Introduction

The *Nios® II Software Developer's Handbook* provides the basic information needed to develop software for the Altera® Nios II processor. This handbook describes the Nios II software development environment, the Nios II Embedded Design Suite (EDS) tools available to you, and the process for developing software.

This chapter provides a high-level overview of the software development environments for the Nios II processor, and contains the following sections:

- "Nios II Software Development Environment" on page 1–1
- "Nios II EDS Development Flows" on page 1–2
- "Nios II Programs" on page 1–4
- "Altera Software Packages" on page 1–5
- "Example Designs" on page 1–6
- "Third-Party Support" on page 1–7
- "Further Nios II Information" on page 1–7

Prerequisites

The *Nios II Software Developer's Handbook* assumes you have a basic familiarity with embedded processor concepts. You do not need to be familiar with any specific Altera technology or with Altera development tools. Familiarity with Altera hardware development tools can give you a deeper understanding of the reasoning behind the Nios II software development environment. However, software developers can create and debug applications without further knowledge of Altera technology.

Finding Nios II EDS Files

When you install the Nios II EDS, you specify a root directory for the EDS file structure. This root directory must be adjacent to the Quartus® II installation. For example, if the Nios II EDS 10.0 is installed on the Windows operating system, the root directory might be **c:\altera\100\nios2eds**.

For simplicity, this handbook refers to this directory as *<Nios II EDS install path>*.

Nios II Software Development Environment

The Nios II EDS provides a consistent software development environment that works for all Nios II processor systems. With the Nios II EDS running on a host computer, an Altera FPGA, and a JTAG download cable (such as an Altera USB-Blaster™ download cable), you can write programs for and communicate with any Nios II processor system. The Nios II processor's JTAG debug module provides a single, consistent

method to connect to the processor using a JTAG download cable. Accessing the processor is the same, regardless of whether a device implements only a Nios II processor system, or whether the Nios II processor is embedded deeply in a complex multiprocessor system. Therefore, you do not need to spend time manually creating interface mechanisms for the embedded processor.

The Nios II EDS includes proprietary and open-source tools (such as the GNU C/C++ tool chain) for creating Nios II programs. The Nios II EDS automates board support package (BSP) creation for Nios II processor-based systems, eliminating the need to spend time manually creating BSPs. The BSP provides a C/C++ runtime environment, insulating you from the hardware in your embedded system. Altera BSPs contain the Altera hardware abstraction layer (HAL), an optional RTOS, and device drivers.

Nios II EDS Development Flows

A development flow is a way of using a set of development tools together to create a software project. The Nios II EDS provides the following development flows for creating Nios II programs:

■ The Nios II Software Build Tools (SBT), which provides two user interfaces:

  ■ The Nios II SBT command line

  ■ The Nios II SBT for Eclipse™

■ The Nios II integrated development environment (IDE)

The two design flows share a number of development tools. However, the flows differ markedly in how they create makefiles. The development flows are not interchangeable. Source code for your applications, libraries, and drivers works in either flow, but the makefiles in the two flows are different and not compatible.

☞ The future direction of Nios II software development support is toward the Nios II SBT and away from the Nios II IDE. The Nios II EDS provides a tool for converting Nios II IDE projects to Nios II SBT projects.

👣 For details about Nios II IDE project conversion, refer to "Porting Nios II IDE Projects to the Software Build Tools" in *Appendix A. Using the Nios II Integrated Development Environment* in the *Nios II Software Developer's Handbook*.

☞ Hardware designs created with SOPC Builder 7.0 or earlier must either use the Nios II IDE development flow or be updated.

The Nios II SBT Development Flow

The Nios II SBT allows you to create Nios II software projects, with detailed control over the software build process. The same Nios II SBT utilities, scripts and Tcl commands are available from both the command line and the Nios II SBT for Eclipse graphical user interface (GUI).

The SBT allows you to create and manage single-threaded programs as well as complex applications based on an RTOS and middleware libraries available from Altera and third-party vendors.

The SBT provides powerful Tcl scripting capabilities. In a Tcl script, you can query project settings, specify project settings conditionally, and incorporate the software project creation process in a scripted software development flow. Tcl scripting is supported both in Eclipse and at the command line.

For information about Tcl scripting, refer to the *Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook.*

The Nios II SBT for Eclipse

The Nios II SBT for Eclipse is a thin GUI layer that runs the Nios II SBT utilities and scripts behind the scenes, presenting a unified development environment. The SBT for Eclipse provides a consistent development platform that works for all Nios II processor systems. You can accomplish all software development tasks within Eclipse, including creating, editing, building, running, debugging, and profiling programs.

The Nios II SBT for Eclipse is based on the popular Eclipse framework and the Eclipse C/C++ development toolkit (CDT) plugins. The Nios II SBT creates your project makefiles for you, and Eclipse provides extensive capabilities for interactive debugging and management of source files.

The SBT for Eclipse also allows you to import and debug projects you created in the Nios II Command Shell.

For details about the Nios II SBT for Eclipse, refer to the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook.* For details about Eclipse, visit the Eclipse Foundation website (www.eclipse.org).

The Nios II SBT Command Line

In the Nios II SBT command line development flow, you create, modify, build, and run Nios II programs with Nios II SBT commands typed at a command line or embedded in a script. You run the Nios II SBT commands from the Nios II Command Shell.

For further information about the Nios II SBT in command-line mode, refer to the *Getting Started from the Command Line* chapter of the *Nios II Software Developer's Handbook.*

To debug your command-line program, import your SBT projects to Eclipse. You can further edit, rebuild, run, and debug your imported project in Eclipse.

The Nios II IDE Development Flow

The Nios II IDE development flow is an integrated environment in which you can create, modify, build, run, and debug Nios II programs with the Nios II IDE GUI. The Nios II IDE flow does not use the Nios II SBT. The makefiles it creates cannot be user-managed. This flow provides limited control over the build process and the project settings, with no support for customized scripting.

☞ In most cases, you should create new projects using the Nios II SBT from the command line or from Eclipse. IDE support is for the following situations:

- Working with pre-existing Nios II IDE software projects
- Creating new projects for the Nios II C2H compiler
- Debugging with the FS2 console

For further information about the Nios II IDE, refer to *Appendix A. Using the Nios II Integrated Development Environment* in the *Nios II Software Developer's Handbook*.

The remainder of this handbook assumes that you are using the Nios II SBT.

Nios II Programs

Each Nios II program you develop consists of an application project, optional user library projects, and a BSP project. You build your Nios II program to create an Executable and Linking Format File (**.elf)** which runs on a Nios II processor.

The Nios II SBT creates software projects for you. Each project is based on a makefile. This section discusses makefiles and projects.

Makefiles and the SBT

The makefile is the central component of a Nios II software project, whether the project is created with the Nios II SBT for Eclipse, or on the command line. The makefile describes all the components of a software project and how they are compiled and linked. With a makefile and a complete set of C/C++ source files, your Nios II software project is fully defined.

As a key part of creating a software project, the SBT creates a makefile for you. Nios II projects are sometimes called "user-managed," because you, the user, are responsible for the content of the project makefile. You use the Nios II SBT to control what goes in the makefile.

☞ Makefiles for Nios II IDE projects differ from the SBT makefiles described in this section.

The *Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* provides detailed information about creating makefiles. For a description of Nios II IDE makefiles, refer to *Appendix A. Using the Nios II Integrated Development Environment* in the *Nios II Software Developer's Handbook.*

Nios II Software Project Types

The following sections describe the project types that constitute a Nios II program.

Application Project

A Nios II C/C++ application project consists of a collection of source code, plus a makefile. A typical characteristic of an application is that one of the source files contains function main(). An application includes code that calls functions in libraries and BSPs. The makefile compiles the source code and links it with a BSP and one or more optional libraries, to create one **.elf** file.

User Library Project

A user library project is a collection of source code compiled to create a single library archive file (**.a**). Libraries often contain reusable, general purpose functions that multiple application projects can share. A collection of common arithmetical functions is one example. A user library does not contain a `main()` function.

BSP Project

A Nios II BSP project is a specialized library containing system-specific support code. A BSP provides a software runtime environment customized for one processor in an SOPC Builder system. The Nios II EDS provides tools to modify settings that control the behavior of the BSP.

A BSP contains the following elements:

■ Hardware abstraction layer—For information, refer to the *Overview of the Hardware Abstraction Layer* chapter of the *Nios II Software Developer's Handbook*.

■ Optional custom newlib C standard library—For information, refer to the *Overview of the Hardware Abstraction Layer* chapter of the *Nios II Software Developer's Handbook*. The complete HTML documentation for newlib resides in the Nios II EDS directory.

■ Device drivers—For information, refer to "Nios II Software Projects" in the *Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook.*

■ Optional software packages—For information, refer to "Altera Software Packages".

■ Optional real-time operating system—For information, refer to the *MicroC/OS-II Real-Time Operating System* chapter of the *Nios II Software Developer's Handbook*.

Altera Software Packages

The Nios II EDS includes software packages to extend the capabilities of your software. You can include these software packages in your BSP. Table 1–1 shows those Altera Nios II software packages that are distributed with the Nios II EDS.

Table 1–1. Software Packages

Name	Description
NicheStack TCP/IP Stack - Nios II Edition	Refer to the *Ethernet and the NicheStack TCP/IP Stack - Nios II Edition* chapter of the *Nios II Software Developer's Handbook*.
Read-only zip file system	Refer to the *Read-Only Zip File System* chapter of the *Nios II Software Developer's Handbook*.
Host file system	Refer to the *Developing Programs Using the Hardware Abstraction Layer* chapter of the *Nios II Software Developer's Handbook*.

Additional software packages are available from Altera's partners. For a complete list, refer to the Embedded Software page of the Altera website.

Example Designs

The Nios II EDS includes documented hardware example designs and software examples to demonstrate all prominent features of the Nios II processor and the development environment. The examples can be very helpful as you start the development of your custom design. They provide a stable starting point for exploring design options. Also, they demonstrate many commonly used features of the Nios II EDS.

Hardware Examples

You can run Nios II hardware designs on many Altera development boards. The hardware examples for each Altera development board reside in the following location:

*<Nios II EDS install path>***/examples/verilog**/*<board>*

where *<board>* is the name of the development board. For example, the Triple Speed Ethernet example design for the Nios II 3C120 development board resides in *<Nios II EDS install path>***/examples/verilog/niosII_cycloneIII_3c120/ triple_speed_ethernet_design**.

☞ **applications_processor_mmu_design** is intended to demonstrate Linux. This design does not work with the SBT, because the SBT does not support the Nios II MMU.

Software Examples

You can run Nios II software examples that run on many of the hardware example designs described in the previous section.

The Nios II software examples include scripts and templates to create the software projects using the Nios II SBT. These scripts and templates do everything necessary to create a BSP and an application project for each software example.

Figure 1–1 shows the directory structure under each hardware example design. There are multiple software examples and BSP examples, each with its own directory. Each software example directory contains a **create-this-app** script and each BSP example directory contains a **create-this-bsp** script. These scripts create software projects, as demonstrated in "Getting Started" in the *Getting Started from the Command Line* chapter of the *Nios II Software Developer's Handbook*.

Figure 1–1. Software Example Design Directory Structure
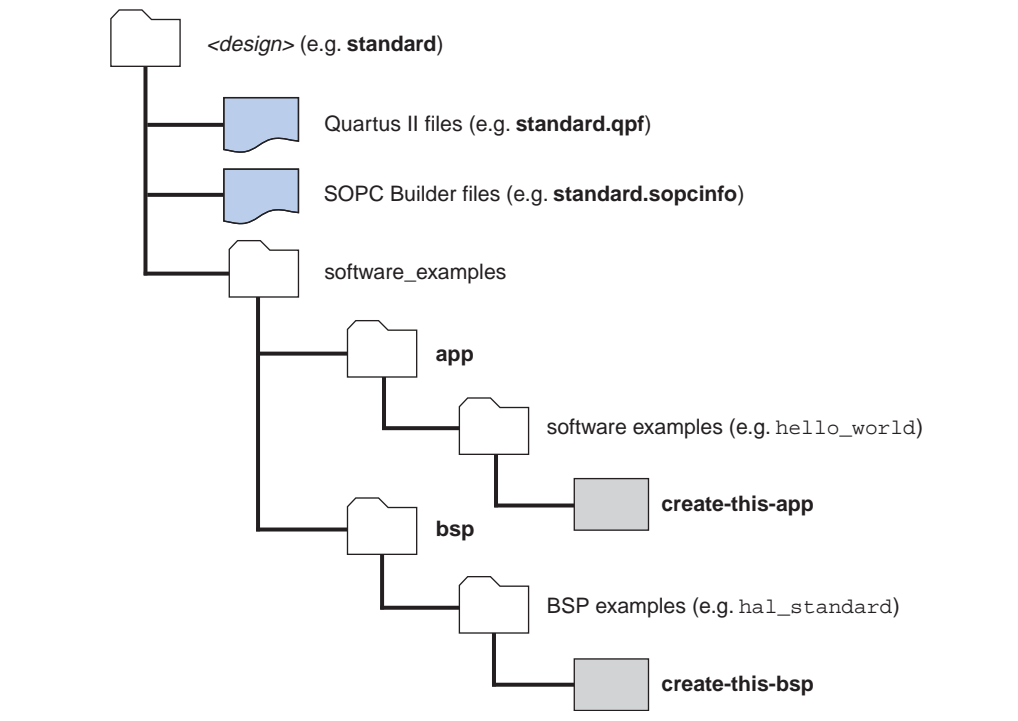
Third-Party Support

Several third-party vendors support the Nios II processor, providing products such as design services, operating systems, stacks, other software libraries, and development tools.

For the most up-to-date information about third-party support for the Nios II processor, visit the Nios II Processor page of the Altera website.

Further Nios II Information

This handbook is one part of the complete Nios II processor documentation suite. Consult the following references for further Nios II information:

■ The *Nios II Processor Reference Handbook* defines the processor hardware architecture and features, including the instruction set architecture.

■ The *Embedded Peripherals IP User Guide* provides a reference for the peripherals distributed with the Nios II processor. This handbook describes the hardware structure and Nios II software drivers for each peripheral.

■ The *Embedded Design Handbook* describes how to use Altera software development tools effectively, and recommends design styles and practices for developing, debugging, and optimizing embedded systems.

■ The Altera Knowledge Database is an Internet resource that offers solutions to frequently asked questions with an easy-to-use search engine. Visit the Knowledge Database page of the Altera website.

■ Altera application notes and tutorials offer step-by-step instructions on using the Nios II processor for a specific application or purpose. These documents are available on the Literature: Nios II Processor page of the Altera website.

■ The Nios II EDS documentation launchpad. The launchpad is an HTML page installed with the Nios II EDS, which provides links to Nios II documentation, examples, and other resources. The way you open the launchpad depends on your software platform.

■ In the Windows operating system, on the Start menu, point to **Programs** > **Altera** > **Nios II EDS**, and click **Nios II** *<version>* **Documentation**.

■ In the Linux operating system, open *<Nios II EDS install path>***/documents/ index.html** in a web browser.

Document Revision History

Table 1–2 shows the revision history for this document.

Table 1–2. Document Revision History (Part 1 of 2)

Date	Version	Changes
February 2011	10.1.0	Removed "Referenced Documents" section.
July 2010	10.0.0	Installation method changed; Nios II EDS always installed in a directory adjacent to Quartus II tools.
November 2009	9.1.0	■ Described the Nios II Software Build Tools for Eclipse. ■ Nios II IDE information moved to *Appendix A. Using the Nios II Integrated Development Environment*. ■ Detailed Nios II Software Build Tools utility information moved to *Nios II Software Build Tools*.
March 2009	9.0.0	■ Incorporate information formerly in *Altera-Provided Development Tools* chapter. ■ Describe BSP Editor. ■ Reorganize and update information and terminology to clarify role of Nios II Software Build Tools. ■ Describe `-data` argument for IDE command-line tools. ■ Correct minor typographical errors.
May 2008	8.0.0	■ Add "What's New" section. ■ SOPC Information File (**.sopcinfo**). ■ Example designs removed from EDS. ■ Memory management unit (MMU) added to Nios II core.
October 2007	7.2.0	Maintenance release.

Table 1–2. Document Revision History (Part 2 of 2)

Date	Version	Changes
May 2007	7.1.0	■ Revise entire chapter to introduce Nios II EDS design flows, Nios II programs, Nios II Software Build Tools, and Nios II BSPs. ■ Add table of contents to Introduction section. ■ Add "Referenced Documents" section.
March 2007	7.0.0	Maintenance release.
November 2006	6.1.0	Maintenance release.
May 2006	6.0.0	Maintenance release.
October 2005	5.1.0	Maintenance release.
May 2005	5.0.0	Maintenance release.
May 2004	1.0	Initial release.