Nios II Embedded Design Suite Release Notes and Errata

© March 2009 RN-EDS-4.0

About These Release Notes

These release notes cover versions 8.0 through 9.0 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*.

Prior to v7.2, Nios II release notes and errata were organized differently. You can refer to the documents from before v7.2 on the Altera website at:

- Nios II Embedded Design Suite Release Notes
- Nios II Embedded Design Suite Errata

Revision History

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

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*.

Revision Details

This section provides details about some of the Nios II EDS revisions shown in Table 1.

Table 1. Nios II Embedded Design Suite Revision History

Version	Date	Description
9.0.0	March 2009	Added the `nios2-convert-ide2sbt` tool, which converts projects from the Nios II integrated development environment (IDE) development flow to the Nios II software build tools development flow
8.1.1	January 2009	Nios Community Wiki URL changing (www.nioswiki.com)
8.1.0	November 2008	Maintenance release
8.0	May 2008	■ The Nios II BSP editor, a standalone graphical user interface (GUI), provides an easy-to-use front end for the Nios II software build tools. ■ The Altera Hardware Abstraction Layer (HAL) supports an advanced exception handler that can handle Nios II memory protection unit (MPU) exceptions. ■ The Nios II Studio is available as a Beta release. The Studio is an alternative to the Nios II IDE. Nios II projects created with the Nios II Studio are 100% compatible with projects created with the Nios II software build tools on the command line. ■ The Nios II software build tools `--sopc` option only accepts **.sopcinfo** files as arguments and not **.sopc** files. ■ The Nios II IDE command-line utilities, which form the infrastructure of the Nios II IDE, are deprecated; support will be discontinued in a future release. The deprecated tools include `nios2-create-system-library`, `nios2-create-application-project`, `nios2-build-project`, `nios2-import-project`, and `nios2-delete-project`. ■ The Lightweight IP (lwIP) transmission control protocol/Internet protocol (TCP/IP) suite is removed from the Nios II EDS. ■ The MicroC/OS-II real-time operating system (RTOS) is upgraded from v2.83 to v2.86. ■ The NicheStack TCP/IP stack — Nios II Edition is upgraded from v3.0 to v3.1. ■ Certain Nios II software examples are removed from the EDS, and are distributed on the Altera website instead. The affected designs and their new locations are listed in “Software Examples Moved to the Web” on page 2. ■ The NicheStack TCP/IP Stack macro definition `INICHE_DEFAULT_IF` is deprecated. For details, see “Deprecated NicheStack Setting”. ■ The **nios_csh** file in **sopc_builder/bin** is removed from the Nios II EDS.

Software Examples Moved to the Web

Starting in v8.0, the software examples shown in Table 2 are removed from the Nios II EDS, and instead distributed on the Web, at the locations shown in the table. This applies to software examples in Nios II IDE template form, and in the form of Nios II software build tools `create-this-app` and `create-this-bsp` scripts.

Deprecated NicheStack Setting

The Nios II IDE and the Nios II software build tools contain a setting to specify a default Ethernet (MAC) interface when using the NicheStack TCP/IP stack. Starting in v8.0, the default MAC setting is deprecated. It is no longer used in Altera software application example code. The NicheStack TCP/IP stack has never depended on this setting.

Table 2. Software Examples Removed from the Nios II EDS

Software Example	Delivery Mechanism
Read-Only Zip File System	The ExampleDesigns page of the Nios Community Wiki (www.nioswiki.com)
MicroC/OS-II Message Box	The ExampleDesigns page of the Nios Community Wiki (www.nioswiki.com)
Hello_LED	The ExampleDesigns page of the Nios Community Wiki (www.nioswiki.com)
Host File System	The ExampleDesigns page of the Nios Community Wiki (www.nioswiki.com)
Dhrystone	■ The Literature: Nios II Processor page, with the *Nios II Performance Benchmark Data Sheet* ■ The Fast Nios II Hardware Design Example page
MicroC/OS II Tutorial	The Literature: Nios II Processor page, with the *Using MicroC/OS-II RTOS with the Nios II Processor Tutorial*
Tightly Coupled Memory (TCM) Tutorial	The Literature: Nios II Processor page, with the *Using Nios II Tightly Coupled Memory Tutorial*
Custom Instruction Tutorial	The Literature: Nios II Processor page, with the *Nios II Custom Instruction User Guide*

The default MAC setting generates the macro `INICHE_DEFAULT_IF` in the generated **system.h** file located in the board support package (BSP). If your software depends on this macro definition, modify it, for example, to define `INICHE_DEFAULT_IF` explicitly. The macro will be removed in a future Nios II EDS release.

Altera networking example applications do not support multiple Ethernet interfaces. If you wish to use the Web Server or Simple Socket Server networking examples in a hardware system with multiple Ethernet MAC interfaces, or with the multi-port Altera Triple Speed Ethernet MegaCore function, you need to modify the `get_ip_addr()` and `get_mac_addr()` functions to provide a unique MAC and IP address for each Ethernet interface in the system.

Errata Summary

Table 3 summarizes the issues that affect the Nios II Embedded Design Suite versions 8.0 through 9.0. For a detailed description of each issue, click on the issue name.

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

Added or Updated	Issue	Page	Affected Version		
			9.0	8.1	8.0
	C2H Compiler				
Nov 2008	Error "java.lang.NoClassDefFoundError" During C2H Compilation	7	✓	✓	—
Jul 2008	C2H Cache Coherency Issue with Global Variables	8	✓	✓	✓
Jul 2008	Functions Declared Without a Return Type Are Not Supported	9	✓	✓	✓
Oct 2007	Pre-7.1 Systems Are Not Supported	9	✓	✓	✓
Oct 2007	--src-dir Software Build Tools Argument Does Not Work With C2H	9	✓	✓	✓
May 2007	Accelerator Generation Failure If Tools Are Installed in Path With Spaces	9	✓	✓	✓
May 2007	C2H Accelerators Time Out Immediately When Untethered	*(1)*	✓	✓	✓
May 2007	The C2H Compiler Regenerates an Accelerator Unnecessarily	10	✓	✓	✓
May 2007	Error: c2h_accelerator_base_addresses.h: No such file or directory	10	✓	✓	✓

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

Added or Updated	Issue	Page	Affected Version		
			9.0	8.1	8.0
May 2007	Java Heap Space Exception if Quartus II Compilation is Enabled	10	✓	✓	✓
May 2007	Pointer Dereferences to Volatile Types	11	✓	✓	✓
Dec 2006	C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	11	✓	✓	✓
Dec 2006	C2H Compiler Out-of-Memory Error	*(1)*	✓	✓	✓
Jun 2006	Array Elements in Structures Do Not Copy Correctly	11	✓	✓	✓
Jun 2006	Clean Build Causes Build Failure	12	✓	✓	✓
Jun 2006	Changing Build Configurations Produces Unexpected Results	12	✓	✓	✓
Jun 2006	Closed System Library While Working With the C2H Compiler	*(1)*	✓	✓	✓
Jun 2006	Hardware Accelerators Remain After Deleting the Software Project *(2)*	13	✓	✓	✓
Jun 2006	Incorrect Results From Logical or Conditional Operation With Side-Effects	13	✓	✓	✓
Jun 2006	Launch SOPC Builder Button in C2H View	14	✓	✓	✓
Jun 2006	Multiple Clock Domains Cause Hardware Accelerator to Fail	*(1)*	✓	✓	✓
Jun 2006	Simulating on ISS Not Supported	*(1)*	✓	✓	✓
Development Boards					
Jun 2006	Intermittent Failures While Accessing CompactFlash Card	14	✓	✓	✓
Documentation Issues					
Mar 2009	Compiler Flags for Building Custom Newlib	15	✓	✓	✓
Jan 2009	Obsolete Wiki URLs in Help Systems and Handbook	16	Fixed	✓	—
Nov 2008	Error Message on Invalid Exception or Reset Vector	16	✓	✓	—
Jul 2008	Missing Source Control Information	17	✓	✓	✓
Dec 2007	Incomplete newlib Documentation	*(1)*	✓	✓	✓
Oct 2007	Nios II IDE Online Help Expand Buttons Do Not Work	18	✓	✓	✓
Download Cables and Debug Hardware					
Jun 2006	Communication Errors During Run/Debug Sessions Using Older Download Cables	*(1)*	✓	✓	✓
Hardware Example Designs					
Mar 2007	Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs	19	✓	✓	✓
Software Examples					
Nov 2008	Simple Socket Server Example Displays Incorrect IP Address	20	✓	✓	—
Oct 2007	Hardware Tutorial Software Example Hangs on Some Boards	20	✓	✓	✓
Jun 2006	Networking Examples	21	✓	✓	✓
Jun 2006	RAM Test Failure When Running Memory Test Software Template on the ISS	*(1)*	✓	✓	✓
Flash Programmer					
May 2007	Delay When Creating New Flash Programmer Configuration	21	✓	✓	✓
Dec 2006	elf2flash File Size Limit	21	✓	✓	✓
Dec 2006	Error Message “No such file or directory” When Programming Flash	22	✓	✓	✓

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

Added or Updated	Issue	Page	Affected Version		
			9.0	8.1	8.0
Dec 2006	Flash Programmer Error with Multiprocessor System	*(1)*	✓	✓	✓
Hardware Simulation					
Dec 2006	Error “UNC paths are not supported” Launching ModelSim	22	✓	✓	✓
Jun 2006	ModelSim Fails to Load Large Memory Models	*(1)*	✓	✓	✓
Jun 2006	Uninitialized .bss Variables in Simulation	22	✓	✓	✓
Host Platform					
Dec 2007	Windows/Cygwin: Nios II Processor Generation Failure	23	✓	✓	✓
Oct 2007	Windows Vista: Limited Support in Nios II EDS	23	✓	✓	✓
Jun 2006	Linux: Process Fails To Terminate While Debugging With the Nios II ISS Target	24	✓	✓	✓
Jun 2006	Linux: The Quartus® II Stand-Alone Programmer is Not Supported	*(1)*	✓	✓	✓
Jun 2006	Windows: Frisk Antivirus Software Causes SOPC Builder and Nios II Command Shell to Be Unresponsive	*(1)*	✓	✓	✓
Nios II IDE					
Building Projects					
Mar 2009	Nios II IDE Command-Line Tools Select Wrong Workspace on Linux	24	✓	✓	✓
Mar 2009	Nios II IDE Command-Line Tools Hang on Windows	25	✓	✓	✓
Nov 2008	Nios II IDE Cannot Find stdio.h in Outline View	25	✓	✓	✓
Jul 2008	Nios II IDE Hangs With UNC Project Path	26	✓	✓	✓
Jul 2008	Build Failure with Nios II Advanced Exceptions, MMU, and MPU	26	✓	✓	✓
May 2007	Build Command Not Functional for BSPs Created Using the Software Build Flow	26	✓	✓	✓
May 2007	Linker Errors with Dual-Port Memories	27	✓	✓	✓
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	27	✓	✓	✓
May 2007	URL Project Location Causes Project Creation Error	27	✓	✓	✓
Dec 2006	Compilation Error with Separate Exception Stack Option	28	✓	✓	✓
Dec 2006	Illegal Project Location Error When Creating New Project	*(1)*	✓	✓	✓
Dec 2006	Incorrect Address Assignment for Dual-Port Memory Mastered by Two Different Master Ports	*(1)*	✓	✓	✓
Dec 2006	Incorrect Stack and Heap Free Space Report	28	✓	✓	✓
Dec 2006	Nios II IDE Reports Problems Without Displaying Error in Console	28	✓	✓	✓
Jun 2006	Build Errors After Changing Component Names in SOPC Builder	*(1)*	✓	✓	✓
Jun 2006	Nios II IDE Unnecessarily Updates the SOPC Builder System File (.ptf)	*(1)*	✓	✓	✓
Debugging Projects					
Nov 2008	IDE Cannot Display Imported Profiling Data	29	✓	✓	—
Jul 2008	Error Message when Downloading .elf File	29	✓	✓	✓
Jul 2008	Variable Casting Unsupported in ISS	29	✓	✓	✓
Jul 2008	Trace Debug Does Not Support Instruction-Related Exceptions	30	✓	✓	✓
Jul 2008	Error Accessing MMU and MPU Registers in the Debugger	30	✓	✓	✓
Oct 2007	Trace Debug Does Not Support the JMPI Instruction	30	✓	✓	✓

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

Added or Updated	**Issue**	**Page**	**Affected Version**		
			9.0	**8.1**	**8.0**
May 2007	Cannot Locate Source Code in Driver Files Shared by Multiple Projects	31	✓	✓	✓
May 2007	Console Window Is Not Updated After ISS Error	31	✓	✓	✓
May 2007	ISS Fails on Designs Containing TSE MAC or SG-DMA Components	31	✓	✓	✓
May 2007	Memory Window Sets Control Register Values Incorrectly	31	✓	✓	✓
May 2007	Previous 7.1 Beta Install Might Cause Internal Error When Launching FS2 in IDE	*(1)*	✓	✓	✓
Dec 2006	Programs That Interact With a Terminal Console on Windows Do Not Work	32	✓	✓	✓
Dec 2006	"Run as ModelSim" in the Nios II IDE Fails	32	✓	✓	✓
Dec 2006	The Restart Command on the Run Menu Does Not Work	32	✓	✓	✓
Jun 2006	Breakpoints on Adjacent Lines of Assembly Fail to Halt the Processor	*(1)*	✓	✓	✓
Jun 2006	Cannot Use Watchpoints in the Nios II IDE When the FS2 Console Is Open	*(1)*	✓	✓	✓
Jun 2006	Debugger Cannot Step Into __sflags(), and Continues Execution Instead	*(1)*	✓	✓	✓
Jun 2006	Incorrect Breakpoint Filtering on Threads	*(1)*	✓	✓	✓
Jun 2006	Missing Address and Data in the Trace View	*(1)*	✓	✓	✓
Jun 2006	nios2-gdb-server Fails to Terminate After Setting a Watch Point	*(1)*	✓	✓	✓
Jun 2006	"Step failed. Target is not responding (timed out)" Error Message	*(1)*	✓	✓	✓
Jun 2006	Uninitialized Memory Error When Executing From ISS	*(1)*	✓	✓	✓
Jun 2006	Watchpoints Do Not Work on Certain Variables	33	✓	✓	✓
Navigating Projects					
Dec 2006	Nios II IDE Freezes While Displaying the Splash Screen	33	✓	✓	✓
Dec 2006	Internal Error When Double-Clicking on a Large Objdump File	33	✓	✓	✓
Jun 2006	C/C++ Scanner Does Not Support Certain C/C++ Constructs	34	✓	✓	✓
Nios II Studio					
Mar 2009	Incorrect Default Workspace Name	34	✓	—	—
Nov 2008	Error in nios2-studio Help System	34	✓	✓	—
Jul 2008	No System ID Validation Performed	35	✓	✓	✓
Jul 2008	Spurious Messages When Importing Pre-8.0 Projects	35	✓	✓	✓
Jul 2008	public.mk Error or System Failure After Hardware Modification	35	✓	✓	✓
Nios II Software Build Tools					
Mar 2009	Cannot Include C++ File in Driver or Software Package	36	—	—	✓
Nov 2008	Incorrect Linker Errors	36	✓	✓	✓
Nov 2008	Java Heap Space Error When Creating BSP	37	✓	✓	—
Jul 2008	BSP Editor Incorrectly Allows Mixed Drivers	37	✓	✓	✓
Jul 2008	nios2-bsp --help Displays Incorrect List of BSP Types	38	✓	✓	✓
Jul 2008	nios2-bsp-create-settings Fails to List BSP Types	38	✓	✓	✓
Jul 2008	Error Creating BSP for Pre-8.0 System	38	✓	✓	✓
Jul 2008	Makefile Error with Absolute Source Path	39	✓	✓	✓
Oct 2007	Software Build Tools Fail if Nios II EDS is Installed in a Path Containing Spaces	39	✓	✓	✓
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	27	✓	✓	✓

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

Added or Updated	Issue	Page	Affected Version		
			9.0	8.1	8.0
May 2007	Upper-Case File Extensions Not Supported	39	✓	✓	✓
Peripherals					
Jul 2008	DMA Transfers Truncated or Stalled	*(1)*	✓	✓	✓
Jul 2008	Unaligned Transfers of Small Payloads Fail on SG-DMA	40	✓	✓	✓
Jul 2008	IDT71V416 SRAM Component HardCopy Incompatibility	40	✓	✓	✓
Jul 2008	Software Build Tools Fail With UART Driver	40	✓	✓	✓
May 2007	DMA Controller Always Busy in Burst Mode	41	✓	✓	✓
May 2007	Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core	41	✓	✓	✓
Jun 2006	Using the DEV_CLRn Signal Can Cause JTAG UART Instability After Device-Wide Reset	*(1)*	✓	✓	✓
Target Software					
Mar 2009	Gigabit Ethernet Performance Issues	42	—	✓	✓
Jul 2008	Degraded TCP/IP Performance with NicheStack TCP/IP stack and TSE MAC	*(1)*	✓	✓	✓
Jul 2008	Slow TSE MAC Performance with NicheStack TCP/IP Stack	*(1)*	✓	✓	✓
Jul 2008	Default IP Address Assigned Incorrectly with NicheStack TCP/IP Stack	*(1)*	✓	✓	✓
Dec 2006	Lightweight IP (lwIP) Failure When Using DHCP	*(1)*	✓	✓	✓
Dec 2006	stdio Does Not Work with MicroC/OS-II and Small C Library	42	✓	✓	✓
Jun 2006	cout From MicroC/OS-II Task Does Not Send Data to stdout	42	✓	✓	✓
Jun 2006	Creating New Custom HAL Components in the Nios II IDE	*(1)*	✓	✓	✓
Jun 2006	malloc(), realloc() Failures With MicroC/OS-II	42	✓	✓	✓
Jun 2006	Problems Using HAL Drivers With Toshiba Flash	*(1)*	✓	✓	✓
Toolchain (gcc, gdb, etc.)					
Jul 2008	sof2flash Names Output Files Incorrectly	43	✓	✓	✓
Dec 2006	GNU Assembler Does Not Accept the --defsym Flag	43	✓	✓	✓
Jun 2006	Breakpoints in C++ Constructors Fail to Halt the Processor	*(1)*	✓	✓	✓

Notes to Table 3:

(1) This issue is documented online. To see the solution, click on the name of the issue, or search the Altera Knowledge Database.

(2) For further information, refer to the *Nios II C2H Compiler User Guide*.

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

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

C2H Cache Coherency Issue with Global Variables

Cache coherency issues might be encountered when the following conditions are all true:

- A C2H-accelerated function writes to a global or static variable
- The Nios II processor is configured with a data cache
- The referenced global variable is located in cacheable memory

In this situation, the data in the global variable can be corrupted.

This issue can only arise if the accelerated function writes to the global or static variable. If the function merely reads from the variable, cache coherency is maintained correctly.

Workaround

Add a data cache flush operation in the accelerator's wrapper function. The wrapper can be found in **alt_c2h_<*function_name*>.c**. In this file, find the following comment:

```
/* Write 1 to address 0 starts the accelerator */
```

Directly after this comment, add the following function call:

```
alt_dcache_flush_all();
```

This function call ensures that the data cache is flushed immediately before the accelerator is started.

☞ Flushing the cache entails a performance penalty.

Alternatively, allocate your global and static variables in non-cacheable regions of memory. Placing these variables in non-cacheable regions eliminates any need to perform a data cache flush.

For information about allocating variables to non-cacheable regions, refer to the *Cache and Tightly-Coupled Memory* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

Not fixed

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 Software Build Tools Argument Does Not Work With C2H

When using the C2H Compiler with Nios II Software Build Tools, 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 information in the include file specific to the hardware accelerator into 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 1 cannot be scheduled concurrently because the `volatile` qualification overrides the `__restrict__` pragma.

Example 1. 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 IRQ-enabled accelerators that are launched concurrently from the processor, and use FIFOs 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 2, 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 2. 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 3.

Example 3. 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.

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 4.

When you create a BSP with a precompiled newlib, the Nios II software build tools select the newlib matching your BSP's compiler settings. When you create a custom newlib, you must ensure that the compiler flags listed in Table 4 match your BSP's settings.

☞ 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.

Table 4. 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

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
- The Nios II Studio help system

Workaround

You can visit the Nios Community Wiki at www.nioswiki.com.

Solution Status

Fixed in v. 9.0.

Error Message on Invalid Exception or Reset Vector

New 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 5 shows possible error messages and their causes.

Table 5. 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.

Table 5. Exception and Reset Vector Error Messages (Part 2 of 2)

Message	Cause
`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.
`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.

In the SOPC Builder system, ensure that the exception and reset vectors conform to all the criteria listed in Table 5.

Solution Status

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

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. This information does not appear in the referenced chapter.

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 executable and linking file (**.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: **app.c**, **header.h**, **assembly.s**, **lookuptable.dat**	No
Eclipse project file	**.project**	No
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
Software configuration file	**application.stf**	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	**.project**	Yes
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
System software configuration file	**system.stf**	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

Not fixed

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 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 dual data rate synchronous dynamic random access memory (DDR SDRAM) clocks are implemented with an incorrect phase-locked loop (PLL) phase shift.

This issue affects the following example designs:

- 2s60 RoHS TSE SG-DMA
- 2s60 RoHS Standard
- 2s60 RoHS Full Featured
- 2c35 TSE 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 `pll` 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.

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

Hardware Tutorial Software Example Hangs on Some Boards

The **count binary** application used in the tutorial might hang when run on some designs that do not contain a liquid crystal display (LCD) peripheral.

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

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.

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.

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 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.

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 SDK 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)
- RedHAT Linux v3.0 (32-bit)
- RedHAT Linux v3.0 (64-bit)
- RedHAT Linux v4.0 (32-bit)
- RedHAT 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.

Linux: Process Fails To Terminate While Debugging With the Nios II ISS Target

If you try to interrupt or terminate a debug session targeting the Nios II ISS, you might see an error message **Interrupt Failed or Terminate Failed**. This means that the `nios2-iss` process failed to terminate. The debug session appears to have terminated in the IDE, but the `nios2-iss` process still continues running.

Workaround

Open a command shell and kill the `nios2-iss` process as follows:

1. Type:

   ```
   jobs
   ```

   to get a list of process IDs.

2. Type:

   `kill -9` *<nios2-iss process ID>*

   to terminate the process.

Solution Status

Not fixed

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 4 shows how to specify a workspace path.

Example 4. 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 5 is the default workspace used by the Nios II IDE in version 9.0.

Example 5. 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 into 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 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 software build tools command-line development flow. Alternatively, use the Nios II Studio.

Solution Status

Not fixed

Build Command Not Functional for BSPs Created Using the Software Build Flow

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

Workaround

To build the BSP, build 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 using the new software build flow and imported into 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 software build tools projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II software build tools 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 Uniform Resource Locator (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 into a new workspace.

Solution Status

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

Compilation Error with Separate Exception Stack Option

Choosing the **Use a separate exception stack option** might cause the compilation error `"UNDEFINED VARIABLE %STACK_POINTER%"` when building a project. 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.

IDE Cannot Display Imported Profiling Data

If you create a software project in the Nios II software build tools command line flow, generate profiling data to **gmon.out** in the Nios II command shell, and then import the project into the IDE, you cannot use the Profiling perspective to view the profiling data. When you attempt to view **gmon.out**, the IDE displays the 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 an executable and linking formal file (**.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 new instruction-related exception handling introduced in version 8.0 is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Error Accessing MMU and MPU Registers in the Debugger

You might get the following error if you try to read or write an MMU or MPU register via the Nios II Debugger:

```
nios2-gdb-server-wrapped: nios2debug.cpp:538: virtual ADI_ERROR \
    NIOS2DEBUG::leave_debug_mode(): Assertion 'm_state == STATE_DEBUG' failed.
```

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.

To set an MPU region with consecutive value changes, only the last one, when leaving debug mode, is committed.

Solution Status

Not fixed

Trace Debug Does Not Support the JMPI Instruction

The new JMPI instruction introduced in version 7.2 is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

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

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 TSE MAC or SG-DMA Components

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera TSE MAC or SG-DMA components because the Nios II ISS does not support these components.

Workaround

Remove the TSE 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 TSE 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

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

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 Studio Errata

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

Incorrect Default Workspace Name

When first launched, the Nios II Studio prompts for a workspace folder name. The default workspace folder name is **nios2-studio-workspace-8.0**. If you accept the default workspace, and v8.0 of the Nios II EDS was previously installed on your system, the Studio might modify project information from the previous installation.

Workaround

To keep your Studio 9.0 projects separate from your Studio 8.0 projects, manually insert a different workspace name, such as **nios2-studio-workspace-9.0**.

Solution Status

Not fixed

Error in nios2-studio Help System

In the Windows operating system, with the Nios II Studio help system viewed in the internal browser, the following error message appears when you click any link:

A Runtime Error has occurred

Workaround

Use an external browser. To configure Nios II Studio to display the help system in an external browser, perform the following steps:

1. On the Window menu, click **Preferences**.

2. In the **Preferences** dialog box, click **Help**, and turn on **Use external browser**.
3. Expand **General** and click **Web Browser**.
4. Select **Use external Web browser**.

Solution Status

Not fixed

No System ID Validation Performed

The **Validate Nios II system ID before software download** option does not perform System ID validation on **.elf** files built outside Nios II Studio unless the System ID is specified.

Workaround

Specify the `--sidp` and `--id` arguments in the **Tcl** extra arguments field of the **Main** tab of your launch configuration.

Solution Status

Not fixed

Spurious Messages When Importing Pre-8.0 Projects

Nios II Studio gives the following message when you import a project from a pre-8.0 version.

```
The project nature will be imported unchanged. If you wish to change the project \
    to a command-line project, remove the .project and .cdtproject files before \
    finishing this import.
```

The message is incorrect. The project is actually upgraded to a Nios II Studio application or BSP. You do not need to remove the specified files.

The following errors might also appear and can be ignored.

```
Unable to invoke: nios2-app-update-makefile --app-dir <app_dir> \
    --add-src-rdir .settings obj
SEVERE: nios2-app-update-makefile failed.
SEVERE: UserErrorException: The following option has the incorrect number of arguments \
    "--add-src-rdir".
```

Workaround

No action needed.

Solution Status

Not fixed

public.mk Error or System Failure After Hardware Modification

If you modify your SOPC Builder system and do not regenerate your BSP in the Nios II Studio, you might receive the following error:

```
make: *** [public.mk] Error 1
```

or see system failures. System failures occur because there is a system ID mismatch that is not detected by the tools.

Workaround

Always regenerate your BSP project if you modify your SOPC Builder system.

Solution Status

Not fixed

Nios II Software Build Tools Errata

This section describes in detail the issues in the Nios II software build tools.

Cannot Include C++ File in Driver or Software Package

In the Nios II software build tools prior to v. 8.1, you cannot configure a device driver or software package to include a C++ source file. You can include a C source file with the `add-sw-property` Tcl command, using the `c_source` property. However, there is no equivalent property for C++ source files.

Workaround

Upgrade to the Nios II EDS v. 8.1 or later, and use the `cpp_source` property.

Solution Status

Fixed in v. 8.1

Incorrect Linker Errors

While building a HAL-based Nios II software project, the linker displays misleading error messages if the SOPC Builder system provides less than 32 bytes of memory at the reset vector. The linker errors are similar to the following:

```
/cygdrive/d/altera/80/nios2eds/bin/nios2-gnutools/H-i686-pc-cygwin/bin/../lib/gcc/ \
    nios2-elf/3.4.6/../../../../nios2-elf/bin/ld:/cygdrive/d/test3/software/ \
    hello_world_0_syslib/Debug/system_description/generated.x:186: warning: memory \
    region UNDEFINED not declared
/cygdrive/d/altera/80/nios2eds/bin/nios2-gnutools/H-i686-pc-cygwin/bin/../lib/gcc/ \
    nios2-elf/3.4.6/../../../../nios2-elf/bin/ld:/cygdrive/d/test3/software/ \
    hello_world_0_syslib/Debug/system_description/generated.x:186: parse error
collect2: ld returned 1 exit status
make: *** [hello_world_0.elf] Error 1
```

Workaround

In the SOPC Builder system, if the reset and exception vectors are in the same memory device, ensure that the exception vector is at least 32 bytes above the reset vector.

Solution Status

Fixed in v. 8.1

Java Heap Space Error When Creating BSP

When creating a BSP, the Nios II software build tools 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 software build tools are unable to create a BSP.

Workaround

Avoid connection cycles in the SOPC Builder system.

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 software build tools 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 above.

Workaround

There is no workaround.

Solution Status

Not fixed

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

In version 8.0, the Nios II software build tools support additional 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

See "nios2-bsp-create-settings Fails to List BSP Types" to get a correct, dynamic list of available BSP types.

Solution Status

Not fixed

nios2-bsp-create-settings Fails to List BSP Types

The help message for `nios2-bsp-create-settings` incorrectly states that using the `--type` option with no value creates a list of available BSP types.

Workaround

To get a list of available BSP types, execute the following command:

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

The following 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

Not fixed

Error Creating BSP for Pre-8.0 System

You receive the following error message if you try to create a BSP with the v8.0 software build tools using a hardware design from v7.2 or prior.

```
SEVERE: ErrorLogException: Attempt to load "NiosII_stratixII_2s60_standard.sopc" \
    failed. The --sopc option no longer accepts *.sopc files from SOPC Builder. \
    Re-generate your system in SOPC Builder (version 8.0 or later) to create a \
    *.sopcinfo file and use that instead.
SEVERE: nios2-bsp-update-settings failed.
nios2-bsp: nios2-bsp-update-settings failed
```

Workaround

You must regenerate your v7.2 or older design in SOPC Builder and recompile in the Quartus II software before creating a BSP using the software build tools v8.0.

Solution Status

Not fixed

Makefile Error with Absolute Source Path

You might receive the following error message from the Nios II Software build tools:

```
Makefile:: *** target pattern contains no '%'
```

This error message can appear if you are using Windows, you specify an absolute source path with the `--src-dir` option, and your application directory, specified with the `--app-dir` option, is the same as your source path.

Workaround

Use a relative path to the source directory, or organize your files so that your application directory is different from your source directory.

Solution Status

Not fixed

Software Build Tools Fail if Nios II EDS is Installed in a Path Containing Spaces

The Nios II Command Line Software Build Tools fail if you install the Nios II EDS in a path containing spaces.

Workaround

Reinstall the Nios II EDS into 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 27.

Upper-Case File Extensions Not Supported

The Nios II software build tools for applications and libraries (`nios2-app-generate-makefile` and `nios2-lib-generate-makefile` commands) do 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.

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 SGDMA, a minimum length of 4 bytes guarantees that data is transferred correctly.

Solution Status

Not fixed

IDT71V416 SRAM Component HardCopy Incompatibility

The IDT71V416 SRAM SOPC Builder component, when generated in a clock domain running at 50 MHz or less, contains logic that triggers the following Design Assistant warning, and thus is not compatible with HardCopy devices.

```
Warning: (Medium) Rule C106: Clock signal source should not drive registers that are \
    triggered by different clock edges. Found 1 node(s) related to this rule.
```

This warning appears because a negedge-clocked register is created for the SRAM write signal, to reduce latency at lower clock speeds.

Workaround

There is no workaround.

Solution Status

Not fixed

Software Build Tools Fail With UART Driver

If you use the software build tools on a design containing the SOPC Builder UART peripheral, and if driver ioctl() support for the `altera_avalon_uart` component is enabled, then the BSP build process might fail. The **system.h** generated by the software build tools contains an invalid type for the UART parameter.

Workaround

To correct **system.h**, carry out the following steps:

1. Manually edit the **system.h** file by changing the following line:

   ```
   #define UART1_PARITY "N"
   ```

   to:

   ```
   #define UART1_PARITY 'N'
   ```

   ☞ Change double quotes to single quotes.

2. Build the BSP.

Solution Status

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

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.

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.

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

Solution Status

Fixed in v. 9.0

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 GNU compiler toolchain (gcc, gdb, etc.).

sof2flash Names Output Files Incorrectly

If **sof2flash** is called with the `--save` option and a relative path is specified for the output file (that is, including "./" or "../"), the raw binary file that is saved is incorrectly named **.rbf** instead of **<flash filename>.rbf**.

Workaround

Provide a full path to the output file(s), or change to the output directory before running the **sof2flash** command so the output file specification is `--output=`<*filename*>`.flash`.

Solution Status

Not fixed

GNU Assembler Does Not Accept the --defsym Flag

According 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 the following table.

Contact *(1)*	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:

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

Typographic Conventions

The following table shows the typographic conventions that this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicates command names, dialog box titles, dialog box options, and other GUI labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	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, **\qdesigns** directory, **d:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicates document titles. For example, *AN 519: Stratix IV Design Guidelines.*
Italic type	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, `data1`, `tdi`, and `input`. Active-low signals are denoted by suffix `n`. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
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.

Visual Cue	Meaning
CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
WARNING	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
Technical Support
www.altera.com/support