA command line option 30
RLU8 linker
/NVBIT command line option 30
RLU8 linker
/ORDER command line option 30
RLU8 linker
/COMB command line option 30
RLU8 linker
/ROM command line option 32
RLU8 linker
/RAM command line option 32
RLU8 linker
/NVRAM command line option 32
RLU8 linker
/NVRAMP command line option 32
RLU8 linker
/ROMWIN command line option 32
RLU8 linker
/PDIF command line option 33

S

setup program 4
system include path 43
system paths 40
system requirements 4

U

uninstalling ~ 4

V

version information 45
View menu
Output 35

W

Warnings Tab
Compile/assemble 22
Word wrap 42

IDEU8
User's Manual

SQ003096E301

9th Edition
ISSUE DATE: Oct 01, 2011

©2008-2011 LAPIS Semiconductor Co., Ltd.