

## About These Release Notes

These release notes cover versions 9.0 through 10.0 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history and errata for the Nios II EDS.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.




For the most up-to-date errata for this release, refer to the latest version of the *Nios II Embedded Design Suite Release Notes and Errata* on the Altera website. You can refer to release notes and errata for the Nios II processor core in the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

## Revision History

Table 1 shows the revision history for the Nios II EDS, including the Nios II C2H Compiler.

**Table 1.** Nios II Embedded Design Suite Revision History

Version	Date	Description
10.0	July 2010	<ul style="list-style-type: none"> <li>■ Software Build Tools (SBT) for Eclipse <ul style="list-style-type: none"> <li>■ Updated Eclipse to version 3.5 and Eclipse C/C++ development toolkit (CDT) to version 6.0</li> <li>■ Improved Source File management</li> <li>■ Supports Multi-CPU launch</li> </ul> </li> <li>■ Updated to Nios II GCC 4.1.2 tool chain <ul style="list-style-type: none"> <li>■ Supports both SBT for Eclipse and Nios II Command Shell</li> <li>■ GCC 4 improves compilation time compared to GCC 3 on Windows</li> </ul> </li> <li>■ Host platform <ul style="list-style-type: none"> <li>■ Supports Windows 7</li> <li>■ Supports SUSE Linux Enterprise 11</li> <li>■ Support for SUSE Linux Enterprise 9 discontinued</li> </ul> </li> <li>■ New Nios II software design examples for Reduced Gigabit Media Independent Interface (RGMI) <ul style="list-style-type: none"> <li>■ Web Server (RGMI)</li> <li>■ Simple Socket Server (RGMI)</li> </ul> </li> <li>■ Legacy hardware design examples (Stratix® II and Cyclone® II) are removed</li> <li>■ Installer changes <ul style="list-style-type: none"> <li>■ Nios II EDS must be installed next to Quartus® II</li> </ul> </li> </ul>
9.1 SP2	March 2010	Maintenance release
9.1 SP1	January 2010	Maintenance release
9.1	November 2009	<ul style="list-style-type: none"> <li>■ The Nios II SBT for Eclipse, an integrated development environment based on Eclipse. Recommended for all new Nios II projects.</li> <li>■ The Nios II Integrated Development Environment (IDE) is available in the Nios II EDS, but not recommended for new projects.</li> <li>■ The Nios II Studio is discontinued.</li> <li>■ The vectored interrupt controller (VIC) core offers high-performance, low-latency interrupt handling.</li> <li>■ Nios II Design Examples with memory management unit (MMU): <ul style="list-style-type: none"> <li>■ Nios II 3C120 Hardware Reference Design with MMU, Cyclone III Edition</li> <li>■ Nios II 4S230 Hardware Reference Design with MMU, Stratix IV GX Edition</li> </ul> Commercial and open-source support for embedded Linux is available for these systems. </li> <li>■ The Embedded Systems Lab for DE1 for Quartus and Nios II EDS v9.1 is available for download from the Nios Community Wiki (<a href="http://www.nioswiki.com">www.nioswiki.com</a>).</li> <li>■ Tcl scripting for software packages and drivers enhanced to support callback functions.</li> </ul>
9.0	March 2009	Added the <code>nios2-convert-ide2sbt</code> tool, which converts projects from the Nios II IDE development flow to the Nios II SBT development flow
8.1 SP1	January 2009	Nios Community Wiki URL changing ( <a href="http://www.nioswiki.com">www.nioswiki.com</a> )
8.1	November 2008	Maintenance release

 For more information on new features and changes, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*. For information about changes to the Nios II processor core, refer to the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

## Notice about Environment Variables

The `QUARTUS_ROOTDIR` and `SOPC_KIT_NIOS2` environment variables will be deprecated in a future release. If you have custom scripts that use these variables, you should be prepared to change them to be independent of these variables, or to set the environment variables yourself.

## Revision Details

This section provides details about some of the Nios II EDS revisions shown in [Table 1](#).

## Nios II Software Build Tools for Eclipse

The Nios II Software Build Tools (SBT) for Eclipse is a set of plugins based on the popular Eclipse framework and the Eclipse C/C++ development toolkit (CDT) plugins. The Nios II SBT for Eclipse provides a consistent development platform that works for all Nios II processor systems. You can accomplish most Nios II software development tasks within Eclipse, including creating, editing, building, running, debugging, and profiling programs. The Nios II SBT for Eclipse is built from the ground up as a set of plugins to standard Eclipse. The Nios II SBT for Eclipse is based on the Nios II Software Build Tools, a set of powerful commands, utilities and scripts for creating applications, board support packages and software libraries, and managing build options.

[Table 2](#) compares the Nios II Integrated Development Environment (IDE) with the Nios II SBT for Eclipse.

**Table 2.** Differences Between Nios II SBT for Eclipse and Nios II IDE (Part 1 of 2)

Feature	Nios II IDE	Nios II SBT for Eclipse
Types of projects	<ul style="list-style-type: none"> <li>■ Application</li> <li>■ Board support package (BSP)—Called a system library in the IDE</li> <li>■ User Library (optional)</li> </ul>	<ul style="list-style-type: none"> <li>■ Application</li> <li>■ BSP</li> <li>■ User Library (optional)</li> </ul>
Handling updates to the SOPC Builder system	Updates the BSP automatically	You must initiate the BSP update (in the Nios II menu, click Generate BSP)
Options for creating projects	Application from template. BSP is created automatically as part of application.	<ul style="list-style-type: none"> <li>■ New Application and BSP from template</li> <li>■ Standalone application</li> <li>■ Standalone BSP</li> <li>■ Standalone library</li> </ul>
Handling updates to BSP	Entire BSP is rebuilt. This process can be time-consuming.	Smart rebuild only recompiles changed files
Compatibility with command-line flow	None	Full compatibility with the Nios II SBT in the Nios II Command Shell


**Table 2.** Differences Between Nios II SBT for Eclipse and Nios II IDE (Part 2 of 2)

Feature	Nios II IDE	Nios II SBT for Eclipse
Modifying BSP properties	Limited control over BSP properties through the System Library page of the Properties dialog box.	Full control over BSP properties through the BSP Editor
Nios II C2H Compiler	Supported	Not supported
Nios II Flash Programmer	Limited control over the contents of the flash image	<ul style="list-style-type: none"> <li>■ Allows you to create complex flash images containing Executable and Linking Format Files (.elf), SRAM Object Files (.sof), and binary data files in specified locations</li> <li>■ Allows you to save flash programmer settings for repeatable flash programming</li> </ul>
Scripting and regression testing	Not Available	Project creation and build can be scripted using the SBT command line


### Converting IDE Projects to the SBT for Eclipse

If you have a pre-existing Nios II IDE project, before you can continue development in the Nios II SBT for Eclipse, you must convert it to the SBT and import it to the Eclipse environment. To convert and import an IDE project to the Nios II SBT for Eclipse, execute the following steps:

1. Convert the IDE project to an SBT-based project with the **nios2-convert-ide2sbt** command line utility.
2. Import the converted project to the Nios II SBT for Eclipse.

 For a detailed description of the conversion process, refer to “Porting Nios II IDE Projects to the Software Build Tools” in the *Using the Nios II Integrated Development Environment* appendix of the *Nios II Software Developer’s Handbook*. For a description of the process of importing a project, refer to “Importing a Command-Line Project” in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer’s Handbook*.

### Learning about the Nios II SBT for Eclipse

 Refer to the following documents to help you get acquainted with the Nios II SBT for Eclipse and related tools:

- *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer’s Handbook*
- *Nios II Hardware Development Tutorial*
- The [Comprehensive Nios II Embedded Design Suite](#) support page
- The [Nios II Software Build Tools for Eclipse](#) support page

## Errata Summary

Table 3 summarizes the issues that affect the Nios II Embedded Design Suite versions 8.1 through 9.1 SP2. For a detailed description of each issue, click on the issue name.

 Refer to the Altera [Knowledge Database](#) for older errata and solutions.

**Table 3.** Nios II EDS Errata (1) (Part 1 of 5)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
<b>Nios II SBT for Eclipse</b>							
July 2010	<a href="#">Nios II SBT for Eclipse Cannot Step into GCC 3 Library Source Code</a>	10	✓	✓	✓	✓	—
	<a href="#">Error Building Imported Project: 'No rule to make target'</a>	10	✓	—	—	—	—
	<a href="#">Error Building Imported Project: 'target pattern contains no %'</a>	11	✓	—	—	—	—
	<a href="#">Missing Nios II Perspective</a>	12	✓	✓	✓	✓	—
	<a href="#">Error Debugging Imported Project: 'Can't find a source file'</a>	12	Fixed	✓	✓	✓	—
	<a href="#">Error Marker Persists on BSP Project After Build Error Corrected</a>	12	✓	✓	✓	✓	—
	<a href="#">Assembly Language Source File Not Found</a>	13	Fixed	✓	✓	✓	—
	<a href="#">'Exclude from build' Not Supported</a>	13	✓	—	—	—	—
May 2010	<a href="#">Errors Creating or Importing Software Projects</a>	13	✓	✓	✓	✓	—
	<a href="#">Nios II Options Do Not Appear in Eclipse</a>	14	✓	✓	✓	✓	—
Feb 2010	<a href="#">Build Errors on Software for Pre-Existing Design with SG-DMA</a>	14	✓	✓	✓	✓	—
	<a href="#">Spurious System ID Mismatch Error</a>	15	✓	✓	✓	✓	—
	<a href="#">Errors Debugging as Local C/C++ Application</a>	15	✓	✓	✓	✓	—
Nov 2009	<a href="#">Error Messages on Console When Debugging</a>	15	✓	✓	✓	✓	—
	<a href="#">java.lang.NullPointerException Error When Running Application</a>	16	✓	✓	✓	✓	—
	<a href="#">Update Failure When Setting Empty Properties in the Nios II Properties Page</a>	16	✓	✓	✓	✓	—
	<a href="#">Errors Converting Nios II IDE Multiprocessor Project</a>	17	✓	✓	✓	✓	—
	<a href="#">DHCP Time-Out on EP3C120</a>	17	✓	✓	✓	✓	—
	<a href="#">No Nios II Multiprocessor Collection Run Configuration</a>	17	✓	✓	✓	✓	—
	<a href="#">Stop on Startup Option in Run Configuration Has No Effect</a>	18	✓	✓	✓	✓	—
	<a href="#">Project Imported From Command Line Fails to Build Correctly</a>	18	✓	✓	✓	✓	—
	<a href="#">Debugger Breaks in crt0.s Instead of main()</a>	18	✓	✓	✓	✓	—
<b>C2H Compiler</b>							
Nov 2009	<a href="#">Error "java.lang.NoClassDefFoundError" During C2H Compilation</a>	19	—	—	—	—	Fixed
Jul 2008	<a href="#">Functions Declared Without a Return Type Are Not Supported</a>	19	✓	✓	✓	✓	✓
Oct 2007	<a href="#">Pre-7.1 Systems Are Not Supported</a>	19	✓	✓	✓	✓	✓
	<a href="#">--src-dir SBT Argument Does Not Work With C2H</a>	20	✓	✓	✓	✓	✓

**Table 3.** Nios II EDS Errata (1) (Part 2 of 5)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
May 2007	Accelerator Generation Failure If Tools Are Installed in Path With Spaces	20	✓	✓	✓	✓	✓
	The C2H Compiler Regenerates an Accelerator Unnecessarily	20	✓	✓	✓	✓	✓
	Error: c2h_accelerator_base_addresses.h: No such file or directory	21	✓	✓	✓	✓	✓
	Java Heap Space Exception if Quartus II Compilation is Enabled	21	✓	✓	✓	✓	✓
	Pointer Dereferences to Volatile Types	21	✓	✓	✓	✓	✓
Dec 2006	C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	22	✓	✓	✓	✓	✓
Jun 2006	Array Elements in Structures Do Not Copy Correctly	22	✓	✓	✓	✓	✓
	Clean Build Causes Build Failure	23	✓	✓	✓	✓	✓
	Changing Build Configurations Produces Unexpected Results	23	✓	✓	✓	✓	✓
	Hardware Accelerators Remain After Deleting the Software Project (2)	23	✓	✓	✓	✓	✓
	Incorrect Results From Logical or Conditional Operation With Side-Effects	24	✓	✓	✓	✓	✓
	Launch SOPC Builder Button in C2H View	24	✓	✓	✓	✓	✓
<b>Development Boards</b>							
Jun 2006	Intermittent Failures While Accessing CompactFlash Card	24	✓	✓	✓	✓	✓
<b>Documentation Issues</b>							
July 2010	Error Building Project: 'No rule to make target'	25	✓	—	—	—	—
	AN543 Contains Incorrect Information about Updating the Flash	26	✓	✓	✓	✓	—
	Incorrect Information about Floating-Point Instruction Precision	26	Fixed	✓	✓	✓	—
	Eclipse CDT Features Not Supported by Nios II Plugins	27	✓	✓	✓	✓	—
May 2010	Incorrect Information about Nested Exceptions	28	✓	✓	✓	✓	—
Feb 2010	Missing Documentation of Interrupt API Properties	29	✓	✓	✓	✓	—
	Valid Range of hal.log_flags is -1 to 3	30	✓	✓	✓	✓	—
Nov 2009	NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP	30	✓	✓	✓	✓	✓
	Error Message After Renaming Project: "Resource is out of sync with the system"	31	✓	✓	✓	✓	—
	Obsolete Wiki URLs in Help Systems and Handbook	31	—	—	—	—	Fixed
	Missing Source Control Information	31	—	—	—	—	Fixed
Mar 2009	Compiler Flags for Building Custom Newlib	33	✓	✓	✓	✓	✓
Nov 2008	Error Message on Invalid Exception or Reset Vector	34	✓	✓	✓	✓	✓
Oct 2007	Nios II IDE Online Help Expand Buttons Do Not Work	35	✓	✓	✓	✓	✓

**Table 3.** Nios II EDS Errata (1) (Part 3 of 5)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
<b>Hardware Abstraction Layer</b>							
Feb 2010	Missing Structure Member Errors in alt_log_printf.c with Small JTAG UART Driver	35	✓	✓	✓	✓	—
	“unused variable” Warning When Building alt_main.c	36	✓	✓	✓	✓	—
	Build Warnings in Performance Counter Driver Files	36	✓	✓	✓	✓	—
<b>Hardware Example Designs</b>							
Mar 2007	Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs	37	✓	✓	✓	✓	✓
<b>Software Examples</b>							
Nov 2009	Hardware Tutorial Software Example Hangs on Some Boards	38	✓	✓	✓	✓	✓
Nov 2008	Simple Socket Server Example Displays Incorrect IP Address	38	✓	✓	✓	✓	✓
Jun 2006	Networking Examples	39	✓	✓	✓	✓	✓
<b>Flash Programmer</b>							
Feb 2010	Unable to Configure FPGA from Flash with Parallel Flash Loader	39	Fixed	Fixed	Fixed	✓	✓
	Boot Failure or elf2flash Error Due to Incorrect Flash File Size	40	✓	✓	✓	✓	✓
May 2007	Delay When Creating New Flash Programmer Configuration	41	✓	✓	✓	✓	✓
Dec 2006	elf2flash File Size Limit	41	✓	✓	✓	✓	✓
	Error Message “No such file or directory” When Programming Flash	41	✓	✓	✓	✓	✓
<b>Hardware Simulation</b>							
May 2010	Vectored Interrupt Controller Does Not Support VHDL Simulation Models	41	✓	✓	✓	✓	—
Dec 2006	Error “UNC paths are not supported” Launching ModelSim	42	✓	✓	✓	✓	✓
Jun 2006	Uninitialized .bss Variables in Simulation	42	✓	✓	✓	✓	✓
<b>Host Platform</b>							
July 2010	Nios II EDS Requires Administrative Permissions on Windows Vista	42	✓	✓	✓	✓	—
Nov 2009	Cannot Launch Nios II SBT for Eclipse on Red Hat Linux	43	✓	✓	✓	✓	—
Dec 2007	Windows/Cygwin: Nios II Processor Generation Failure	43	✓	✓	✓	✓	✓
Oct 2007	Windows Vista: Limited Support in Nios II EDS	44	✓	✓	✓	✓	✓
<b>Nios II IDE</b>							
<b>Building Projects</b>							
Mar 2009	Nios II IDE Command-Line Tools Select Wrong Workspace on Linux	44	✓	✓	✓	✓	✓
	Nios II IDE Command-Line Tools Hang on Windows	45	✓	✓	✓	✓	✓
Nov 2008	Nios II IDE Cannot Find stdio.h in Outline View	46	✓	✓	✓	✓	✓

**Table 3.** Nios II EDS Errata (1) (Part 4 of 5)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
Jul 2008	Nios II IDE Hangs With UNC Project Path	46	✓	✓	✓	✓	✓
	Build Failure with Nios II Advanced Exceptions, MMU, and MPU	46	✓	✓	✓	✓	✓
May 2007	Build Command Not Functional for BSPs Created With the Nios II SBT	46	✓	✓	✓	✓	✓
	Linker Errors with Dual-Port Memories	47	✓	✓	✓	✓	✓
	User-Managed BSP Settings Not Supported in Nios II IDE	47	✓	✓	✓	✓	✓
	URL Project Location Causes Project Creation Error	47	✓	✓	✓	✓	✓
Dec 2006	Compilation Error with Separate Exception Stack Option	48	✓	✓	✓	✓	✓
	Incorrect Stack and Heap Free Space Report	48	✓	✓	✓	✓	✓
	Nios II IDE Reports Problems Without Displaying Error in Console	48	✓	✓	✓	✓	✓
<b>Debugging Projects</b>							
Nov 2009	Method for Accessing MMU and MPU Registers in the Debugger	49	✓	✓	✓	✓	✓
Nov 2008	IDE Cannot Display Imported Profiling Data	49	✓	✓	✓	✓	✓
Jul 2008	Error Message when Downloading .elf File	50	✓	✓	✓	✓	✓
	Variable Casting Unsupported in ISS	50	✓	✓	✓	✓	✓
	Trace Debug Does Not Support Instruction-Related Exceptions	50	✓	✓	✓	✓	✓
Oct 2007	Trace Debug Does Not Support the JMPI Instruction	50	✓	✓	✓	✓	✓
May 2007	Cannot Locate Source Code in Driver Files Shared by Multiple Projects	51	✓	✓	✓	✓	✓
	Console Window Is Not Updated After ISS Error	51	✓	✓	✓	✓	✓
	ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components	51	✓	✓	✓	✓	✓
	Memory Window Sets Control Register Values Incorrectly	52	✓	✓	✓	✓	✓
Dec 2006	Programs That Interact With a Terminal Console on Windows Do Not Work	52	✓	✓	✓	✓	✓
	“Run as ModelSim” in the Nios II IDE Fails	52	✓	✓	✓	✓	✓
	The Restart Command on the Run Menu Does Not Work	52	✓	✓	✓	✓	✓
Jun 2006	Watchpoints Do Not Work on Certain Variables	53	✓	✓	✓	✓	✓
<b>Navigating Projects</b>							
Dec 2006	Nios II IDE Freezes While Displaying the Splash Screen	53	✓	✓	✓	✓	✓
	Internal Error When Double-Clicking on a Large Objdump File	53	✓	✓	✓	✓	✓
Jun 2006	C/C++ Scanner Does Not Support Certain C/C++ Constructs	54	✓	✓	✓	✓	✓
<b>Nios II SBT</b>							
July 2010	Error Running Nios II Project: ‘Downloading ELF Process failed’	54	✓	✓	✓	✓	—
May 2010	BSP Editor Does Not Show Command-Line Help	55	✓	✓	✓	✓	—
Feb 2010	BSP Not Updated for Memory Size Changes in SOPC Builder	55	✓	✓	✓	✓	—

**Table 3.** Nios II EDS Errata (1) (Part 5 of 5)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
Nov 2009	Build Errors on Interrupt API Calls	55	✓	✓	✓	✓	—
	BSP Editor Incorrectly Allows Mixed Drivers	56	—	—	—	Fixed	✓
	nios2-bsp --help Displays Incorrect List of BSP Types	57	—	—	—	Fixed	✓
	Java Heap Space Error When Creating BSP	57	—	—	—	—	Fixed
Oct 2007	SBT Fails if Nios II EDS is Installed in a Path Containing Spaces	58	✓	✓	✓	✓	✓
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	47	✓	✓	✓	✓	✓
	Upper-Case File Extensions Not Supported	58	✓	✓	✓	✓	✓
<b>Peripherals</b>							
May 2010	Vectored Interrupt Controller Does Not Support VHDL Simulation Models	41	✓	✓	✓	✓	—
Jul 2008	Unaligned Transfers of Small Payloads Fail on SG-DMA	59	✓	✓	✓	✓	✓
May 2007	DMA Controller Always Busy in Burst Mode	59	✓	✓	✓	✓	✓
	Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core	59	✓	✓	✓	✓	✓
<b>Target Software</b>							
Nov 2009	NicheStack tcp_wake() Function Does Not Awaken All select() Sleepers	60	✓	✓	✓	✓	—
	EIC-Based Nios II System Crashes With NicheStack	60	✓	✓	✓	✓	—
Mar 2009	Gigabit Ethernet Performance Issues	60	—	—	—	—	Fixed
Dec 2006	stdio Does Not Work with MicroC/OS-II and Small C Library	61	✓	✓	✓	✓	✓
Jun 2006	cout From MicroC/OS-II Task Does Not Send Data to stdout	61	✓	✓	✓	✓	✓
	malloc(), realloc() Failures With MicroC/OS-II	61	✓	✓	✓	✓	✓
<b>Toolchain (gcc, gdb, etc.)</b>							
July 2010	C++ Exceptions Unsupported in a Multi-Threaded Environment	62	✓	✓	✓	✓	✓
	C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain	62	✓	—	—	—	—
	memcpy() Optimization Misalignment	63	✓	✓	✓	✓	✓
Feb 2010	Unable to Configure FPGA from Flash with Parallel Flash Loader	39	—	—	Fixed	✓	✓
Dec 2006	GNU Assembler Does Not Accept the --defsym Flag	63	✓	✓	✓	✓	✓

**Notes to Table 3:**

- (1) Refer to the Altera Knowledge Database for older errata and solutions.
- (2) For further information, refer to the *Nios II C2H Compiler User Guide*.

## Nios II SBT for Eclipse Errata

This section describes in detail the issues related to the Nios II SBT for Eclipse.

### Nios II SBT for Eclipse Cannot Step into GCC 3 Library Source Code

If your project uses the Nios II GCC 3 toolchain, when debugging with the Nios II SBT for Eclipse, you might be unable to step into library source code. A message such as the following message appears:

```
Resource '/xxx/xxx.c' does not exist.
```

#### Workaround

Upgrade to the Nios II EDS v10.0 or later and use the Nios II GCC 4 toolchain.

#### Solution Status

Not fixed

### Error Building Imported Project: 'No rule to make target'

If an application or library project is created with v9.1 SP2 or earlier (using the GCC 3 toolchain), and you import it to the v10.0 SBT for Eclipse using the GCC 4 toolchain, you might see the build error shown in [Example 1](#).

#### Example 1.

---

```
make all
Info: Building ../hw_bsp
make --no-print-directory -C ../hw_bsp
make[1]: *** No rule to make target `/cygdrive/c/tmp/
verilog_niosII_stratixII_2s60_standard/software/hw_bsp/
alt_sys_init.c', needed by `all'. Stop.
make: *** [../hw_bsp-recurs-make-lib] Error 2
```

---

This error can also happen if you import a GCC 3 project to the v10.0 SBT for Eclipse using the GCC 3 toolchain, and later change the toolchain to **MinGW Nios II GCC4**.

This issue only affects platforms running the Windows operating system.

## Workaround

To avoid this error, follow these steps:

1. Edit the application project and library project makefiles and add the italicized code as shown in [Example 2](#).

### Example 2.

---

```
#-----
# The adjust-path macro
#
# If COMSPEC is defined, Make is launched from Windows through
# Cygwin. This adjust-path macro will call 'cygpath -u' on all
# paths to ensure they are readable by Make.
#
# If COMSPEC is not defined, Make is launched from *nix, and no adjustment
# is necessary
#-----

# Beginning of inserted code
ifndef COMSPEC
ifdef ComSpec
COMSPEC = $(ComSpec)
endif # ComSpec
endif # !COMSPEC
#end of inserted code

ifdef COMSPEC
adjust-path = $(shell cygpath -u "$1")
adjust-path-mixed = $(shell cygpath -m "$1")
else
adjust-path = $1
adjust-path-mixed = $1
endif
```

---

2. Right-click the BSP project, point to **Nios II** and click **Generate BSP**.
3. Clean and rebuild the application and library projects.

## Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Error Building Imported Project: 'target pattern contains no %'

If your application or library makefile uses an absolute path and is generated with the GCC 3 toolchain, and you import it to the v10.0 SBT for Eclipse using the GCC 4 toolchain, you get the following build error:

```
makefile:587: *** target pattern contains no '%'. Stop.
```

## Workaround

Recreate the project using the SBT for Eclipse v10.0.

Alternatively, use the GCC 3 toolchain.

## Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Missing Nios II Perspective

If you launch the Nios II SBT for Eclipse from a universal naming convention (UNC) path, the Nios II perspective does not appear in Eclipse.

This issue appears only on platforms running the Windows operating system.

### Workaround

Do not use a UNC path to launch the Nios II SBT for Eclipse. For example, if the SBT for Eclipse is installed on a network drive, map the network path to a Windows drive letter.

### Solution Status

Not fixed

## Error Debugging Imported Project: 'Can't find a source file'

If you create a project using the Nios II SBT in the Nios II Command Shell, and then import the project to Nios II SBT for Eclipse for debugging, the debugger cannot find source files located outside of project folder.

Example of error message:

```
Can't find a source file at "../src/test.c"
```

### Workaround

To link the affected source files to the project, follow these steps:

1. Right-click the project and click **Properties** to open the **Properties** dialog box.
2. Click + to expand **C/C++ General**, and click the **Paths and Symbols** tab.
3. Click the **Source Location** tab.
4. Click **Link Folder** and navigate to the folder containing the source file.

### Solution Status

Fixed in v10.0 of the Nios II EDS

## Error Marker Persists on BSP Project After Build Error Corrected

In the Nios II SBT for Eclipse, if an application project fails to build, error markers appear in the BSP project, if the build error message contains a path to BSP project source files. Even after you have rebuilt the application project without errors, the error markers persist in the BSP project.

### Workaround

Rebuild your BSP project.

Alternatively, if your application will always be associated with the same BSP, set the Eclipse project references by following these steps:

1. Right-click your application project, point to **Properties** and click **Project References**.
2. Select the desired BSP for your application.
3. Build your application project again to clear the error markers in the BSP project.



If you later link your application project with a different BSP, you must manually correct the Eclipse project references.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Assembly Language Source File Not Found

The Nios II SBT for Eclipse does not support assembly language files with the upper case file extension `.S`. If your project includes a file with an upper case extension, the compiler is not able to find the assembly language file during build.

### Workaround

Rename the assembly language files with the lower-case extension `.s`.

### Solution Status

Fixed in v10.0 of the Nios II EDS

## 'Exclude from build' Not Supported

In the Nios II SBT for Eclipse v9.1 SP2 and earlier, you can use the **Exclude from build** command to omit a source file from your project at build time. Starting with v10.0, **Exclude from build** is replaced by the **Remove from Nios II build** and **Add to Nios II build** commands.

Due to an Eclipse platform limitation, **Exclude from build** still appears in the project context menu in v10.0. However, it has no effect.

### Workaround

To exclude a source file from your project when it is built, on the project context menu, click **Remove from Nios II build**. To restore it to the build, click **Add to Nios II build**.

## Errors Creating or Importing Software Projects

If you place software projects in your Eclipse workspace directory, you might experience project-related errors. For example, you might see one of the following error messages:

- Unable to create project  
Reason: Unable to create project in workspace directory
- Failed to import  
Reason: Unable to import project

### Workaround

Switch to a different workspace, or create a new workspace, separate from your project files. To switch workspaces or create a workspace, on the File menu, click **Switch Workspace**.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Nios II Options Do Not Appear in Eclipse

When Nios II SBT for Eclipse starts, the Nios II plugins might fail to load, resulting in the following symptoms:

- The Nios II perspective is not available
- No Nios II items appear when you click **New** in the File menu

Eclipse can behave this way if you have installed the Nios II EDS in the same directory as a previous installation.

When the Altera Complete Design Suite (ACDS) installer installs the Nios II EDS, you specify a root directory for the EDS file structure. For example, if the Nios II EDS 9.1 is installed on the Windows operating system, the root directory might be `c:\altera\91\nios2eds`. For simplicity, Altera documentation refers to this directory as *<Nios II EDS install path>*.

When the ACDS installer removes the Nios II EDS, it leaves behind some files in *<Nios II EDS install path>*. If you reinstall the Nios II EDS in the same directory, these leftover files might cause the Nios II SBT for Eclipse to work incorrectly.

### Workaround

To correct this problem, execute the following steps:

1. Uninstall the Nios II EDS.
2. Delete *<Nios II EDS install path>*, including all subdirectories, before reinstalling the Nios II EDS.

Alternatively, reinstall the Nios II EDS in a different directory.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Build Errors on Software for Pre-Existing Design with SG-DMA

If you have a Nios II system generated with SOPC Builder v. 9.0 or earlier, and it contains the Scatter-Gather DMA (SG-DMA) component, you cannot build software for it with the Nios II SBT for Eclipse v. 9.1 or later. If you attempt to do so, the Nios II compiler reports errors.

### Workaround

Regenerate the design with SOPC Builder v. 9.1 or later before attempting to compile the software in the Nios II SBT for Eclipse.

## Solution Status

Not fixed

## Spurious System ID Mismatch Error

You might see the following error message when attempting to run or debug a Nios II software project, or to edit a launch configuration:

```
[Target Connection]:System ID mismatch - actual: "0xffffffff", expected: "<system id>"  
<system id> is replaced by the system ID in your software project. When this error  
occurs, you cannot run or debug software.
```

This error normally means that the FPGA is unconfigured, or configured with a `.sof` file that does not correspond to your software project. However, the error occasionally appears even when the FPGA is configured with the correct `.sof`.

## Workaround

After verifying that the FPGA is configured with the correct `.sof`, you can use one of the following methods to work around this issue:

- In the **Run Configuration** or **Debug Configuration** dialog box, on the **Target Connection** tab, click **Refresh Connections** repeatedly until the error message disappears.
- Delete the launch configuration and create a new launch configuration.

## Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Errors Debugging as Local C/C++ Application

If you try to debug a Nios II software project as the Local C/C++ Application launch configuration type, you see one of the following errors, depending on your platform:

- Windows: "Error creating session"
- Linux: "Execution is suspended because of error"

The Nios II debug perspective fails to open.

This is expected CDT behavior in the Eclipse platform. Local C/C++ Application is the launch configuration type for a standard CDT project. To invoke the Nios II plugins, you must use a Nios II launch configuration type.

## Workaround

Always debug Nios II software projects as the Nios II Hardware launch configuration type.

## Error Messages on Console When Debugging

When debugging, you might see the following messages in the Nios II SBT for Eclipse console:

```
No symbol "auto" in current context.  
No symbol "new" in current context.
```

These are normal messages from the GNU Debugger (GDB) console, not error messages. This issue does not impact the debug process.

**Workaround**

No workaround needed.

**Solution Status**

None needed.

**java.lang.NullPointerException Error When Running Application**

When you attempt to run or debug an application, the Nios II SBT for Eclipse might show an internal error message similar to the following:

```
Launching New_configuration java.lang.NullPointerException
```

This error occurs in either of the following circumstances:

- You attempt to run your project without rebuilding after performing a `make clean`.
- You delete a run configuration belonging to your project while the project is running.

You must restart Nios II SBT for Eclipse to recover from this error.

**Workaround**

Build the project before attempting to run or debug it. Do not delete a run configuration while it is in use.

**Solution Status**

This issue will be fixed in a future release of the Nios II EDS.

**Update Failure When Setting Empty Properties in the Nios II Properties Page**

If you use the Nios II Properties page to update more than one board support package (BSP) setting to the empty string (" "), the BSP is not updated.

**Workaround**

You can use either of the following workarounds to set multiple BSP settings to the empty string:

- Use the keyword `none` instead of " " to set settings to the empty string.
- Use the BSP Editor instead of the Nios II Properties page.

**Solution Status**

This issue will be fixed in a future release of the Nios II EDS.

## Errors Converting Nios II IDE Multiprocessor Project

If you try use the `nios2-convert-ide2sbt` utility to convert an IDE multiprocessor project to the Nios II SBT, error messages appear, and the project fails to generate properly.

### Workaround

Create new Nios II SBT application and BSP projects using project settings equivalent to the original Nios II IDE project settings.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## DHCP Time-Out on EP3C120

If you create a project based on the web server template targeting the Altera Embedded Systems Development Kit, Cyclone® III Edition (EP3C120) board, the system is unable to get an IP address at startup. DHCP times out and the system reverts to the default IP address.

### Workaround

Create the project in the Nios II Command Shell and then import it to the Nios II SBT for Eclipse, by executing the following steps:

1. In the Nios II Command Shell, navigate to the following path:


```
<Nios II EDS install path>/examples/<language>/niosII_cycloneIII_3c120/tserd/  
software_examples/app/web_server_tse_3c120_rgmii
```

where *<language>* is either `vhdl` or `verilog`.

2. Type the following command to create a web server example for the EP3C120 kit:

```
create-this-app↵
```

3. Import the command-line project to the Nios II SBT for Eclipse.

 For information about importing command-line projects, refer to “Importing a Command-Line Project” in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer’s Handbook*.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## No Nios II Multiprocessor Collection Run Configuration

The multiprocessor collection run configuration is not available in the Run menu. Therefore, you cannot use a single run configuration to download Executable and Linking Format Files (.elf) to run on multiple Nios II processors in the Nios II SBT for Eclipse.

### Workaround

Create a separate run configuration for each Nios II processor in the hardware design, and launch them one at a time

### Solution Status

Not fixed

## Stop on Startup Option in Run Configuration Has No Effect

Turning on **Stop on Startup** at in the **Debugger** tab of a run configuration has no effect.

### Workaround

You can manually set any breakpoint in the debug perspective.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Project Imported From Command Line Fails to Build Correctly

If you try to import a command-line project to the Nios II SBT for Eclipse, and your project has source files located outside the project folder, the Nios II SBT for Eclipse fails to build your project because the makefile does not include source files located outside the project folder.

### Workaround

You can work around this issue with either of the following methods:

- When importing the project, disable source management through the **Nios II Application Properties** or **Nios II Library Properties** dialog box.



You must manually keep your makefile updated to reflect source files added to or removed from the project.

- Manually link the affected source files to the project. Right-click the project, click **Properties**, expand **C/C++ General**, select **Paths and Symbols** and then **Source Location**, and click **Link Folder**.

### Solution Status

This issue will be fixed in future release of the Nios II EDS.

## Debugger Breaks in crt0.s Instead of main()

If you attempt to launch a debug session by right clicking the project, without first creating a run configuration, the debugger breaks in the initialization code in **crt0.s**, rather than at the top of the **main()** function. You must explicitly create the run configuration before launching the debug session.

### Workaround

You can work around this issue with either of the following methods:

- Create the run configuration with the **Run Configuration** dialog box. You can then debug with this run configuration by right-clicking the project, clicking **Debug As**, and selecting the debug configuration. The program correctly breaks at `main()`.
- Manually set a breakpoint at `main()`.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## C2H Compiler Errata

This section describes in detail the issues related to the Nios II C2H Compiler.

### Error "java.lang.NoClassDefFoundError" During C2H Compilation

C2H compilation fails with the following error:

```
java.lang.NoClassDefFoundError: com/altera/c2h/C2Hgenerate  
Exception in thread "main" make: *** [c2h_hdl-t] Error 1
```

This error occurs when you invoke C2H from the Nios II IDE on the Windows operating system. This problem does not affect Linux installations, nor C2H compilation from the Nios II command shell.

### Workaround

To correct the problem, visit [mySupport](#) and request the Nios II EDS 8.1 Patch 0.01 (PC version).

### Solution Status

Fixed in v. 9.0 of the Nios II EDS

### Functions Declared Without a Return Type Are Not Supported

The C2H compiler does not support functions without an explicitly declared return type.

### Workaround

If you are using the implicit `int` return type, declare the return type explicitly. If your function has no return value, declare it as `void`.

### Solution Status

Not fixed

### Pre-7.1 Systems Are Not Supported

Starting in v. 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder 7.0 or earlier (systems based on a `class.ptf` file).

**Workaround**

Before using the C2H Compiler, you must upgrade your SOPC Builder system. Open the system in SOPC Builder v. 7.1 or later. SOPC Builder prompts you to convert the system, creating a \*\_hw.tcl file compatible with the C2H Compiler.

**Solution Status**

Not fixed

**--src-dir SBT Argument Does Not Work With C2H**

When using the C2H Compiler with Nios II SBT, an error occurs when using the `nios2-app-generate-makefile` argument `--src-dir` to specify a directory of source files. An example of the error is:

```
multiple definition of 'my_accelerated_function'
```

**Workaround**

Specify source files individually using the `--src-files` argument.

**Solution Status**

Not fixed

**Accelerator Generation Failure If Tools Are Installed in Path With Spaces**

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

**Workaround**

Reinstall the tools to a path containing no spaces.

**Solution Status**

This issue will be fixed in a future release of the Nios II EDS.

**The C2H Compiler Regenerates an Accelerator Unnecessarily**

The C2H Compiler might regenerate an accelerator even when the accelerated function is unchanged. This problem can result from changes to files included by the C file containing the accelerated function. The C2H Compiler fails to check that the generated hardware description language (HDL) matches the previously generated HDL causing the system to be regenerated.

**Workaround**

To avoid this issue move the accelerator-specific information from the include file to a separate include file. This workaround prevents regeneration of the system when the HDL is unchanged.

**Solution Status**

Not fixed

## Error: c2h\_accelerator\_base\_addresses.h: No such file or directory

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected:

```
c2h_accelerator_base_addresses.h: No such file or directory.
```

### Workaround

Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Java Heap Space Exception if Quartus II Compilation is Enabled

If your design contains a C2H accelerator, and you select **Build software, generate SOPC Builder system, and run Quartus II compilation** in the C2H view, you might see the following error during Quartus® II compilation:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
make: *** [c2h_hdl-t] Error 1
```

### Workaround

Select the **Build software and generate SOPC builder system** option in the C2H Compiler settings window, and then manually launch the Quartus II software to compile the design.

### Solution Status

Not fixed

## Pointer Dereferences to Volatile Types

The C2H Compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way.

The two loops in [Example 3](#) cannot be scheduled concurrently because the volatile qualification overrides the `__restrict__` pragma.

### Example 3. Non-Concurrent Loops

```
volatile int * __restrict__ fifo_rd = FIFO_RD_BASE;
volatile int * __restrict__ fifo_wr = FIFO_WR_BASE;
for ()
{
    *fifo_wr = ....;
}
for ()
{
    ... = *fifo_rd;
}
```

**Workaround**

Divide the function into multiple interrupt request (IRQ)-enabled accelerators that are launched concurrently from the processor, and use FIFO buffers to communicate between them.

**Solution Status**

Not fixed

**C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File**

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

**Workaround**

Include all subfunctions called by the accelerated function within the same source code file.

**Solution Status**

Not fixed

**Array Elements in Structures Do Not Copy Correctly**

C2H accelerators do not correctly copy array elements that are elements of structures.

In [Example 4](#), the a and b elements of the structure copy correctly, but the buf element does not. After this assignment, struct\_a equals {9, 8, {3, 3, 3, 3}}.

**Example 4. Array Elements of Structs**


---

```
typedef struct my_struct {
    int a;
    int b;
    int buf[BUF_SIZE];
}MY_STRUCT;
MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};
struct_a = struct_b;
```

---

**Workaround**

Copy the array elements explicitly, as shown in [Example 5](#).

**Example 5. Copying Array Elements Explicitly**


---

```
{
    int i=0;
    do
    {
        struct_a.buf[i] = struct_b.buf[i];
        i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

---

### **Solution Status**

This issue will be fixed in a future release of the Nios II EDS.

## **Clean Build Causes Build Failure**

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

### **Workaround**

Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

### **Solution Status**

This issue will be fixed in a future release of the Nios II EDS.

## **Changing Build Configurations Produces Unexpected Results**

The C2H Compiler does not support multiple build configurations (for example Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

### **Workaround**

For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

### **Solution Status**

Not fixed

## **Hardware Accelerators Remain After Deleting the Software Project**

If a system contains C2H accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

### **Workaround**

To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

### **Solution Status**

Not fixed

## Incorrect Results From Logical or Conditional Operation With Side-Effects

The C2H Compiler always evaluates both operands of logical (&&, ||) and conditional (? :) operators. This is different from expected American National Standards Institute (ANSI) C behavior, where operands are evaluated left-to-right, and unnecessary operands are skipped.

For example, in the expression `(i-- && j--)`, if the value of `i` is zero, ANSI C does not evaluate the right-hand-side (RHS) expression, and `j` is not decremented. By contrast, the C2H Compiler evaluates both sides, decrementing `j`.

The following expressions are other examples that might be affected by this issue:

```
if (i-- || j++)
    ...;

a = ((cond == 1)? i++ : j++);
```

### Workaround

Use logical and conditional operations whose operators have no side effects. Operations with side effects include pre- and post- increment and decrement operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

### Solution Status

Not fixed

## Launch SOPC Builder Button in C2H View

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click **Launch SOPC Builder** in the C2H view.

### Workaround

Launch SOPC Builder from the Quartus II software. Alternatively, keep only one system library project open at a time while using the C2H Compiler.

### Solution Status

Not fixed

## Development Board Errata

This section describes in detail the Nios II EDS issues related to Nios development boards.

### Intermittent Failures While Accessing CompactFlash Card

The Nios II Development Kit version 5.0 and later includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. For True IDE mode to operate, CompactFlash cards require that the `ATASEL_N` input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register, used to cycle power to CompactFlash cards from Nios II software through a metal oxide semiconductor field-effect transistor (MOSFET) on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the ATASEL\_N pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the ATASEL\_N pin when power is first applied to the development board, when I/O is not yet driven by the FPGA (before FPGA configuration).

### Workaround

If you encounter errors with CompactFlash when using the Nios development boards, try one of the following solutions:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended if you are familiar and comfortable with board-level modifications. Disconnect pin 9 (ATASEL\_N) on the CompactFlash socket on your Nios development board and tie this pin to ground.



The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True IDE mode operation.

### Solution Status

Not fixed

## Documentation Errata

This section describes in detail the Nios II EDS documentation issues.

### Error Building Project: 'No rule to make target'

The *Getting Started with Graphical User Interface* and *Getting Started from the Command Line* chapters of the *Nios II Software Developer's Handbook* do not include the following information:

You cannot link Nios II GCC 3 projects with Nios II GCC 4 projects. Your application, library and BSP projects must all use the same Nios II GCC toolchain.

If you switch between Nios II GCC 3 and Nios II GCC 4, make sure you run **make clean** on your application, library and BSP projects before rebuilding.

When importing a GCC 3 BSP, such as a project created with v9.1 SP2 or earlier, to v10.0 SBT for Eclipse using the GCC 4 toolchain, after importing the BSP, regenerate the makefile.

If you do not follow these rules, you might see the following error:

```
make[1]: *** No rule to make target `/cygdrive/c/.../bsp/  
alt_sys_init.c', needed by 'all'. Stop.  
make: *** [../bsp/-recurs-make-lib] Error 2
```

### Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

## AN543 Contains Incorrect Information about Updating the Flash

“Updating the Flash Option Bits on the Cyclone III Development Board” in AN543, *Debugging Nios II Software Using the Lauterbach Debugger* contains incorrect information about updating the flash.

### Workaround

The correct information is as follows:

#### Updating the Flash on the Cyclone III Development Board

To restore the flash factory image, refer to “Restoring the Factory Design to the Flash Device” in the *Cyclone III Development Kit User Guide*.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

## Incorrect Information about Floating-Point Instruction Precision

In “Arithmetic Logic Unit” in the *Processor Architecture* chapter, the *Nios II Processor Reference Handbook* incorrectly states that the Nios II compiler treats floating-point constants as double-precision numbers.

### Workaround

The correct information is as follow:

All the floating-point custom instructions are single-precision operations. Double-precision floating-point operations are implemented in software.

When the floating-point custom instructions are not present, the Nios II compiler treats floating-point constants as double-precision values. However, with the floating-point custom instructions, the Nios II compiler treats floating-point constants as single-precision numbers by default. This allows all floating-point expressions to be evaluated in hardware, at a possible cost in precision.

If you do not wish floating-point constants to be cast down to single precision values, append `L` to each constant value, to instruct the compiler to treat the constant as a double-precision floating-point value. In this case, if an expression contains a floating-point constant, each term in the expression is cast to double precision. As a result, the expression is computed with software-implemented double-precision arithmetic, at a possible cost in computation speed.

[Table 4](#) shows code examples using floating-point constants, indicating how each computation is implemented.

**Table 4.** Floating-Point Constant Examples

Example Code	Floating-Point Custom Instructions Present?	Precision	Implementation
$b = a \times 4.67$	No	Double	Software
$b = a \times 4.67f$	No	Single	Software
$b = a \times 4.67L$	No	Double	Software
$b = a \times 4.67$	Yes	Single	Hardware
$b = a \times 4.67f$	Yes	Single	Hardware
$b = a \times 4.67L$	Yes	Double	Software

**Solution Status**

Fixed in v10.0 documentation

**Eclipse CDT Features Not Supported by Nios II Plugins**

The following information is missing from the *Getting Started with Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*:

The features listed in the left column of [Table 5](#) are supported by the Eclipse CDT plugins, but are not supported by Nios II plugins. The right column lists alternative features supported by the Nios II plugins.

**Table 5.** Eclipse CDT Features Not Supported by the Nios II Plugins (Part 1 of 2)

Unsupported CDT Feature	Alternative Nios II Feature
<b>New Project Wizard</b> C/C++ <ul style="list-style-type: none"> <li>■ C Project</li> <li>■ C++ Project</li> <li>■ C/C++               <ul style="list-style-type: none"> <li>■ Convert to a C/C++ Project</li> <li>■ Source Folder</li> </ul> </li> </ul>	To create a new project, use one of the following Nios II wizards: <ul style="list-style-type: none"> <li>■ Nios II Application</li> <li>■ Nios II Application and BSP from Template</li> <li>■ Nios II Board Support Package</li> <li>■ Nios II Library</li> </ul>
<b>Build configurations</b> Right-click project and point to <b>Build Configurations</b>	The Nios II plugins only support a single build configuration.
<b>Exclude from Build</b> (from v10.0 onwards) Right-click source files	Use <b>Remove from Nios II Build</b> and <b>Add to Nios II Build</b> . See <a href="#">“Exclude from build’ Not Supported” on page 13</a> .

**Table 5.** Eclipse CDT Features Not Supported by the Nios II Plugins (Part 2 of 2)

Unsupported CDT Feature	Alternative Nios II Feature
<p><b>Project Properties</b></p> <p>C/C++ Build</p> <ul style="list-style-type: none"> <li>■ Builder Settings <ul style="list-style-type: none"> <li>■ Makefile generation</li> <li>■ Build location</li> </ul> </li> <li>■ Behaviour <ul style="list-style-type: none"> <li>■ Build on resource save (Auto build)</li> </ul> </li> <li>■ Build Variables</li> <li>■ Discovery Options</li> <li>■ Environment</li> <li>■ Settings</li> <li>■ Tool Chain Editor <ul style="list-style-type: none"> <li>■ Current builder</li> <li>■ Used tool</li> </ul> </li> </ul> <p>C/C++ General</p> <ul style="list-style-type: none"> <li>■ Enable project specific settings</li> <li>■ Documentation</li> <li>■ File Types</li> <li>■ Indexer <ul style="list-style-type: none"> <li>■ Build configuration for the indexer</li> </ul> </li> <li>■ Language Mappings</li> <li>■ Paths and Symbols</li> </ul>	<p>By default, the Nios II SBT generates makefiles automatically. The build location is fixed.</p> <p>To change the toolchain, use the <b>Current tool chain</b> option</p> <p>The Nios II plugins only support a single build configuration.</p> <p>Use <b>Nios II Application Properties</b> and <b>Nios II Application Paths</b></p>
<p><b>Window Preferences</b></p> <p>C/C++</p> <ul style="list-style-type: none"> <li>■ Build scope</li> <li>■ Build project configurations</li> <li>■ Build Variables</li> <li>■ Environment</li> <li>■ File Types</li> <li>■ Indexer <ul style="list-style-type: none"> <li>■ Build configuration for the indexer</li> </ul> </li> <li>■ Language Mappings</li> <li>■ New CDT project wizard</li> </ul>	<p>The Nios II plugins only support a single build configuration.</p> <p>The Nios II plugins only support a single build configuration.</p>

## Incorrect Information about Nested Exceptions

“Exception Processing” in the *Programming Model* chapter of the *Nios II Processor Reference Handbook* incorrectly states that multiple interrupts with different requested interrupt levels (RILs) must not be assigned to the same shadow register set.

## Workaround

The correct information is as follows:

Multiple interrupts with different RILs can be assigned to a single shadow register set. However, with multiple register sets, you must not allow the RILs assigned to one shadow register set to overlap the RILs assigned to another register set.

Table 2–6 and Table 2–7 illustrate the validity of register set assignments when preemption within a register set is enabled.

**Table 2–6.** Example of Illegal RIL Assignment

RIL	Register Set 1	Register Set 2
1	IRQ0	
2	IRQ1	
3		IRQ2
4	IRQ3	
5		IRQ4
6		IRQ5
7		IRQ6

**Table 2–7.** Example of Legal RIL Assignment

RIL	Register Set 1	Register Set 2
1	IRQ0	
2	IRQ1	
3	IRQ3	
4		IRQ2
5		IRQ4
6		IRQ5
7		IRQ6

## Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

## Missing Documentation of Interrupt API Properties

“Tcl Commands” in the *Nios II Software Build Tools Reference* chapter of the *Nios II Software Developer’s Handbook* does not include information about two `set_sw_property` command properties related to the application programming interfaces (APIs) for interrupts.

## Workaround


The missing information belongs with the `set_sw_property` command, in “Tcl Commands” in the *Nios II Software Build Tools Reference* chapter of the *Nios II Software Developer’s Handbook*.


The missing information follows:

- `supported_interrupt_apis`—Specifies the interrupt API that the device driver supports. The Nios II SBT analyzes this property for each driver in the system to determine the appropriate API to be used in the system.

Specify `legacy_interrupt_api` if the device driver supports the legacy API only or `enhanced_interrupt_api` if the device driver supports the enhanced API only. Specify both using a quoted list if the device driver supports both APIs.

If you do not specify which API your device driver supports, the Nios II SBT assumes that only the legacy interrupt API is supported.

 This property is only available for device drivers.

 For more information about the legacy and enhanced APIs, refer to “Exception Handling” in the *Nios II Software Build Tools* chapter of the *Nios II Software Developer’s Handbook*.

- `isr_preemption_supported`—Specify `true` if your device driver’s interrupt service routine (ISR) can be preempted by a higher priority ISR. If you do not specify whether the driver supports preemption, the Nios II SBT assumes that your device driver does not support preemption. If your driver does not have an ISR, but the associated device has an interrupt port, you can set this property to `true`.

 This property is valid for operating systems and device drivers.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

## Valid Range of `hal.log_flags` is -1 to 3

The Nios II EDS documentation incorrectly states the valid range for the `hal.log_flags` BSP setting. Valid values of `hal.log_flags` range from -1 through 3.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

## NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP

The Nios II documentation should include the following information:

Versions of the NicheStack TCP/IP Stack other than the Nios II Edition include optional Trivial File Transfer Protocol (TFTP) client and server applications. However, the TFTP client and server are not thread safe. They cannot be used in systems with the MicroC/OS-II RTOS.

TFTP is not available with the NicheStack TCP/IP Stack - Nios II Edition. The TFTP client and server features are disabled.

### Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

## Error Message After Renaming Project: “Resource is out of sync with the system”

The *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer’s Handbook* does not explain the correct procedure for renaming a project.

### Workaround

The missing information is as follows:

To rename a project in the Nios II SBT for Eclipse, execute the following steps:

1. Right click the BSP project and click **Rename**.
2. Type the new BSP name.
3. Right click the BSP project and click **Refresh**.



If you neglect to refresh the project, you might see the following error message when you attempt to build it:

```
Resource <original_bsp_name> is out of sync with the system
```

### Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

## Obsolete Wiki URLs in Help Systems and Handbook

The following documents contain obsolete URLs for the Nios Community Wiki:

- The *Nios II Software Developer’s Handbook*
- The *Nios II Embedded Design Handbook*
- The Nios II IDE help system

### Workaround

You can visit the Nios Community Wiki at [www.nioswiki.com](http://www.nioswiki.com).

### Solution Status

Fixed in the documentation for v. 9.0

## Missing Source Control Information

The Quartus II Handbook refers to the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer’s Handbook* for information about putting Nios II projects in source control. The referenced chapter only describes the process for Nios II SBT projects.

## Workaround

The missing information follows.

### Archiving Nios II IDE Software Projects

This section helps you identify the files you must include when archiving a Nios II IDE software project. With this information, you can archive a Nios II application project and its associated Nios II system library project.

You might want to archive your projects for one of the following reasons:

- To place them under source control
- To create backups
- To bundle the projects for transfer to another location

This section covers the following information:

- How to find and identify the files that you must include in an archived Nios II IDE software project.
- Which files must have write permission to allow the software projects to be built.

Table 6 and Table 7 list the files required by Nios II IDE software projects. This is the minimum set of files needed to completely rebuild a software project, including the `.elf`.

Archive your Nios II IDE software projects together with the SOPC Builder system on which they are based. You cannot rebuild a Nios II IDE software project without its associated SOPC Builder system.

The files listed in Table 6 are located in the Nios II application project directory.


**Table 6.** Files Required for a Nios II Application Project

File Description	File Name	Write Permission Required?
All source files	for example: <code>app.c</code> , <code>header.h</code> , <code>assembly.s</code> , <code>lookuptable.dat</code>	No
Eclipse project file	<code>.project</code>	No
C/C++ Development Toolkit project file	<code>.cdtproject</code>	Yes
C/C++ Development Toolkit option file	<code>.cdtbuild</code>	No
Software configuration file	<code>application.stf</code>	No

The files listed in Table 7 are located in the Nios II system library project directory.

**Table 7.** Files Required for a Nios II System Library Project

File Description	File Name	Write Permission Required?
Eclipse project file	<code>.project</code>	Yes
C/C++ Development Toolkit project file	<code>.cdtproject</code>	Yes
C/C++ Development Toolkit option file	<code>.cdtbuild</code>	No
System software configuration file	<code>system.stf</code>	Yes

 You must have write permission for certain files. The tools write to these files as part of the build process. If the files are not writable, the toolchain fails. Many source control tools mark local files read-only by default. In this case, you must override this behavior. You do not have to check the files out of source control unless you are modifying the Nios II software project.

### Solution Status

In the documentation for v. 9.0, the missing information appears in the *Nios II Integrated Development Environment* chapter of the *Nios II Software Developer's Handbook*. In the documentation for v. 9.1, the missing information appears in the *Using the Nios II Integrated Development Environment* appendix of the *Nios II Software Developer's Handbook*.

## Compiler Flags for Building Custom Newlib

"Common BSP Tasks" in the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* describes how to compile a custom version of the newlib library. However, it does not list the compiler flags that must be selected to generate the correct library.

### Workaround


The missing information is as follows:

The Nios II EDS provides a number of precompiled newlib implementations. The provided libraries are precompiled with every viable combination of the GCC compiler flags shown in [Table 8](#).

When you create a BSP with a precompiled newlib, the Nios II SBT selects the newlib matching your BSP's compiler settings. When you create a custom newlib, you must ensure that the compiler flags listed in [Table 8](#) match your BSP's settings.

**Table 8.** GCC Compiler Options for Newlib

Option Name	Purpose
-pg	Link for profiling with gprof
-EB	Generate big-endian code
-mstack-check	Enable stack checking
-mno-hw-mul	Disable use of the mul family of instructions
-mhw-mulx	Enable use of the mulx family of instructions
-mcustom-fpu-cfg	Use a floating point custom instruction

 The Nios II-specific compiler flag `-mcustom-fpu-cfg` requires the Nios II processor core to be implemented with the floating-point custom instruction as specified in the flag argument. The compiler flag can have one of the following two argument values:

- `-mcustom-fpu-cfg=60-1`—Use the Nios II floating-point custom instruction without divider
- `-mcustom-fpu-cfg=60-2`—Use the Nios II floating-point custom instruction with divider

- For details about the Nios II-specific compiler flags `-mstack-check`, `-mno-hw-mul`, and `-mhw-mulx`, refer to “Altera Nios II Options” in *Using the GNU Compiler Collection (GCC)*, installed with the Nios II EDS. For details about the `-pg` compiler flag, refer to “Compiling a Program for Profiling” in *GNU Profiler*, installed with the Nios II EDS. To find the installed documentation, on the Windows Start menu, click **Programs > Altera > Nios II EDS <version> > Nios II <version> Documentation**, and then click **Literature**.

### Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

## Error Message on Invalid Exception or Reset Vector

Nios II compiler error messages, added in v 8.1, are not documented in the *Nios II Software Developer's Handbook*.

### Workaround

The missing information is as follows:

While building a Nios II software project, the compiler produces an error message if the SOPC Builder system defines an invalid exception or reset vector. [Table 9](#) shows possible error messages and their causes.

In the SOPC Builder system, ensure that the exception and reset vectors conform to all the criteria listed in [Table 9](#).

### Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

**Table 9.** Exception and Reset Vector Error Messages (Part 1 of 2)

Message	Cause
The section mapping ".entry" is not mapped to a memory region.	The .entry section is mapped to a nonexistent memory region.
The section mapping ".exceptions" is not mapped to a memory region.	The .exceptions section is mapped to a nonexistent memory region.
The section mapping ".entry" maps to a memory region "<region name>" which is not the required length "32".	The .entry section is mapped to a memory region that is smaller than 32 bytes.
The section mapping ".entry" maps to a memory region "<region name>" which does not start at the SOPC Design specified address "<base address>".	The .entry section is mapped to a memory region that does not start at the reset vector base address.
The section mapping ".exceptions" maps to a memory region "<region name>" which does not start at the SOPC Design specified address "<base address>".	The .exceptions section is mapped to a memory region that does not start at the exception vector base address.
The SOPC design Nios II reset and exception vector addresses are mapped to the same memory. For HAL the exception address must be at least 32 bytes larger than the reset address.	The .entry section and .exceptions section map to the same memory device, and the base of the .exceptions section is less than 32 bytes above the base of the .entry section.

**Table 9.** Exception and Reset Vector Error Messages (Part 2 of 2)

Message	Cause
The section mapping ".exceptions" mapped device "<memory 1>" is not the same as the SOPC Design specified device "<memory 2>".	The .exceptions section is mapped to a device that is different from the exception vector device specified in the SOPC Builder system.
The section mapping ".entry" mapped device "<memory 1>" is not the same as the SOPC Design specified device "<memory 2>".	The .entry section is mapped to a device that is different from the reset vector device specified in the SOPC Builder system.

## Nios II IDE Online Help Expand Buttons Do Not Work

The expand text arrow and the **Show All** option in the online help do not work in Internet Explorer.

### Workaround

In Internet Explorer, carry out the following actions:

1. Click the **Refresh/Show Current Topic** icon (yellow arrows icon at the top of the search pane). The table of contents appears with the selected topic highlighted.
2. Click the highlighted topic to refresh the browser frame. Expanded text works.

Alternatively, use Firefox 2.0.0.6.

### Solution Status

Not fixed

## Hardware Abstraction Layer Errata

This section describes in detail the Nios II Hardware Abstraction Layer issues.

### Missing Structure Member Errors in `alt_log_printf.c` with Small JTAG UART Driver

In a BSP, if you enable the small JTAG UART driver (`altera_avalon_jtag_uart_driver.enable_small_driver`), and enable Altera logging (`hal.log_port`) using the JTAG UART as the log port, when you build the project, the compiler reports missing structure members in `alt_log_printf.c`. The project fails to build.

### Workaround

To avoid this issue, execute the following steps:

1. In function `alt_log_jtag_uart_print_control_reg()` in `alt_log_printf.c`, insert the italicized code as shown in [Example 8](#).

#### Example 8. ALT\_LOG\_PRINTF() Correction

---

```
ALT_LOG_PRINTF(
    "%s SW CirBuf = %d, HW FIFO wspace=%d AC=%d WI=%d RI=%d WE=%d RE=%d\r\n",
    header,

    /* Beginning of inserted code */
    #ifndef ALTERA_AVALON_JTAG_UART_SMALL
        (dev->tx_out-dev->tx_in),
    #else
        0,
    #endif
    /* End of inserted code */

    space,ac,wi,ri,we,re);
```

---

2. Rebuild the project.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## “unused variable” Warning When Building `alt_main.c`

If you set the `hal.enable_exit` BSP setting to false, when building the project you see the following warning in `alt_main.c`:

```
warning: unused variable 'result'
```

### Design Impact

This warning is harmless.

### Workaround

Set `hal.enable_exit` to true.

Alternatively, you can ignore this harmless warning.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Build Warnings in Performance Counter Driver Files

If your Nios II system contains a performance counter, you might see the following warning when you build a BSP:

```
warning: return makes pointer from integer without a cast
```

This warning appears in `alt_get_performance_counter_base()` in the software file `altera_avalon_performance_counter.c`.

This warning appears when the your BSP uses the Altera performance counter driver, and the C preprocessor symbol `PERFORMANCE_COUNTER_0_BASE` is defined.

This warning is harmless.

### Workaround

In the `alt_get_performance_counter_base()` function in `altera_avalon_performance_counter.c`, add `typecast (void *)` to `PERFORMANCE_COUNTER_0_BASE`.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Hardware Example Design Errata

This section describes in detail the Nios II EDS hardware example design issues.

### Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs

In several example designs targeting the Nios II Cyclone II 2C35 and Nios II Stratix® II 2S60 RoHS development boards, the synchronous static random access memory (SSRAM) and legacy DDR SDRAM clocks are implemented with an incorrect phase-locked loop (PLL) phase shift.

This issue affects the following example designs:

- 2s60 RoHS Triple Speed Ethernet SG-DMA
- 2s60 RoHS Standard
- 2s60 RoHS Full Featured
- 2c35 Triple Speed Ethernet SG-DMA
- 2c35 Standard
- 2c35 Full Featured

This issue also affects any custom 2c35 or 2s60 RoHS design using SSRAM or legacy DDR SDRAM.


The incorrect phase shift might cause timing violations when reading from or writing to SSRAM or legacy DDR SDRAM.

### Workaround

To correct the PLL settings, perform the following steps:

1. Open the standard design in SOPC Builder.
2. Double-click the component instance named `p11` to launch the PLL MegaWizard™ interface.
3. Click **Launch Altera's ALTPLL MegaWizard** to launch the MegaWizard interface.
4. Select the **Output Clocks** page and then the clock page used to generate the SSRAM clock. Depending on the particular design, this might be `clk c1`, `clk c2`, or another clock.

5. Ensure that **Clock phase shift** is  $-3.38$  ns.
6. Click the clock page used to generate the write clock for the legacy DDR SDRAM, if any. Depending on the particular design, this might be `clk c1`, `clk c2`, or another clock.
7. Ensure that **Clock phase shift** is 270 degrees.
8. Click **Finish** to save changes and exit the PLL MegaWizard interface.
9. Click **Finish** to save settings to the PLL instance in SOPC Builder.
10. Regenerate the system in SOPC Builder and recompile in the Quartus II software. Refer to the `readme.txt` file in the design example folder for information on compiling designs containing the DDR controller.

 For further information refer to `ssram_interface_readme.html`, located in the `<Quartus II installation directory>/sopc_builder/documents` folder. This document discusses the SSRAM timing analysis methodology in detail. Additionally, [AN 411: Understanding PLL Timing for Stratix II Devices](#) discusses clock phase shift calculations and assignments for PLLs in Stratix II devices.

### Solution Status

Not fixed

## Software Example Errata

This section describes in detail the Nios II EDS software example issues.

### Hardware Tutorial Software Example Hangs on Some Boards

The `count binary` application used in the tutorial hangs when run on some Altera development boards. The software runs through the binary count once and stops responding.

This issue affects boards that do not have a display to handle the LCD output. The affected boards include the following:

- Embedded Systems Development Kit, Cyclone III Edition (EP3C120)
- Stratix IV GX FPGA Development Kit (EP4SGX230)

### Workaround

Modify line 18 of the `count_binary.h` file, as follows:

```
# define LCD_PRINTF(lcd, args...) /* Do Nothing */
```

Rebuild and run the software again.

### Solution Status

Not fixed

### Simple Socket Server Example Displays Incorrect IP Address

The simple socket server networking software example displays an incorrect IP address for the incoming connection.

### Workaround

There is currently no workaround for this problem. Ignore the IP address of any incoming connection that is displayed on the console.

### Solution Status

Not fixed

## Networking Examples

If you are running a networking software example, you might be asked for a nine-digit number. You are directed to find this number on a sticker on your Nios development board, identified by the prefix **ASJ**. Not all Nios development boards have this sticker.

### Workaround

If your Nios development board does not have a sticker with the letters **ASJ** followed by a nine-digit number, enter a unique nine-digit number when prompted. To avoid network address conflicts, ensure that this number is unique to each Nios board connected to your network.

### Solution Status

Not fixed

## Flash Programmer Errata

This section describes in detail the Nios II EDS issues related to the flash programmer.

### Unable to Configure FPGA from Flash with Parallel Flash Loader

If you program an FPGA image to flash memory on a board that utilizes the Parallel Flash Loader, the FPGA subsequently fails to load from flash memory. This issue arises because the **sof2flash** utility generates a flash file with an incompatible Programmer Object File (**.pof**) bitstream format.

#### Affected Configurations

This issue impacts the following target hardware:

- The Stratix IV GX FPGA Development Kit
- The Arria® II GX FPGA Development Kit
- Any hardware using the Parallel Flash Loader

### Workaround

1. Visit the [MySupport](#) website and request Nios II EDS patch 0.73.

Alternatively, upgrade to the Nios II EDS v. 9.1 SP1 or later.

2. Invoke **sof2flash** with the following options:

- `--pfl`
- `--optionsbits=<option bit address>`

With these options, **sof2flash** generates a flash file with a compatible **.pof** file bitstream format.

### Solution Status

Fixed in v. 9.1 SP1 of the Nios II EDS

## Boot Failure or elf2flash Error Due to Incorrect Flash File Size

The `sof2flash` utility fails to recognize certain devices. As a result, it generates Motorola S-record (**.flash**) files of the wrong size.

With erasable programmable configurable serial (EPCS) memory, the `elf2flash --after` command fails to recognize the **.flash** file size. The resulting error prevents you from programming flash memory.

With Common Flash Interface (CFI) memory, you can program the **.flash** file, but the Nios II system fails to boot.

This error occurs when programming flash memory for the following devices:

- The following Stratix III devices:
  - EP3SL50
  - EP3SL50F484
  - EP3SL50F780
  - EP3SL110
  - EP3SL110F1152
  - EP3SL110F780
  - EP3SE80
  - EP3SE80F1152
  - EP3SE80F780
- All Stratix IV devices
- All Arria II GX devices

### Workaround

Visit the [MySupport](#) website and request Nios II EDS 9.0 SP2 patch 2.01 and Quartus II 9.0 SP2 patch 2.63.

Alternatively, download and install the Altera Complete Design Suite v. 9.1 or later.

**Solution Status**

Fixed in v. 9.1 of the Nios II EDS

**Delay When Creating New Flash Programmer Configuration**

You might experience a delay of several seconds when creating a new Flash Programmer configuration in the Nios II IDE.

**Workaround**

There is no workaround.

**Solution Status**

Not fixed

**elf2flash File Size Limit**

The `elf2flash` utility supports `.elf` files up to approximately 24 MBytes in size. The `elf2flash` utility might fail on files larger than 24 MBytes, with the error message `java.lang.OutOfMemoryError`.

**Workaround**

Lower the number of symbols in your `.elf` file by turning off debug symbols. Alternatively, specify less initialized data in the application.

**Solution Status**

Not fixed

**Error Message “No such file or directory” When Programming Flash**

When programming flash from a project stored in a path containing spaces, you might receive the spurious error message **No such file or directory**. The flash programmer does not correctly handle spaces in the directory path. However, this error is harmless, because flash programming completes successfully.

**Workaround**

None required.

**Solution Status**

Not fixed

**Hardware Simulation Errata**

This section describes in detail the Nios II EDS issues related to hardware simulation.

**Vectored Interrupt Controller Does Not Support VHDL Simulation Models**

SOPC Builder reports errors when you try to generate VHDL simulation files for the Vectored Interrupt Controller.

**Workaround**

There is no workaround.

**Solution Status**

This issue will be fixed in a future release of the Nios II EDS.

**Error “UNC paths are not supported” Launching ModelSim**

If you launch ModelSim® from a working directory that is mapped via a universal naming convention (UNC) path (a path that starts with // instead of drive letter), you receive the following error message in SOPC Builder: **UNC paths are not supported. Defaulting to Windows directory.** This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

**Workaround**

Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

**Solution Status**

Not fixed

**Uninitialized .bss Variables in Simulation**

If your program reads the value of an uninitialized .bss variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being 'x'. This warning appears because when this property is enabled, the code that clears the .bss memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The .bss region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II processor reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing .bss clearing code. The Hardware Abstraction Layer (HAL) code that executes before and after `main()` might use .bss variables, so these warnings might appear even if your application does not use the .bss section.

**Solution Status**

Not fixed

**Host Platform Errata**

This section describes in detail the Nios II EDS issues related to the host development platform.

**Nios II EDS Requires Administrative Permissions on Windows Vista**

On the Windows Vista operating system, the Nios II EDS requires your user account to have administrative permissions.

Attempts to write to a directory other than the user home directory might fail, especially writing to an installation directory. For example, you might be unable to create a software project under an example design directory. Error messages might vary.

### Workaround

Obtain an administrator account on your workstation.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Cannot Launch Nios II SBT for Eclipse on Red Hat Linux

You might be unable to launch Nios II SBT for Eclipse on the Red Hat Linux operating system. This is an issue with the version of XULRunner in Eclipse 3.4. The issue is resolved in XULRunner 1.9.1, which is available with Eclipse 3.5.

### Workaround

Upgrade XULRunner to a newer version. To determine what version of XULRunner you have, type the following command at the command prompt:

```
xulrunner -v
```

To upgrade XULRunner using the **yum** software package manager, type the following command:

```
sudo yum update xulrunner
```

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Windows/Cygwin: Nios II Processor Generation Failure

You might see the following error message when you generate your Nios II system in SOPC Builder:

```
Error: Generator program for module 'cpu_fpoint' did NOT run successfully
```

This error might occur on a Windows-based system when all of the following conditions are true:

- You have installed a version of Cygwin other than the one distributed with the Quartus II software.
- You launch the Quartus II software and SOPC Builder from a Nios II Command Shell.
- You enable the floating point custom instruction (FPCI) in the Nios II processor core.

### Workaround

Launch the Quartus II software and SOPC Builder from the Windows Start menu, and regenerate your system.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Windows Vista: Limited Support in Nios II EDS

The Quartus II software introduces Windows Vista (32-bit and 64-bit) support in version 7.2. However, the Nios II Embedded Design Suite supports only the following platforms:

- Windows 2000
- Windows XP
- Windows XP (64)
- Windows Vista Business Edition (32-bit)
- SUSE 9 (32-bit)
- SUSE 9 (64-bit)
- Red Hat Linux v3.0 (32-bit)
- Red Hat Linux v3.0 (64-bit)
- Red Hat Linux v4.0 (32-bit)
- Red Hat Linux v4.0 (64-bit)

### Workaround

There is no workaround.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Nios II IDE Errata: Building Projects

This section describes in detail the issues related to building projects in the Nios II IDE.

### Nios II IDE Command-Line Tools Select Wrong Workspace on Linux

On Linux systems, the Nios II IDE command-line tools use the default Eclipse project workspace, regardless what workspace you are using in the IDE. The default workspace is at `$SOPC_KIT_NIOS2/eclipse/nios2-ide-workspace-<version>`.

This issue affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

### Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace.

[Example 9](#) shows how to specify a workspace path.

---

#### Example 9. Providing a Non-Default Workspace Location in Linux

---

```
nios2-create-project -data $HOME/myworkspace <other arguments>↵
```

---

### Solution Status

Not fixed

## Nios II IDE Command-Line Tools Hang on Windows

On Windows systems, the Nios II IDE command-line tools are sometimes unable to locate the project workspace. When this happens, the tools might hang.

This issue potentially affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

### Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. The workspace shown in [Example 10](#) is the default workspace used by the Nios II IDE in version 9.0.

---

#### Example 10. Providing a Workspace Location

---

```
nios2-create-project -data c:/altera/90/nios2eds/\  
  bin/eclipse/nios2-ide-workspace-9.0 <other arguments>
```

---

### Solution Status

Not fixed

## Nios II IDE Cannot Find `stdio.h` in Outline View

If you create a new project using the Nios II IDE and try to open the `stdio.h` file from the Outline view before building the project, the IDE displays the error message **No include files were found that matched that name.**

### Workaround

Build the project before attempting to open `stdio.h`.

### Solution Status

Not fixed

## Nios II IDE Hangs With UNC Project Path

In the **New Project** dialog box, if you turn on **Specify Location** and specify the path in UNC form, the IDE might hang.

### Workaround

Map the UNC path to a remote drive which looks like a Windows drive to the Nios II IDE.

### Solution Status

Not fixed

## Build Failure with Nios II Advanced Exceptions, MMU, and MPU

Projects created in the Nios II IDE for an SOPC Builder system containing a Nios II processor configured with advanced exceptions, the memory management unit (MMU), or the memory protection unit (MPU) fail to build, and generate the following error messages:

```
ERROR - Classic build flow for Nios II system library does not support
the Nios II \
    advanced exceptions.
Use the non-classic Nios II Board Support Package instead.
<timestamp> - (SEVERE) generate: java.lang.IllegalStateException: \
    java.lang.IllegalStateException: \
    com.altera.ingenuous.GTFElement.GTFElementErrorException: \
    <error> element in GTF script
make[1]: *** [system_description/../obj/system.h-t] Error 1
make: *** [system_project] Error 2
```

### Workaround

Create the software project with the Nios II SBT command-line development flow. Alternatively, use the Nios II SBT for Eclipse.

### Solution Status

Not fixed

## Build Command Not Functional for BSPs Created With the Nios II SBT

The build option in the Nios II IDE menu does not rebuild BSPs imported to the IDE.

### Workaround

The Nios II SBT for Eclipse can both build and debug projects created on the command line. The Nios II SBT for Eclipse is the preferred tool for debugging Nios II SBT projects.



For information 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*.

In the Nios II IDE, you can build the BSP by building the associated application project.

### Solution Status

Not fixed

## Linker Errors with Dual-Port Memories

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

### Workaround

Assign the same address to both ports of the dual-port memory.

### Solution Status

Not fixed

## User-Managed BSP Settings Not Supported in Nios II IDE

For projects created with the Nios II SBT and imported to the Nios II IDE, the IDE configuration settings have no effect.

For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II SBT projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II SBT projects. The IDE ignores these options during the build process.

### Workaround

Make these settings in the project's makefile.

### Solution Status

Not fixed

## URL Project Location Causes Project Creation Error

When you try to create a new project in an existing workspace, you might see an error dialog box saying:

```
Project cannot be created. Reason: Internal Error
```

This error might occur if the path to any project in the workspace is a URL location, for example `file:/F:/Design`. To view the path, right-click on the project and select **Properties**.

### **Workaround**

Import your existing application and system library projects to a new workspace.

### **Solution Status**

Not fixed

## **Compilation Error with Separate Exception Stack Option**

Choosing the **Use a separate exception stack option** might cause the following compilation error when building a project:

```
VARIABLE %STACK_POINTER%
```

This error occurs if the exception stack is larger than the memory available for it.

### **Workaround**

On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

### **Solution Status**

Not fixed

## **Incorrect Stack and Heap Free Space Report**

The makefile reports an incorrect number of bytes free for the stack and heap, if the heap and stack are in different memory regions.

### **Workaround**

No workaround available.

### **Solution Status**

Not fixed

## **Nios II IDE Reports Problems Without Displaying Error in Console**

When building a project, the Nios II IDE reports problems, but the build output in the console does not contain any errors.

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

### **Workaround**

If the Console output does not contain errors, then the project built correctly. On subsequent builds, the linker step is skipped and the errors do not appear.

### Solution Status

Not fixed

## Nios II IDE Errata: Debugging Projects

This section describes in detail the issues related to debugging projects in the Nios II IDE.

### Method for Accessing MMU and MPU Registers in the Debugger

You might observe an error if you try to read or write an MMU or MPU register via the Nios II Debugger.

#### Workaround

To read an MPU region, execute the following steps:

1. Set region `INDEX` in the `MPUBASE` register.
2. Exit and reenter debug mode, that is, single step.
3. Set the `RD` bit in the `MPUACC` register.
4. Exit and reenter debug mode, that is, single step.
5. Read back `MPUBASE` for pertinent information.
6. Read back `MPUACC` for pertinent information.

For a system with an MMU or MPU, this workaround allows you to read and write the current values of the registers. However, you cannot use it to control MPU regions or MMU TLB entries.

The debug core copies the Nios II processor's register values to its internal memory when the processor enters debug mode. The debug core writes register values back to the processor only when the processor leaves debug mode. Therefore, if you attempt to set an MPU region with several consecutive values, only the last one, when leaving debug mode, is committed to the processor.

### Solution Status

Not fixed

### IDE Cannot Display Imported Profiling Data

If you create a software project in the Nios II SBT command line flow, generate profiling data to `gmon.out` in the Nios II command shell, and then import the project to the IDE, you cannot use the Profiling perspective to view the profiling data. When you attempt to view `gmon.out`, the IDE displays the following error message:

```
nios2-elf-gprof: ../: not in a.out format.
```

#### Workaround

Run `nios2-elf-gprof` from the Nios II command shell to generate a profiler report.

**Solution Status**

Not fixed

**Error Message when Downloading .elf File**

If the **Build Automatically** option is turned on in the IDE, you might get the following error when downloading a **.elf**:

```
"2 [main] sh 5736 fork: child 3892 - died waiting; \  
  for longjmp before initialization, errno 11"
```

The **Build Automatically** option is off by default, as recommended by the Eclipse help system.

**Workaround**

Turn off the **Build Automatically** option and download the **.elf** file again.

**Solution Status**

Not fixed

**Variable Casting Unsupported in ISS**

Casting variables in the Variables view in the Debug perspective when using the Nios II ISS might cause an exception dialog box to open.

This exception might occur if you select **Debug as ISS** in the Nios II IDE and try to cast variables via the Variables view.

**Workaround**

There is no workaround.

**Solution Status**

Not fixed

**Trace Debug Does Not Support Instruction-Related Exceptions**

The instruction-related exception handler is not supported by the Nios II trace tools.

**Workaround**

There is no workaround.

**Solution Status**

Not fixed

**Trace Debug Does Not Support the JMPI Instruction**

The JMPI instruction is not supported by the Nios II trace tools.

**Workaround**

There is no workaround.

**Solution Status**

Not fixed

**Cannot Locate Source Code in Driver Files Shared by Multiple Projects**

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

**Workaround**

Open the closed system library project and resume debugging.

**Solution Status**

Not fixed

**Console Window Is Not Updated After ISS Error**

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

**Workaround**

Close the console window after receiving an ISS error. A new console window opens when a new message is available.

**Solution Status**

Not fixed

**ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components**

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera Triple Speed Ethernet media access control (MAC) or SG-DMA components because the Nios II ISS does not support these components.

**Workaround**

Remove the Triple Speed Ethernet MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim or test it on hardware.



Before removing the Triple Speed Ethernet MAC and SG-DMA components, make a copy of the unmodified system to ensure that you can return to the original configuration.

**Solution Status**

Not fixed

## Memory Window Sets Control Register Values Incorrectly

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window shows this incorrect value.

### Workaround

Use the GNU debugger (GDB) console window in the IDE, instead of the memory window, to write to the registers. For example, type the following command:

```
set {int} <register address>=0x1234←
```



You must refresh the memory window for it to correctly display the target value.

### Solution Status

Not fixed

## Programs That Interact With a Terminal Console on Windows Do Not Work

Programs with this behavior work in v. 6.0 and earlier, but do not work in Nios II IDE v. 6.1 and later.

The Eclipse platform in v. 6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with `nios2-terminal`, which still just sends `\n`.

### Workaround

Change the software running on the Nios II processor to parse for `\r\n` as well as `\n`.

### Solution Status

Not fixed

## “Run as ModelSim” in the Nios II IDE Fails

The **Run as ModelSim** command might fail on launch configurations created in v. 7.0 or earlier of the IDE. This problem does not occur for new launch configurations.

### Workaround

Select a location for the ModelSim tool from the launch configuration dialog box. You can use the **Browse** button next to the **ModelSim path** group, or type in a path to the ModelSim directory (for example `c:/altera/71/modelsim_ae/win32aloem`).

### Solution Status

Not fixed

## The Restart Command on the Run Menu Does Not Work

The **Restart** command on the **Run** menu does not work.

**Workaround**

Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following **bash** alias to break the target, then stop it:

```
alias break="kill -2 `ps -a | grep nios2-elf-gdb | cut -f6 -d' '\`"
```

**Solution Status**

Not fixed

**Watchpoints Do Not Work on Certain Variables**

Watchpoints do not work on a variable whose size is not 32 bits.

**Workaround**

Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

**Solution Status**

Not fixed

**Nios II IDE Errata: Navigating Projects**

This section describes in detail the issues related to navigating projects in the Nios II IDE.

**Nios II IDE Freezes While Displaying the Splash Screen**

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures the dialog box that asks you to specify the new workspace. As a result, the IDE appears to freeze.

**Workaround**

Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

**Solution Status**

Not fixed

**Internal Error When Double-Clicking on a Large Objdump File**

On Windows when opening a large **objdump** file in the Nios II IDE, you might get the following error message:

```
Unable to create this part due to an internal error.  
Reason for the failure: Editor could not be initialized.
```

### Workaround

Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS** program icon, right-click it, then click **Properties**. The **Windows Properties** dialog box appears.
2. In the **Target** field, append "vmargs -Xmx1024m" to the end of the path to the Nios II IDE executable. For example:

```
C:\altera\72\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
```

### Solution Status

Not fixed

## C/C++ Scanner Does Not Support Certain C/C++ Constructs

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with C code constructs not supported by the C++ language. An example is functions that take a function pointer as an argument.

### Workaround

If the C/C++ Search fails, use the File Search facility.

### Solution Status

Not fixed

## Nios II SBT Errata

This section describes in detail the issues in the Nios II SBT.

### Error Running Nios II Project: 'Downloading ELF Process failed'

If the Nios II processor's `cpu.data_master` port is not connected to all program memories (memories to which the `.elf` file is downloaded) the software project fails to run on Nios II hardware.

Failure to connect `cpu.data_master` to all program memories is a design error that the Nios II SBT does not detect.

### Workaround

Connect `cpu.data_master` to all program memories.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## BSP Editor Does Not Show Command-Line Help

The following commands in the Nios II Command Shell should display command-line help:

- `nios2-bsp-editor --help`
- `nios2-bsp-editor --extended-help`

Instead, these commands launch the BSP Editor.

### Workaround

For help with the BSP Editor, refer to “Using the BSP Editor” in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer’s Handbook*.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## BSP Not Updated for Memory Size Changes in SOPC Builder

If you change the size of a memory in SOPC Builder, the memory region size in any previously created BSP is no longer correct. Regenerating the BSP does not update the BSP’s memory region size.

### Workaround

In the BSP Editor, on the **Linker Script** tab, you can correct the memory region size either of the following ways:

- Edit the memory region size manually.
- Click **Restore Defaults** to rerun the default Tcl script. The default Tcl script reads the updated memory region size from the SOPC information file (`.sopcinfo`) and updates the BSP.



For detailed information about keeping your BSP consistent with changes in the underlying SOPC Builder system, refer to “Revising Your BSP” in the *Nios II Software Build Tools* chapter of the *Nios II Software Developer’s Handbook*.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Build Errors on Interrupt API Calls

You might observe build errors if your application project installs an ISR that calls the `alt_irq_register()` function.

The Nios II EDS v9.1 includes support for an optional External Interrupt Controller (EIC) interface on the Nios II processor. To support the EIC interface, the HAL includes an enhanced interrupt API.

When you create a BSP, the SBT determines which interrupt API(s) the device drivers in your system support. If all registered drivers in the BSP identify themselves as supporting the enhanced API, the SBT implements the enhanced API. All Altera device drivers in v9.1 and later support the enhanced API.

When the enhanced API is implemented, the legacy interrupt API, including `alt_irq_register()`, is not available. Therefore, if the application code contains a call to a legacy API function, a linker error occurs.

This issue might affect application projects that call `alt_irq_register()` or other legacy interrupt API functions directly, rather than depending on drivers in the BSP for interrupt support.

If your Nios II application project's source code registers an ISR using `alt_irq_register()`, or includes an ISR, your application might fail to compile in the Nios II EDS v. 9.1, due to undefined references to `alt_irq_register()`, or to a change to the ISR function prototype.

### Workaround

Modify your application code to use the new enhanced interrupt API. To use the enhanced API, you must modify several function calls and all ISR function prototypes.



For information about supporting the enhanced interrupt API, refer to the *Exception Handling* chapter of the *Nios II Software Developer's Handbook*, or to *AN 595: Vectored Interrupt Controller Usage and Applications*.



You must upgrade to the enhanced API if you wish to use an EIC in your hardware design to accelerate interrupt response time. The enhanced API also works with the Nios II processor's internal interrupt controller.

### Solution Status

Not fixed

## BSP Editor Incorrectly Allows Mixed Drivers

If multiple drivers are found for a single peripheral type, and you have multiple instances of the peripheral type in your SOPC Builder system, the Nios II BSP editor allows you to manually select a different driver for each instance of the peripheral or processor. For example, if you have two `altera_avalon_timer` devices, you can select two different timer drivers.

This behavior is not fully supported. You must choose the same driver for both instances of the peripheral type.

You can make driver assignments in one of the following ways:

- In the Nios II BSP editor, within the **Drivers** tab. The **Driver name** associated with each peripheral can be edited, if multiple driver types are found that support the peripheral. You must make the same driver type selection for each instance of a peripheral type in the list of peripherals as defined in the **Module Name** column.
- With the Nios II SBT in command-line mode (`nios2-bsp`, `nios2-bsp-create-settings`, `nios2-bsp-update-settings`): the **set\_driver** command can be invoked (either from a Tcl script or with the `--cmd` option). The **set\_driver** command accepts a peripheral module instance name and driver name as arguments. If you are using the command line development flow and wish to use this command, you must invoke it once for each instance of a peripheral type in your system, similar to the BSP editor flow described in the previous paragraph.

### Workaround

There is no workaround.

### Solution Status

Fixed in v. 9.1 of the Nios II EDS

## **nios2-bsp --help Displays Incorrect List of BSP Types**

The Nios II SBT supports custom HAL-based operating systems, when they are added as Tcl-scripted software packages. However, the `nios2-bsp` tool help message indicates that there are only two BSP types, HAL and UCOSII. If other HAL-based operating system components are added they do not appear in the `nios2-bsp` help message.

### Workaround

To get a correct, dynamic list of available BSP types, type the following command:

```
nios2-bsp-create-settings --type x --settings x --sopc=<existing sopcinfo path>←
```

This command returns an error message that lists available BSP types:

```
SEVERE: Available BSP type values for the --type argument are: [<list of BSP types>]
```

### Solution Status

Fixed in v. 9.1 of the Nios II EDS

## **Java Heap Space Error When Creating BSP**

When creating a BSP, the Nios II SBT might hang with the following message:

```
SEVERE: OutOfMemoryError: Java Heap Space.
```

This problem occurs when your SOPC Builder design has connection cycles. A connection cycle consists of a group of components that master one another in a circular arrangement. For example, if `ddr_sdram_bridge` masters `lcd_sgdma`, and `lcd_sgdma` masters `ddr_sdram_bridge`, the two components form a simple connection cycle. In this situation, the SBT is unable to create a BSP.

**Workaround**

Avoid connection cycles in the SOPC Builder system.

**Solution Status**

Fixed in v. 9.0 of the Nios II EDS

**SBT Fails if Nios II EDS is Installed in a Path Containing Spaces**

The Nios II Command Line SBT fails if you install the Nios II EDS in a path containing spaces.

**Workaround**

Reinstall the Nios II EDS to a path that does not contain spaces.

**Solution Status**

Not fixed

**User-Managed BSP Settings Not Supported in Nios II IDE**

This issue is described on [page 47](#).

**Upper-Case File Extensions Not Supported**

The Nios II SBT for applications and libraries (`nios2-app-generate-makefile` and `nios2-lib-generate-makefile` commands) does not support source files with certain upper-case extensions. If a file with an upper-case extension is included, the make command stops with no descriptive warning.

Only Nios II assembly language files built by the C preprocessor can have upper-case file extension (`.S`). All C language files must have the extension `.c` or `.h`. C++ language source files must have the extension `.cpp`, `.cxx`, `.cc`, or `.h`.

**Workaround**

Rename all C language files with the extension `.c` or `.h`. Rename all C++ language files with the extension `.cpp`, `.cxx`, `.cc`, or `.h`.

**Solution Status**

This issue will be fixed in a future release of the Nios II EDS.

**Peripheral Errata**

This section describes in detail the Nios II EDS issues related to peripheral components.

**Vectored Interrupt Controller Does Not Support VHDL Simulation Models**

This issue is described on [page 41](#).

## Unaligned Transfers of Small Payloads Fail on SG-DMA

The Scatter Gather DMA SOPC Builder peripheral does not correctly handle unaligned transfers with small payloads. A payload length smaller than the data width causes erroneous data transfers.

### Workaround

Avoid using DMA devices to transfer small payloads.

If absolutely necessary, for a 32-bit SG-DMA, a minimum length of 4 bytes guarantees that data is transferred correctly.

### Solution Status

Not fixed

## DMA Controller Always Busy in Burst Mode

The DMA controller component (`altera_avalon_dma`), when enabled for burst transactions, does not perform transfers at widths other than its full data width. The DMA controller is always busy.

### Workaround

When bursting is enabled, the DMA controller must be programmed to perform transactions at its full data width.

### Solution Status

Not fixed

## Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core

The `altera_avalon_mailbox` peripheral might not be initialized properly when a soft (non-system-wide) reset occurs. In this condition, mailbox contents (read and write pointers) are not reinitialized and might show potentially stale data.

### Workaround

Ensure that a system-wide reset event occurs by asserting the `reset_n` input to the SOPC Builder system containing the mailbox. This resets all peripherals and Nios II processors in the system.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Target Software Errata

This section describes in detail the Nios II EDS issues related to target software packages.

### NicheStack `tcp_wake()` Function Does Not Awaken All `select()` Sleepers

This issue impacts software based on the NicheStack TCP/IP Stack - Nios II Edition, with multiple sleeping tasks that depend on the same `select()` event to wake up. The `tcp_sleep()` and `tcp_wake()` functions do not work properly for tasks that call `select()`.

Tasks can normally use `tcp_sleep()` and `tcp_wake()` to suspend and then awaken themselves upon specific events. The `select()` function passes an event pointer to `tcp_sleep()` that is identical for every task that might call `select()`. All other events that call `tcp_sleep()` pass it a unique event pointer that specifically identifies them as the task to be awakened at the next event.

The `tcp_wake()` function apparently assumes that the list of sleepers contains unique event pointers by which they can be identified. Therefore, when a `select()` event occurs, instead of awakening all tasks waiting for it, `tcp_wake()` awakens only the first task in the list. Therefore, when more than one task calls `select()` to await some outcome on a socket, only the highest-priority task in the sleepers list is awakened at `select()` events. Other sleeping tasks that depend on the same `select()` event do not wake up.

#### Workaround

There is no workaround.

#### Solution Status

This issue will be fixed in future release of the Nios II EDS.

### EIC-Based Nios II System Crashes With NicheStack

When you attempt to run software based on the NicheStack TCP/IP Stack - Nios II Edition on a hardware design incorporating an external interrupt controller such as the VIC, the Nios II system crashes during initialization, with unpredictable results.

#### Workaround

There is no workaround.

#### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

### Gigabit Ethernet Performance Issues

The NicheStack TCP/IP Stack - Nios II Edition might hang while sending transmission control protocol (TCP) packets to the host, or experience dropped packets when receiving TCP packets from the host. These issues can arise with version 3.1 of the NicheStack TCP/IP Stack, using the triple-speed Ethernet component to transmit over a Gigabit Ethernet point-to-point connection.

**Workaround**

Download and install the patches at [www.altera.com/support/kdb/solutions/rd01132009\\_588.html](http://www.altera.com/support/kdb/solutions/rd01132009_588.html).

Alternatively, upgrade to the Nios II EDS v. 9.0 or later.

**Solution Status**

Fixed in v. 9.0 of the Nios II EDS

**stdio Does Not Work with MicroC/OS-II and Small C Library**

`stdin`, `stdout`, and `stderr` do not work in MicroC/OS-II applications built with the Small C library option.

**Workaround**

Disable the small C library option.

**Solution Status**

Not fixed

**cout From MicroC/OS-II Task Does Not Send Data to stdout**

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` does not work from a task.

**Workaround**

Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

**Solution Status**

Not fixed

**malloc(), realloc() Failures With MicroC/OS-II**

When you use the MicroC/OS-II RTOS, calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()` or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block is originally allocated.

### Workaround

Use one of the following workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. Preallocated memory blocks make it possible to change thread priorities at runtime.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. Fixed memory arrays make it possible to change thread priorities at runtime.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You can change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

### Solution Status

Not fixed

## Toolchain Errata

This section describes in detail the Nios II EDS issues related to the Nios II and GNU compiler toolchains, including tools such as `gcc`, `gdb`, and `sof2flash`.

### C++ Exceptions Unsupported in a Multi-Threaded Environment

C++ exceptions are only supported in a single-threaded environment. They cannot be used in a multi-threaded environment such as MicroC/OS-II.



Also see [“C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain”](#).

### Workaround

In a multi-threaded environment, you need to guard C++ exceptions with semaphores.

### Solution Status

Not fixed

### C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain

With the GCC 4 toolchain, C++ exceptions are not supported.



Also see [“C++ Exceptions Unsupported in a Multi-Threaded Environment”](#).

### Workaround

If your single-threaded project requires C++ exceptions, use the GCC 3 toolchain.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## memcpy() Optimization Misalignment

When optimization is turned on (-O1 or higher), if you use `memcpy()` and the source pointer is aligned to a 32-bit boundary, the compiler implements `memcpy()` with word-oriented instructions as part of the optimization process. This optimization technique causes unexpected results in your software if `memcpy()` is used on a misaligned address.

### Workaround

Take steps to ensure that an optimized implementation of `memcpy()` is called only with aligned data pointers.

### Solution Status

This issue will be fixed in a future release of the Nios II EDS.

## Unable to Configure FPGA from Flash with Parallel Flash Loader

This issue is described on [page 39](#).

## GNU Assembler Does Not Accept the --defsym Flag

According to the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`.

### Workaround

There is no workaround.

### Solution Status

Not fixed

## How to Contact Altera

For the most up-to-date information about Altera products, refer to [Table 10](#).

**Table 10.** Contact Information

Contact (1)	Contact Method	Address
Technical support	Website	<a href="http://www.altera.com/support">www.altera.com/support</a>
Technical training	Website	<a href="http://www.altera.com/training">www.altera.com/training</a>
	Email	<a href="mailto:custrain@altera.com">custrain@altera.com</a>
Altera literature services	Email	<a href="mailto:literature@altera.com">literature@altera.com</a>
Non-technical support (General) (Software Licensing)	Email	<a href="mailto:nacomp@altera.com">nacomp@altera.com</a>
	Email	<a href="mailto:authorization@altera.com">authorization@altera.com</a>






**Note:**

(1) You can also contact your local Altera sales office or sales representative.

# Typographic Conventions

Table 11 shows the typographic conventions that this document uses.

**Table 11.** Typographic Conventions

Visual Cue	Meaning
<b>Bold Type with Initial Capital Letters</b>	Indicates command names, dialog box titles, dialog box options, and other graphical user interface (GUI) labels. For example, <b>Save As</b> dialog box. For GUI elements, capitalization matches the GUI.
<b>bold type</b>	Indicates directory names, project names, disk drive names, file names, file name extensions, dialog box options, software utility names, and other GUI labels. For example, <code>\qdesigns</code> directory, <code>d:</code> drive, and <code>chiptrip.gdf</code> file.
<i>Italic Type with Initial Capital Letters</i>	Indicates document titles. For example, <i>AN 519: Stratix IV Design Guidelines</i> .
<i>Italic type</i>	Indicates variables. For example, $n + 1$ . Variable names are enclosed in angle brackets (< >). For example, <file name> and <project name>.pof file.
Initial Capital Letters	Indicates keyboard keys and menu names. For example, Delete key and the Options menu.
“Subheading Title”	Quotation marks indicate references to sections within a document and titles of Quartus II Help topics. For example, “Typographic Conventions.”
Courier type	Indicates signal, port, register, bit, block, and primitive names. For example, <code>data1</code> , <code>tDi</code> , and <code>input</code> . Active-low signals are denoted by suffix <code>n</code> . For example, <code>resetn</code> .  Indicates command line commands and anything that must be typed exactly as it appears. For example, <code>c:\qdesigns\tutorial\chiptrip.gdf</code> .  Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword <code>SUBDESIGN</code> ), and logic function names (for example, <code>TRI</code> ).
1., 2., 3., and a., b., c., and so on.	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■	Bullets indicate a list of items when the sequence of the items is not important.
	The hand points to information that requires special attention.
	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
	A warning calls attention to a condition or possible situation that can cause you injury.
	The angled arrow instructs you to press Enter.
	The feet direct you to more information about a particular topic.



101 Innovation Drive  
San Jose, CA 95134  
[www.altera.com](http://www.altera.com)  
Technical Support  
[www.altera.com/support](http://www.altera.com/support)

Copyright © 2010 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



I.S. EN ISO 9001