

### Introduction

This chapter familiarizes you with the main features of the Nios II integrated development environment (IDE). This chapter is only a brief introduction to the look-and-feel of the Nios II IDE—it is not a user guide. The easiest way to get started using the Nios II IDE is to launch the tool and perform the Nios II software development tutorial, available in the help system.



Because of evolution and improvement of the software, the figures in this chapter may not match exactly what you see in the actual software.

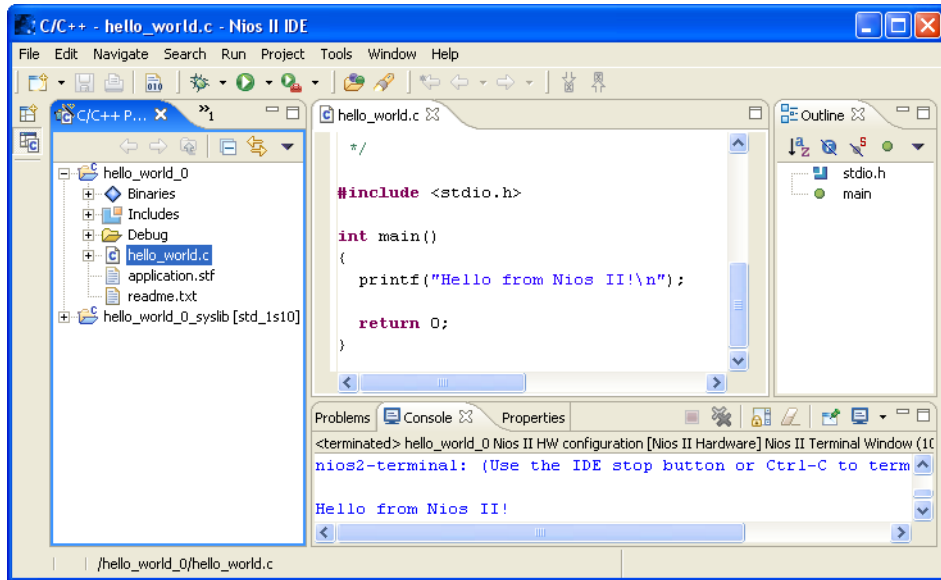


For more information on all IDE-related topics, refer to the Nios II IDE help system.

### The Nios II IDE Workbench

The term “workbench” refers to the desktop development environment for the Nios II IDE. The workbench is where you edit, compile and debug your programs. [Figure 2–1](#) shows an example of the workbench.

Figure 2-1. The Nios II IDE Workbench



## Perspectives, Editors & Views

Each workbench window contains one or more perspectives. Each perspective provides a set of capabilities aimed at accomplishing a specific type of task. For example, [Figure 2-1](#) shows the C/C++ development perspective.

Most perspectives in the workbench comprise an editor area and one or more views. An editor allows you to open and edit a project resource (i.e., a file, folder, or project). Views support editors, provide alternative presentations, and ways to navigate the information in your workbench. [Figure 2-1](#) shows a C program open in the editor, and the C/C++ **Projects** view in the left-hand pane of the workbench. The C/C++ **Projects** view displays information about the contents of open projects.

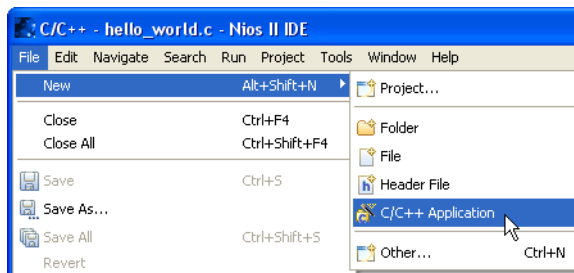
Any number of editors can be open at once, but only one can be active at a time. The main menu bar and toolbar for the workbench window contain operations that are applicable to the active editor. Tabs in the editor area indicate the names of resources that are currently open for editing. An asterisk (\*) indicates that an editor has unsaved changes. Views can also provide their own menus and toolbars, which, if present, appear along the top edge of the view. To open the menu for a view, click

the drop-down arrow icon at the right of the view's toolbar or right-click in the view. A view may appear on its own, or stacked with other views in a tabbed notebook.

## Creating a New Project

The Nios II IDE provides a **New Project** wizard that guides you through the steps to create a new C/C++ application project. To start the C/C++ application **New Project** wizard, click File, New, C/C++ Application. See [Figure 2–2](#).

**Figure 2–2. Starting the C/C++ Application New Project Wizard**



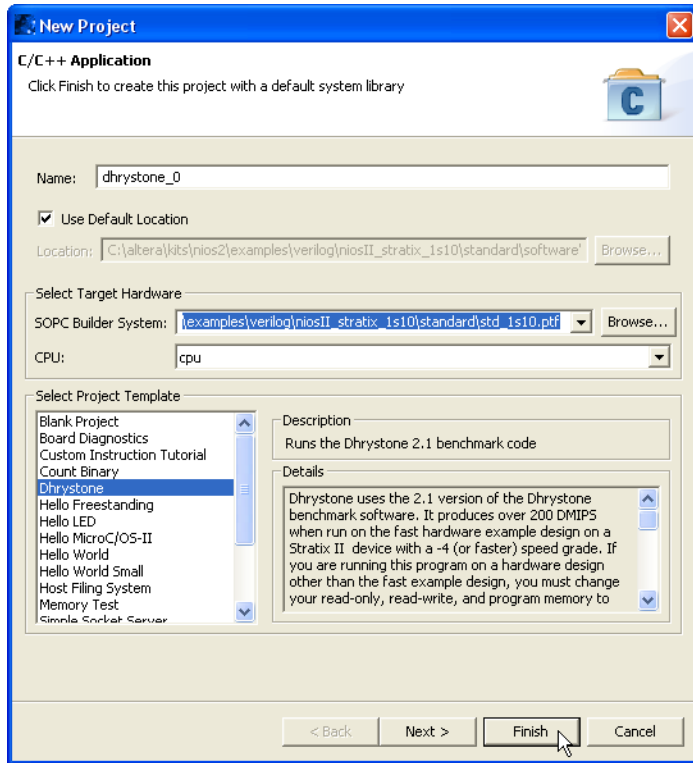
The C/C++ application **New Project** wizard prompts you to specify:

1. A name for your new project.
2. The target hardware.
3. A template for the new project.

Project templates are ready-made, working designs that serve as examples to show you how to structure your own projects. It is often easier to start with a working “Hello World” project, than to start a blank project from scratch.

[Figure 2–3](#) shows the C/C++ application **New Project** wizard, with the template for a Dhrystone benchmark design selected.

Figure 2–3. The C/C++ Application New Project Wizard

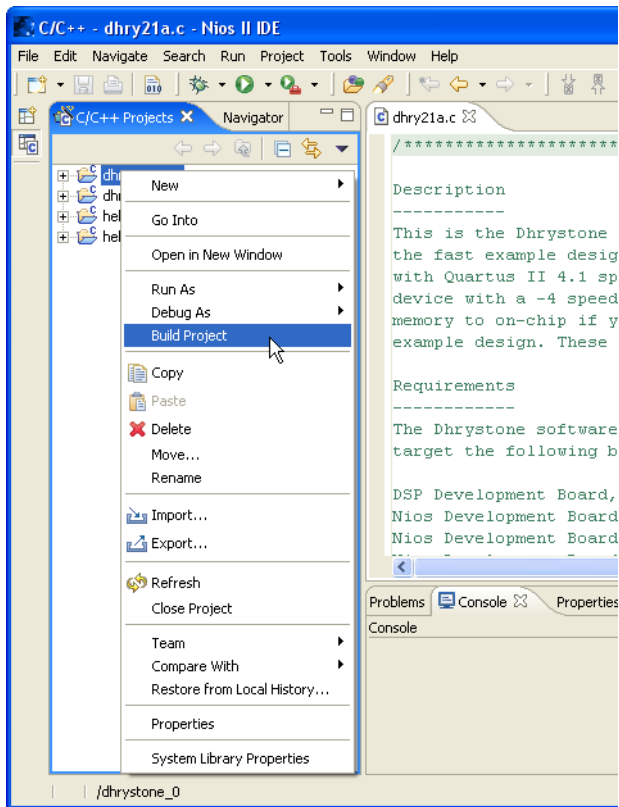


After you click **Finish**, the Nios II IDE creates the new project. The IDE also creates a system library project, `*_syslib` (for example, `dhrystone_0_syslib` for Figure 2–3). These projects show up in the **C/C++ Projects** view of the workbench.

## Building & Managing Projects

Right-clicking on any resource (a file, folder, or project) opens a context-sensitive menu with operations you can perform on the resource. Right-clicking is usually the quickest way to find the operation you need, though operations are also available in menus and toolbars.

To compile a project, right-click the project in the **C/C++ Projects** view, and choose **Build Project**. Figure 2–4 shows the context-sensitive menu for the project `dhrystone_0`, with the **Build Project** option chosen. When building, the Nios II IDE first builds the system library project (and any other project dependencies), and then compiles the main project. Any warnings or errors are displayed in the **Tasks** view.

**Figure 2–4. Building a Project Using the Context-Sensitive (Right-Click) Menu**

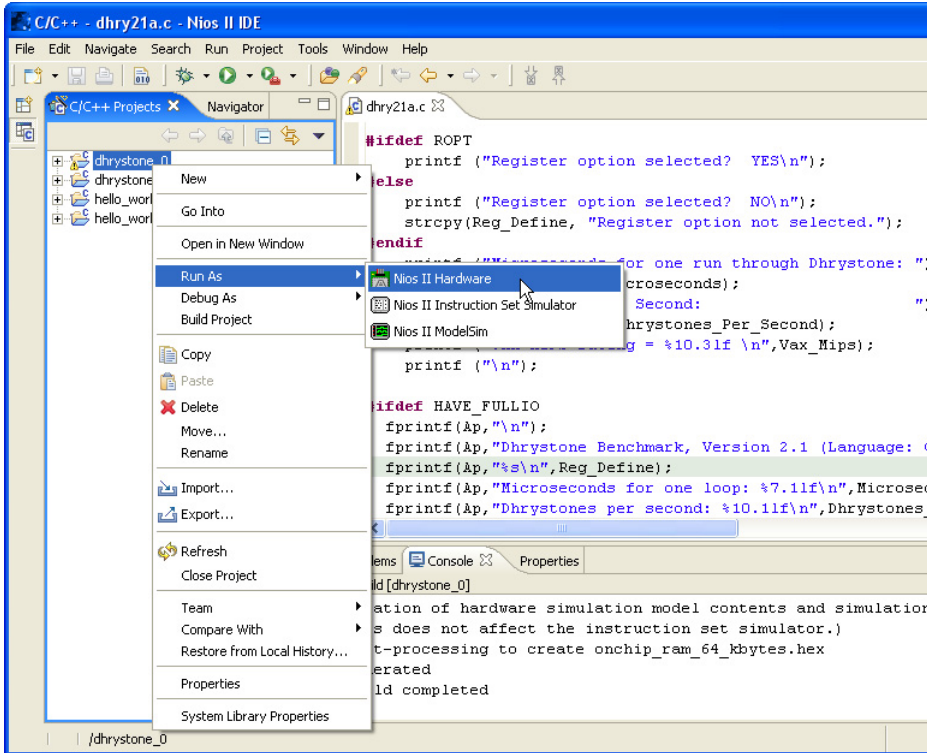
Right-clicking a project in C/C++ Projects also allows you to access the following important options for managing the project:

- **Properties**—Manage the dependencies on target hardware and other projects
- **System Library Properties**—Manage hardware-specific settings, such as communication devices and memory partitioning
- **Build Project**—i.e., make
- **Rebuild Project**—i.e., make all
- **Run As**—Run the program on hardware or the ISS
- **Debug As**—Debug the program on hardware or the ISS

## Running & Debugging Programs

Run and debug operations are available by right-clicking the project. The Nios II IDE allows you to run or debug the project either on a target board or the Nios II instruction set simulator (ISS). For example, to run the program on a target board, right-click the project in the **C/C++ Projects** view, then click **Run As, Nios II Hardware**. See [Figure 2-5](#). Character I/O to stdout and stderr are displayed in the Console view.

**Figure 2-5. Running a Program on Target Hardware**



Starting a debug session is similar to starting a run session. For example, to debug the program on the ISS, right-click the project in the **C/C++ Projects** view, then click **Debug As, Nios II Instruction Set Simulator**. See [Figure 2-6](#).

Figure 2-6. Launching the Instruction Set Simulator

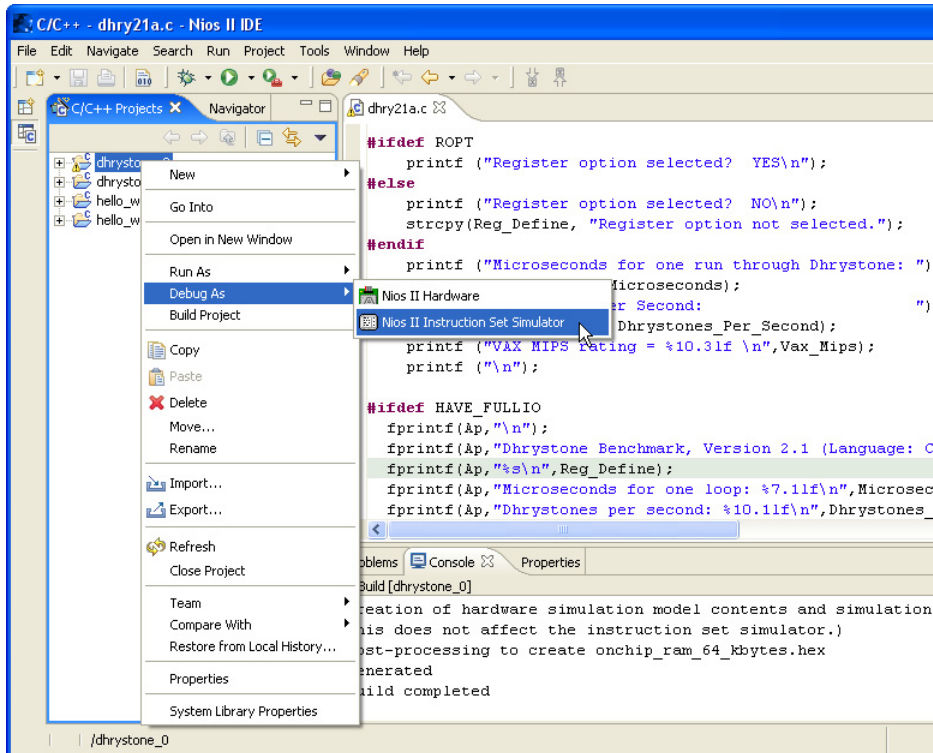
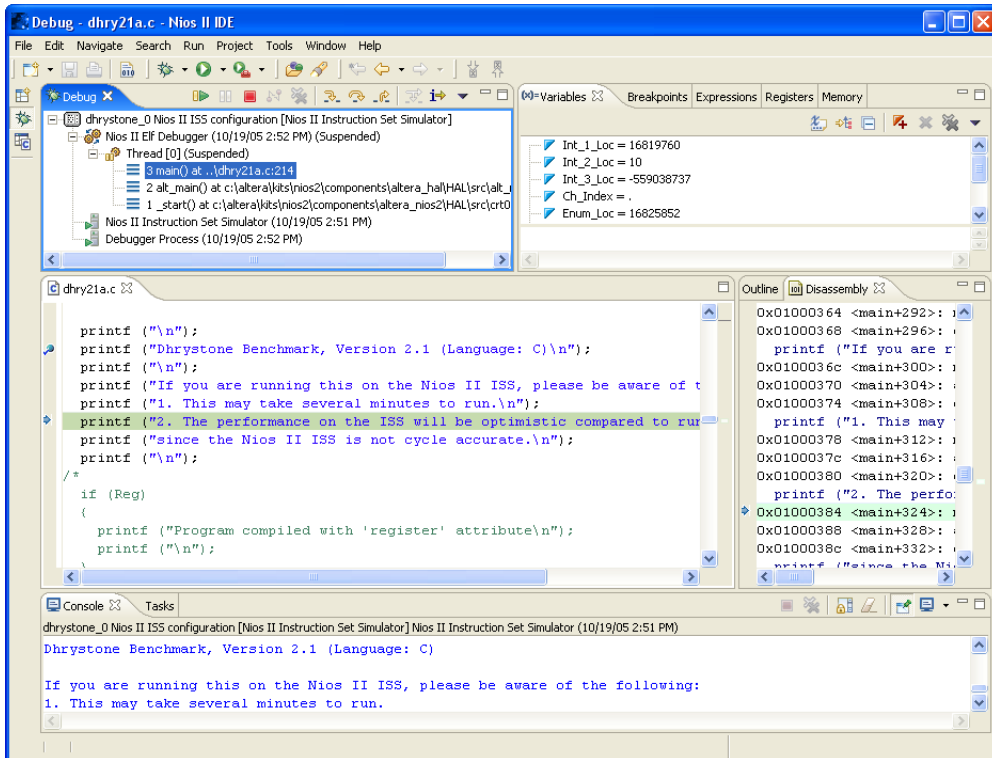


Figure 2-7 shows a debug session in progress for the `dhrystone_0` project.

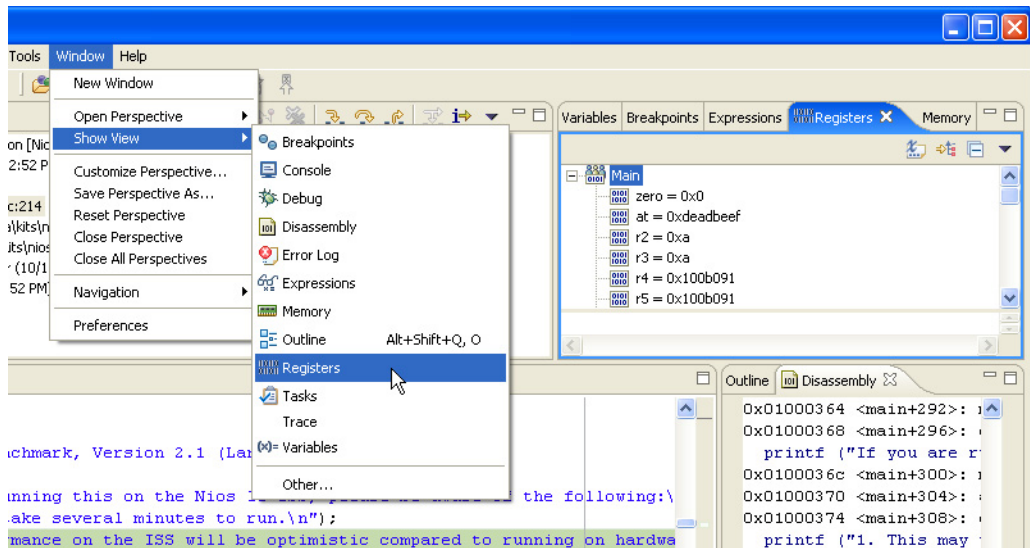
Figure 2-7. Debugging `dhrystone_0` on the ISS

Launching the debugger changes the workbench perspective to the debug perspective. You can easily switch between the debug perspective and the C/C++ development perspective, by clicking on the perspective icons on the left-most side of the workbench window.

After you start a debug session, the debugger loads the program, sets a breakpoint at `main()`, and begins executing the program. You use the usual controls to step through the code: Step Into, Step Over, Resume, Terminate, etc. To set a breakpoint, double click in the left-hand margin of the code view, or right-click and choose **Add Breakpoint**.

The Nios II IDE offers many debug views that allow you to examine the status of the processor while debugging: Variables, Expressions, Registers, Memory, etc. Figure 2-8 shows the Registers view.



**Figure 2–8. The Registers View While Debugging**

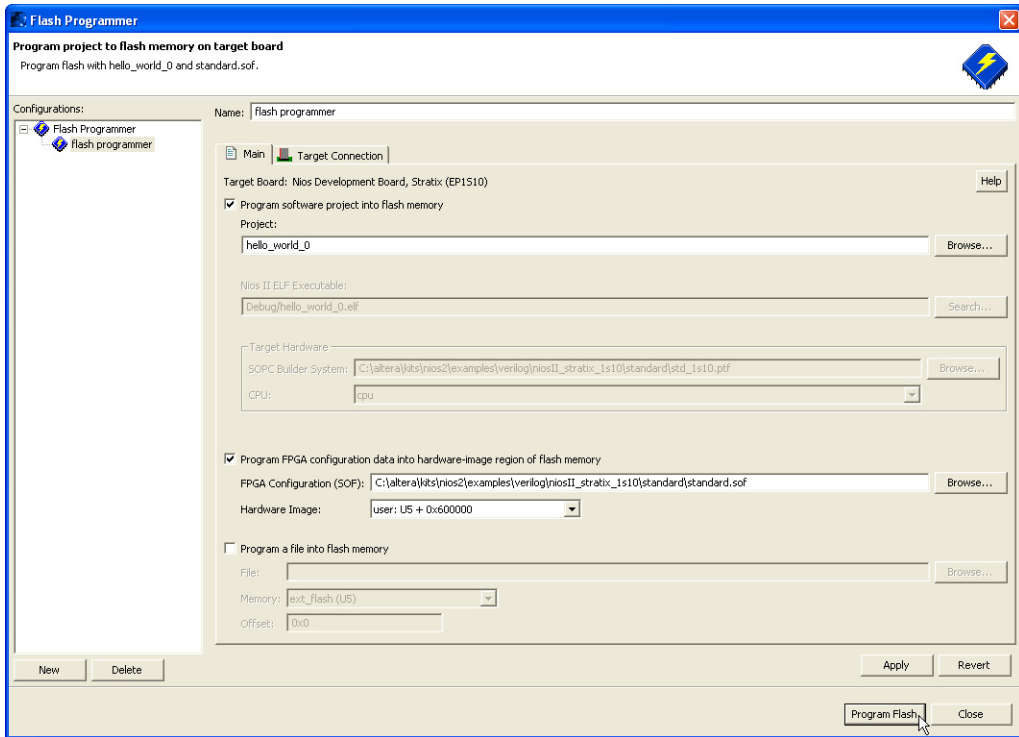
## Programming Flash

Many Nios II processor systems use external flash memory to store one or more of the following items:

- Program code
- Program data
- FPGA configuration data
- File systems

The Nios II IDE provides a Flash Programmer utility to help you manage and program the contents of flash memory. [Figure 2–9](#) shows the Flash Programmer.

Figure 2–9. The Nios II IDE Flash Programmer



## Help System

The Nios II IDE help system provides documentation on all IDE-related topics. To launch the help system, click Help, **Help Contents**. On Windows, you can also press **F1** at any time for context-sensitive help. The Nios II IDE help system contains hands-on tutorials that guides you step-by-step through the process of creating, building, and debugging a project. [Figure 2–10](#) shows the Nios II IDE help system displaying a tutorial.

Figure 2–10. Tutorials in the Nios II IDE Help System

