

## About These Release Notes

These release notes cover versions 10.1 through 11.1 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 recent list of errata for the Nios® II EDS, refer to the [Knowledge Base](#) page of the Altera website. Also, you can use the Knowledge Base to search for errata based on the product version affected and other criteria.

## Product 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   |
|---------|---------------|---|
| 11.1    | November 2011 | Maintenance release   |
| 11.0    | May 2011      | <ul style="list-style-type: none"> <li>■ The Nios II EDS by default supports the GCC 4.1.2 version of the GNU Compiler Collection (GCC). GCC 3.4.6, the Nios II IDE, and Nios II C2H Compiler are optional features. They are available only if you enable <b>Legacy Package: Nios II IDE / GCC3 Toolchain / C2H Compiler</b> when you install the Altera Complete Design Suite.</li> <li>■ Projects created with version 10.1 and earlier of the Nios II SBT need to be re-imported to create the Nios II launch configuration correctly.</li> </ul> |
| 10.1    | December 2010 | Maintenance release   |



For more information about Nios II EDS features, refer to the [Nios II Processor Reference Handbook](#), the [Nios II Software Developer's Handbook](#), and the [Nios II C2H Compiler User Guide](#).

## 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.

## Errata

---

## Nios II IDE Help Content Missing on Linux Platforms

### Description

You might be unable to view Nios II IDE Help content when running on newer versions of the Linux operating system. The Nios II IDE requires an outdated version of SWT library `libswt-mozilla-gtk-3235.so`, which is not included in newer versions of Linux.

### Workaround

Altera recommends migrating your projects to the Nios II Software Build Tools for Eclipse.

---

## Missing Design Examples Mentioned in Nios II Software Developer's Handbook

### Description

The *Nios II Software Developer's Handbook* refers to design files that are no longer bundled with the current Nios II EDS installation. The [Overview of Nios II Embedded Development](#) and [Nios II Software Build Tools Reference](#) chapters refer to design files that are no longer part of the installation.

### Workaround

Obtain the design examples directly from the [Design Examples](#) page on the Altera website.

---

## **SOPC Builder Not Supported for DDR SDRAM Controller with ALTMEMPHY**

### **Description**

SOPC Builder does not support the DDR SDRAM Controller with ALTMEMPHY in version 10.1. The DDR2 SDRAM Controller with ALTMEMPHY is supported.

### **Workaround**

No workaround is available at this time. Consider migrating your system to Qsys.

## **nios2-download Does Not Recognize CPU name in Qsys Subsystems**

### **Description**

You cannot download .elf files to Nios II processors in Qsys subsystems if all the processors have the same name.

### **Workaround**

Rename the Nios II processors in the Qsys subsystems with unique names, such as **cpu\_one** and **cpu\_two**.

## Run as Nios II Hardware Fails

### Description

Sometimes you cannot run a design as Nios II Hardware.

### Workaround

Select **Run Configurations** from the Run menu, point to **Apply**, and click **Run**.

---

## Uninitialized .bss Variables in Simulation

### Description

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.

### Workaround

No workaround is available.

---

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

### **Description**

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.

---

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

### Description

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);
```

---

## Watchpoints Do Not Work on Certain Variables

### Description

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.

## Hardware Accelerators Remain After Deleting the Software Project

### Description

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.

---

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

### Description

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.

---

## Changing Build Configurations Produces Unexpected Results

### Description

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.

## Launch SOPC Builder Button in C2H View

### Description

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.

---

## Clean Build Causes Build Failure

### Description

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.

---

## Array Elements in Structures Do Not Copy Correctly

### Description

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

In “[Array Elements of Structs](#)”, 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 1. 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 “[Copying Array Elements Explicitly](#)”.

### Example 2. Copying Array Elements Explicitly

---

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

---

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

### **Description**

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 is available.

---

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

### Description

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.

---

## elf2flash File Size Limit

### Description

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.

---

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

### Description

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' '\"
```

---

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

### Description

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.

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

### **Description**

`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.

---

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

### Description

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

The Eclipse platform in version 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`.

---

## Compilation Error with Separate Exception Stack Option

### Description

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.

---

## Nios II IDE Freezes While Displaying the Splash Screen

### Description

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.

---

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

### Description

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
```

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

### Description

If you launch the ModelSim®-Altera simulation software 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.

---

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

### **Description**

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

### **Workaround**

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

---

## Pointer Dereferences to Volatile Types

### Description

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

---

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

### Description

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.

---

## Memory Window Sets Control Register Values Incorrectly

### Description

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.

---

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

### Description

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.

---

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

### Description

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.

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

### **Description**

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.

---

## Console Window Is Not Updated After ISS Error

### Description

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.

---

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

### Description

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.

---

## Linker Errors with Dual-Port Memories

### Description

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.

---

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

### Description

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.

---

## DMA Controller Always Busy in Burst Mode

### Description

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.

---

## URL Project Location Causes Project Creation Error

### Description

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.

---

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

### Description

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.

---

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

### Description

The **Run as ModelSim** command might fail on launch configurations created in version 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`).

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

### **Description**

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.

---

## The C2H Compiler Regenerates an Accelerator Unnecessarily

### Description

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.

---

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

### **Description**

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.

## Pre-7.1 Systems Are Not Supported

### Description

Starting in version 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder version 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 version 7.1 or later. SOPC Builder prompts you to convert the system, creating a \*\_**hw.tcl** file compatible with the C2H Compiler.

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

### Description

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.

---

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

### **Description**

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.

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

### **Description**

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

### **Workaround**

There is no workaround.

---

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

### Description

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 1](#) is the default workspace used by the Nios II IDE in version 9.0.

#### **Example 1.** Providing a Workspace Location

---

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

---

---

## Windows/Cygwin: Nios II Processor Generation Failure

### Description

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.

---

## Windows Vista: Limited Support in Nios II EDS

### Description

The Quartus II software introduced Windows Vista (32-bit and 64-bit) support in version 7.2. However, several ensuing releases of the Nios II Embedded Design Suite support 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 version 3.0 (32-bit)
- Red Hat Linux version 3.0 (64-bit)
- Red Hat Linux version 4.0 (32-bit)
- Red Hat Linux version 4.0 (64-bit)

### Workaround

Upgrade to a more recent version of the Nios II EDS.

To see the current operating system support for the Altera Complete Design Suite tools, refer to the [Operating System Support](#) page of the Altera website.

---

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

### Description

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 1](#) shows how to specify a workspace path.

#### **Example 1.** Providing a Non-Default Workspace Location in Linux

---

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

---

## Variable Casting Unsupported in ISS

### Description

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.

---

## Trace Debug Does Not Support Instruction-Related Exceptions

### Description

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

### Workaround

There is no workaround.

## Nios II IDE Hangs With UNC Project Path

### Description

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.

---

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

### Description

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.

---

## IDE Cannot Display Imported Profiling Data

### Description

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.

---

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

### Description

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.

---

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

### Description

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.

---

## Error Message when Downloading .elf File

### Description

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.

---

## Functions Declared Without a Return Type Are Not Supported

### Description

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`.

---

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

### Description

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`.

## Compiler Flags for Building Custom Newlib

### Description

“Common BSP Tasks” in 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 1](#).

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 1](#) match your BSP’s settings.

**Table 1.** GCC Compiler Options for Newlib

| Option Name                   | Purpose  |
|-------------------------------|--|
| <code>-pg</code>              | Link for profiling with gprof                              |
| <code>-EB</code>              | Generate big-endian code                                   |
| <code>-mstack-check</code>    | Enable stack checking                                      |
| <code>-mno-hw-mul</code>      | Disable use of the <code>mul</code> family of instructions |
| <code>-mhw-mulx</code>        | Enable use of the <code>mulx</code> family of instructions |
| <code>-mcustom-fpu-cfg</code> | 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**.

---

## **memcpy() Optimization Misalignment**

### **Description**

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.

---

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

### Description

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.

---

## Error Messages on Console When Debugging

### Description

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

Disregard the messages.

---

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

### Description

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.

---

## Build Errors on Interrupt API Calls

### Description

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

The Nios II EDS version 9.1 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 version 9.1 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 version 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.

---

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

### Description

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.

---

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

### Description

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 -vr
```

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

```
sudo yum update xulrunner
```

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

### **Description**

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.

## Errors Converting Nios II IDE Multiprocessor Project

### Description

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.

---

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

### Description

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
```

---

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

### Description

The Nios II EDS documentation incorrectly states the valid range for the `hal.log_flags` BSP setting.

### Workaround

Valid values of `hal.log_flags` range from -1 through 3.

---

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

### Description

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*.

---

## BSP Editor Does Not Show Command-Line Help

### Description

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*.

---

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

### Description

If your application or library makefile uses an absolute path and is generated with the GCC 3 toolchain, and you import it to the version 10.0 Nios II 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 version 10.0.

Alternatively, use the GCC 3 toolchain.

---

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

### Description

If an application or library project is created with version 9.1 SP2 or earlier (using the GCC 3 toolchain), and you import it to the version 10.0 Nios II SBT for Eclipse using the GCC 4 toolchain, you might see the build error shown in [Example 1](#).

### Example 1. Build Error in GCC 4

---

```
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 version 10.0 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:

- Edit the application project and library project makefiles and add the inserted code as shown in [Example 2](#).
- Right-click the BSP project, point to **Nios II** and click **Generate BSP**.
- Clean and rebuild the application and library projects.

---

**Example 2. Makefile Correction for GCC 4 Build Error**

---

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

---

Alternatively, install the Altera Complete Design Suite version 11.0 or later.

---

## Nios II EDS Requires Administrative Permissions on Windows Vista and Windows 7

### Description

On the Windows Vista and Windows 7 operating systems, 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.

Alternatively, start Nios II EDS applications in administrator mode. For example, when launching the Nios II SBT for Eclipse, right-click **Nios II <version> Software Build Tools for Eclipse** and click **Run As Administrator**.

## Missing Nios II Perspective

### Description

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.

---

## AN543 Contains Incorrect Information about Updating the Flash

### Description

“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:

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

---

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

### Description

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.

---

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

### **Description**

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

### **Workaround**

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

---

## The Restart Button in the Debugger Does Not Work

### Description

When debugging in the Nios II SBT for Eclipse, if you click the **Restart** icon, execution is suspended, with the following error message:

```
Don't know how to run. Try "help target."
```

### Workaround

Terminate the program, download it again, and start the debugger.

---

## **nios2-console Does Not Work in the GCC 4 Command Shell**

### **Description**

If you try to run the **nios2-console** command in the GCC 4 Command Shell, you see the following error message:

```
Can't locate strict.pm in @INC ...
```

### **Workaround**

Use the GCC 3 Command Shell.

---

## Quartus II Compilation Warnings for Nios II Stratix II 2S60 ROHS Example

### Description

You might see the following warnings if you try to compile the Nios II Stratix® II 2S60 ROHS example design, installed at *<Nios II EDS install path>/examples/vhdl/niosII\_stratixII\_2s60/standard* or downloaded from the Altera Wiki ([www.alterawiki.com](http://www.alterawiki.com)):

```
Warning (10541): VHDL Signal Declaration warning at
NiosII_stratixII_2s60_standard.vhd(59): used implicit default value
for signal
"cpu_data_master_read_data_valid_NiosII_stratixII_2s60_standard_clock_
0_in" because signal was never assigned a value or an explicit default
value. Use of implicit default value may introduce unintended design
optimizations.
```

```
Warning (10542): VHDL Variable Declaration warning at
altera_europa_support_lib.vhd(340): used initial value expression for
variable "arg_copy" because variable was never assigned a value
```

```
Warning (10542): VHDL Variable Declaration warning at
altera_europa_support_lib.vhd(344): used initial value expression for
variable "arg_length" because variable was never assigned a value
```

You can safely ignore these warnings.

The Nios II Stratix II 2S60 ROHS example is deprecated.

### Workaround

None.

---

## Warning Message: "'pragma\_reverse\_bitfields' attribute directive ignored"

### Description

The Nios II GCC 4 tool chain does not support the C compiler pragma and flag for reverse bit fields. If you try to use the compiler pragma `reverse_bitfields`, you see the following warning:

```
'pragma_reverse_bitfields' attribute directive ignored
```

Similarly, the C compiler flag `-mreverse-bitfields` is unsupported.

### Workaround

Use the GCC 3 tool chain.

---

## Nios II SBT for Eclipse Unable to Create New Application and BSP from Template

### Description

You should be able to create a new application and BSP from a template with the following steps:

- 1 Select **Nios II Application and BSP from Template**.
- 2 Choose your **.sopcinfo** file and select a template.
- 3 Click **Select an existing BSP project from your workspace**.
- 4 Click **Create**.
- 5 Choose the desired BSP options.
- 6 Click **Finish**.

However, if you try to create a new application and BSP with this method, the SBT creates a BSP, but no application, and the GUI becomes unresponsive.

### Workaround

Perform the following steps:

- 1 Click **Cancel** to close the previous GUI.
- 2 Select **Nios II Application and BSP from Template** again.
- 3 Choose your **.sopcinfo** file and select a template.
- 4 Click **Select an existing BSP project from your workspace**.
- 5 From the BSP list, select the BSP created previously.
- 6 Click **Finish**.

---

## Incorrect information about Embedded C++

### Description

The [Embedded Design Handbook](#) contains the following incorrect statement about C++ support:

The HAL supports only the standard Embedded C++ subset of the full C++ language. C++ programs that use features beyond this subset fail in the HAL environment. C++ features not available in Embedded C++ include polymorphism, templates, and single and multiple object inheritance. In general, features that consume a large amount of memory are not included in Embedded C++. Catch/throw exceptions fail in the MicroC/OS-II environment.

Nios II C++ support is not restricted to the obsolete Embedded C++ specification.

### Workaround

In place of the incorrect paragraph, refer to the following correct information:

Nios II C++ language support depends on the GCC tool chain. The Nios II GCC 4 C++ tool chain supports the following features:

- Polymorphism
- Friendship and inheritance
- Multiple inheritance
- Virtual base classes
- Run-time type information (`typeid`)
- The `mutable` type qualifier
- Namespaces
- Templates
- New-and-delete style dynamic memory allocation
- Operator overloading
- Standard Template Library (STL)

Exceptions and new-style dynamic casts are not supported.

---

## The Nios II Software Build Tools for Eclipse generate errors for Qsys (Beta)

### Description

If you generate the simple socket server and web server Nios II example designs that contain tristate components, the Nios II Software Build Tools for Eclipse generates errors similar to the following:

```
[Error ] `EXT_FLASH_BASE' undeclared
[Error ] `EXT_FLASH_BASE' undeclared
```

### Workaround

In the `network_utilities.c` file, on lines 213 and 419 change

```
EXT_FLASH_NAME
```

to

```
AV_TRI_S1_EXT_FLASH_0_NAME
```

and on line 290 change

```
EXT_FLASH_BASE
```

to

```
AV_TRI_S1_EXT_FLASH_0_BASE
```

---

## **C2H Compiler Does Not Work in the Nios II Command Shell with GCC 4**

### **Description**

The Nios II Command Shell with GCC 4 does not support the C2H Compiler.

### **Workaround**

Use the Nios II Command Shell with GCC 3.

---

## Nios II Launch Configurations Not Visible after Service Pack Installation

### Description

After installing Altera Complete Design Suite version 10.1 Service Pack 1 over an existing version 10.1 installation, the first time you start the Nios II SBT for Eclipse, you might not see Nios II-specific launch configurations, such as **Run As Nios II Hardware** and **Run As Nios II ModelSim**. Only the **Run As Local C/C++ Application** configuration might be visible.

### Workaround

Close and restart the Nios II SBT for Eclipse to see the Nios II-specific launch configurations.

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

### Description

If you have a Nios II system generated with SOPC Builder version 10.1 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 version 11.0 or later. If you attempt to do so, the Nios II compiler reports errors.

The SG-DMA component is updated for version 11.0 and the software driver is not compatible with older versions of hardware. The minimum compatible hardware version of a component is defined in the component's `_sw.tcl` file, for example `set_sw_property min_compatible_hw_version 11.0`.

### Workaround

Regenerate the design with SOPC Builder version 11.0 or later before attempting to compile the software in the Nios II SBT for Eclipse.

## **System Timestamp for System ID Peripheral Is Always 0**

### **Description**

When the System ID Peripheral is not added to the top level of Qsys system, the system timestamp for the System ID peripheral is always 0.

### **Workaround**

Remove the System ID Peripheral from any of the Qsys subsystems if you have them and add the System ID Peripheral only in the top level of Qsys system.

---

## Missing .dat File Error Message: “Failed to open VHDL file”

### Description

When you simulate a Qsys hardware design in VHDL that contains the JTAG UART core, and you run the simulation using the `ld_debug` command, you might see the following error message:

```
# ** Error: (vsim-7) Failed to open VHDL file
"system_tb_system_inst_jtag_input_stream.dat" in r mode.
```

You can safely ignore this error, because it does not affect the `stdout` output of the JTAG UART.

This error message does not appear when simulating a hardware design in the Verilog HDL.

### Workaround

Run the simulation using the `ld` command, and the error is not displayed.

Alternatively, ignore the error message.

---

## JTAG UART Interactive Window for Simulation Does Not Work

### Description

When you simulate a Qsys system, the JTAG UART interactive window might not work.

### Workaround

There is no workaround.

---

## Nios II IDE shows “Source not found” during debugging

### Description

When you are debugging a project in Nios II IDE and attempt to step into the `printf()` function, you might see an error message saying `Source not found`.

### Workaround

Port your project to the Nios II SBT for Eclipse for future software development and debugging purposes, as described in the [Using the Nios II Integrated Development Environment](#) appendix of the *Nios II Software Developer's Handbook*.

---

## Incorrect IRQ Information in system.h

### Description

For Qsys hardware designs, the Vectored Interrupt Controller (VIC) is not supported properly by the Nios II SBT. The SBT publishes incorrect information to the **system.h** file. The `ALT_ENHANCED_INTERRUPT_API_PRESENT` definition is missing, and IRQ priorities are defined as -1.

### Workaround

Use one of the following techniques to work around this issue:

- Create a set of `#define` statements in a new header (**.h**) file. This technique is recommended because your definitions are preserved if you regenerate the BSP. Ensure that the following information is defined properly:

```
#define ALT_ENHANCED_INTERRUPT_API_PRESENT
#define <component>_IRQ <interrupt priority from Qsys system>
```

- Manually code the correct `#define` statements in **system.h**. However, if you use this technique, you will lose your changes every time the BSP is regenerated and **system.h** is updated.
- Generate your system with SOPC Builder.
- Use the internal interrupt controller.

---

## Intermittent Failures While Accessing CompactFlash Card

### Description

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.

---

## 'Exclude from build' Not Supported

### Description

In the Nios II SBT for Eclipse version 9.1 SP2 and earlier, you can use the **Exclude from build** command to omit a source file from your project at build time. Starting with version 10.0, **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 version 10.0. 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**.

---

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

### **Description**

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.

---

## Networking Examples

### Description

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.

## The Qsys (Beta) Board Support Package (BSP) does not correctly specify the required driver when migrating a Common Flash Interface (CFI) memory component (altera\_avalon\_cfi\_flash) that requires a Nios II software driver

### Description

If you use Qsys upgrade transforms to migrate a Common Flash Interface (CFI) memory component (altera\_avalon\_cfi\_flash), which requires a Nios II software driver with initialization, the Board Support Package (BSP) does not correctly specify the required driver. Although the BSP compiles and no error message is issued, the component does not function correctly.

### Workaround

Instantiate the legacy Flash Memory Interface (CFI) component (altera\_avalon\_cfi\_flash). To transfer the required driver assignments to an upgraded Generic Tristate Controller:

- 1 On the **System** menu, select **Run Qsys Upgrade Transforms**.
- 2 In the Generic Tristate Controller GUI, enter `embeddedsw.configuration.softwareDriver` in the **Key** column of the **Module Assignments** table and enter `altera_avalon_flash_driver` in the **Value** column.
- 3 Modify the `alt_sys_init.c` file created during BSP generation by following these steps:
  - a. In the Device Headers section, change the include statement from `altera_generic_tristate_controller.h` to `altera_avalon_cfi_flash.h`.
  - b. In the Device Storage section, change  

```
ALTERA_GENERIC_TRISTATE_CONTROLLER_INSTANCE (<instance_name_caps>,  
<instance_name>)to  
ALTERA_AVALON_CFI_FLASH_INSTANCE(<instance_name_caps>,  
<instance_name>).
```

In the `alt_sys_init` function, change `ALTERA_GENERIC_TRISTATE_CONTROLLER_INIT (<instance_name_caps>, <instance_name>)` to `ALTERA_AVALON_CFI_FLASH_INIT (<instance_name_caps>, <instance_name>)`.

---

## The Qsys (Beta) Board Support Package (BSP) might fail to compile if your system requires a Nios II software driver with initialization

### Description

If your system includes a custom component that requires a Nios II software driver with initialization, the driver might not be found and the Board Support Package might fail to compile with errors about missing identifiers and assignments. Components are affected if the `_sw.tcl` file sets the property `set_sw_property auto_initialize true` to request `alt_sys_init.c` initialization.

### Workaround

Instantiate the custom component in SOPC Builder and then open the system in Qsys. To transfer the required driver assignments to an upgraded Generic Tristate Controller, on the **System** menu, select **Run Qsys Upgrade Transforms**. Then, in the Generic Tristate Controller GUI, enter `embeddedsd.configuration.softwareDriver` in the **Key** column of the **Module Assignments** table and, in the Value column, enter the driver name that is listed in the component's `_sw.tcl` file. You might need to add an entry into the `alt_irq_init` function, depending on the exact requirements of the component's driver. You might also need to add additional driver-specific header files to the Device Headers section.

---

## The Qsys (Beta) Board Support Package (BSP) fails to compile if you instantiate a Generic Tristate Controller for CFI

### Description

If you instantiate a Generic Tristate Controller for CFI in Qsys (which requires a Nios II software driver with initialization) the Board Support Package fails to compile with errors regarding missing identifiers and assignments.

### Workaround

Modify the `alt_sys_int.c` file created during BSP generation by following these steps:

- 1 In the Device Headers section, change the include statement from `altera_generic_tristate_controller.h` to `altera_avalon_cfi_flash.h`.
- 2 In the Device Storage section, change `ALTERA_GENERIC_TRISTATE_CONTROLLER_INSTANCE (<instance_name_caps>, <instance_name>)` to `ALTERA_AVALON_CFI_FLASH_INSTANCE (<instance_name_caps>, <instance_name>)`.

In the `alt_sys_init` function, change

```
ALTERA_GENERIC_TRISTATE_CONTROLLER_INIT (<instance_name_caps>,
<instance_name>) to ALTERA_AVALON_CFI_FLASH_INIT (<instance_name_caps>,
<instance_name>)
```

---

## Cannot simulate Nios II software with Qsys (Beta) systems

### Description

The Nios II Software Build Tools (SBT) for Eclipse do not support the Run as ModelSim simulation flow to simulate Nios II software code with Qsys systems.

### Workaround

Run your simulation outside the Eclipse environment. You can use the generated ModelSim script `mti_setup.tcl` as an example for your testbench and simulation environment.

To simulate a Nios II processor running software code, you must generate a Nios II Memory Initialization File (`.mif`). To generate the `.mif` file:

- 1 In Eclipse, right-click the application project, point to **Make Targets**, and then click **Build**.
- 2 Select `mem_init_install` and then click **Build**.

For the latest simulation recommendations, refer to New Qsys Issues.



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 © 2011 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