



# **SW Development Tools for Intel XScale® Microarchitecture – Build Tools FAQ**

Target OS: Windows\* CE and Windows Mobile\*

Version 1.0.0

Copyright © 2005 Intel Corporation

## Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL(R) PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications. Intel may make changes to specifications and product descriptions at any time, without notice.

The software described in this document may contain software defects which may cause the product to deviate from published specifications. Current characterized software defects are available on request.

This document as well as the software described in it is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Intel Corporation.

Developers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Improper use of reserved or undefined features or instructions may cause unpredictable behavior or failure in developer's software code when running on an Intel processor. Intel reserves these features or instructions for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from their unauthorized use.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino logo, Chips, Core Inside, Dialogic, EtherExpress, ETOX, FlashFile, i386, i486, i960, iCOMP, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel XScale, IPLink, Itanium, Itanium Inside, MCS, MMX, MMX logo, Optimizer logo, OverDrive, Paragon, PDCharm, Pentium, Pentium II Xeon, Pentium III Xeon, Performance at Your Command, Pentium Inside, skool, Sound Mark, The Computer Inside., The Journey Inside, VTune, Xeon, Xeon Inside and Xircom are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

\* Other names and brands may be claimed as the property of others.

Copyright © 2005 - 2006, Intel Corporation.

# Table of Contents

<b>Disclaimer .....</b>	<b>1</b>
<b>Topic: Compiler.....</b>	<b>3</b>
• I found a possible compiler bug. How should the bug be reported?.....	3
• What should I do if the Intel® C++ Compiler for Windows* CE build reports that x0__ar10.lib cannot be found? .....	3
• I get the warning /O3 replaced by /Ox. What does this mean? .....	4
• Enabling the Vectorizer .....	4
• Use of inline assembly with the Intel® C++ Compiler for Windows* CE triggers error message "code cannot be used in this architecture". .....	5
• What does ' <i>BUILD: [01:0000012954:ERROR] cl : Command line error D2004 : '/Tp' requires an argument'</i> ' mean? .....	5
<b>Topic: Assembler.....</b>	<b>6</b>
• What are the necessary changes required in a project to have Assembly files built with the Intel® Assembler? .....	6
Embedded Visual C++ 4.x.....	6
Visual Studio 2005.....	8
• Intel Assembler doesn't accept option -cpu .....	9
<b>Topic: File Formats.....</b>	<b>9</b>
<b>Problem Report Submission.....</b>	<b>10</b>

## Topic: Compiler

- **I found a possible compiler bug. How should the bug be reported?**

Any build tool bug requires three pieces of information: 1) a test case that can be built and run, 2) the exact command sequence to build the test case, including the command line arguments, and 3) a description of the failing behavior and the expected behavior. Once you have all three pieces of information, submit a problem report as follows:

Go to <https://premier.intel.com/>.

Type in your Login and Password. Both are case-sensitive.

Click the **Submit** button.

Read the Confidentiality Statement and click the **I Accept** button.

Click on the **Go** button next to the **Product** drop-down list.

Click on the **Submit Issue** link in the left navigation bar.

Choose **Development Environment (tools, SDV, EAP)** from the Product Type drop-down list.

If this is a software or license-related issue, choose **Intel® C++ Compiler, Windows\* CE, Std** or **Intel® C++ Compiler, Windows\* CE, Pro** from the **Product Name** drop-down list.

Enter your question and complete the fields in the windows that follow to successfully submit the issue.

**See Also:** N/A

**Product/Version:** N/A

- **What should I do if the Intel® C++ Compiler for Windows\* CE build reports that x0\_\_ar10.lib cannot be found?**

The Intel(R) C++ runtime library and floating point library is installed into and normally at build time automatically included from the binary execution directory of the compiler.

In the case of eVC 4.0 this would for instance be:

C:\Program Files\Microsoft eMbedded C++ 4.0\EVC\wce500\bin

In the case of VS2005 this would for instance be:

C:\Program Files\Microsoft Visual Studio 8\VC\ce\bin\x86\_arm

When you migrate an older project it could be that this relationship was implicitly stored in the project. Then you may have to explicitly add this directory to your library include directories. In eVC 4.0 this would be the Tools pull down menu and there under "Options" you would select the Directories tab and select "Show directories for library files" to add the directory there.

**See Also:** N/A

**Product/Version:** N/A

- **I get the warning /O3 replaced by /Ox. What does this mean?**

If the 'Default' optimization is set in the Project Settings, the Microsoft optimization switch overrides the Intel optimizations settings entered in the Intel C++ Compiler Options dialog.

Use 'Custom' optimization in the Project Settings instead. The Intel optimization settings (e.g. /O3) are passed on then to the build process.

**See Also:** N/A

**Product/Version:** All

- **Enabling the Vectorizer**

The Vectorizer is implicitly set when using the following optimization switches: /O1, /O2, /O3, /Qip, /Qipo. When using any of those, the Vectorizer option -Qvec is automatically switched on. But the Vectorizer is NOT automatically switched on when only using the XScale specific option -QTPxscn, where n is 1, 2, 3, 4 (see Compiler User's Manual).

You have to enable the Vectorizer explicitly by adding -Qvec when NOT also using one of those options: /O1, /O2, /O3, /Qip, /Qipo.

**See Also:** N/A

**Product/Version:** All

- **Use of inline assembly with the Intel® C++ Compiler for Windows\* CE triggers error message "code cannot be used in this architecture".**

Although the Intel® C++ Compiler for Windows\* CE supports the keyword `__asm()` and has basic support for ARM style inline assembly for Intel XScale® microarchitecture targets, the Microsoft compiler does not.

If you use precompiled headers the \*.pch file generation triggers the Microsoft compiler, which will produce an error message. The code is translated by Intel compiler (\*.phi1) and by the Microsoft compiler (\*.phi) the Microsoft compiler fails.

There are two workarounds for this problem. If you use Intel specific code like ARM inline assembly, which is not supported by the Microsoft\* compiler, you may disable precompiled headers.

Alternatively you may wrap your Intel® C++ Compiler specific code in `ifdef` constructs.

```
#ifdef __INTEL_COMPILER
#endif
```

**See Also:** N/A

- **What does '*BUILD: [01:0000012954:ERRORE] cl : Command line error D2004 : '/Tp' requires an argument'*' mean?**

The problem is related to an issue in the `makefile.def` of some versions of the Platform Builder. It is just a simple typo:

It should be:

```
!IF ("$(PRECOMPILED_CXX)" == "") && ("$(USECXX_FLAG)" == "")
PRECOMPILED_GENERATE_CMD_LINE=/TC
!ELSE
PRECOMPILED_GENERATE_CMD_LINE=/TP
!ENDIF
```

Where as it is (in some releases of the Platform Builder):

```
!IF ("$(PRECOMPILED_CXX)" == "") && ("$(USECXX_FLAG)" == "")
```

```

PRECOMPILED_GENERATE_CMD_LINE=/Tc
!ELSE
PRECOMPILED_GENERATE_CMD_LINE=/Tp
!ENDIF

```

So correcting /Tp to /TP and /Tc to /TC manually should solve the problem.

**See Also:** N/A

## Topic: Assembler

- **What are the necessary changes required in a project to have Assembly files built with the Intel® Assembler?**

The procedure for setting up a Project to use the Intel Assembler is different in embedded Visual C++ and Visual Studio 2005. Both cases are explained below separately.

### Embedded Visual C++ 4.x

The project file (\*.vcp) has to be manually edited to enable Intel Assembler for Assembly files. A Custom build section has to be added to the Project file for each Assembly File. Here is an example for a file named init.s

```

# Begin Source File
SOURCE=init.s

!IF "$(CFG)" == "ppc - Win32 (WCE ARMV4) Release"

# Begin Custom Build
OutDir=.\ARMV4Rel
InputPath= init.s
InputName= init

"$(OutDir)\$(InputName).obj" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
    asxscee -mcpu 3 $(InputPath) -o $(OutDir)\$(InputName).obj

```

```
# End Custom Build
```

```
!ELSEIF "$(CFG)" == "ppc - Win32 (WCE ARMV4) Debug"
```

```
# Begin Custom Build
```

```
OutDir=.\ARMV4Dbg
```

```
InputPath= init.s
```

```
InputName= init
```

```
"$(OutDir)\$(InputName).obj" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"  
    asxsce -debug -mcpu 3 $(InputPath) -o $(OutDir)\$(InputName).obj
```

```
# End Custom Build
```

```
!ENDIF
```

```
# End Source File
```



## Visual Studio 2005

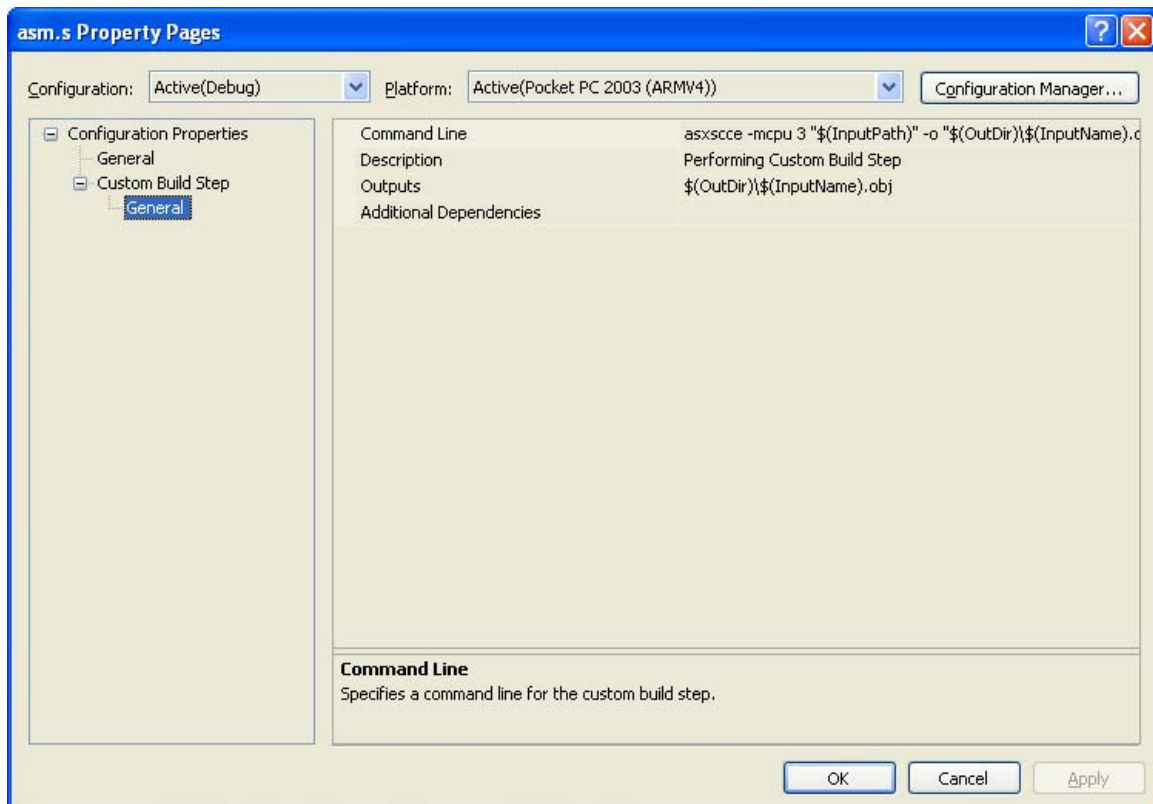
For each ASM File, add a custom build step. The steps are,

1. Open the Project/Solution in Visual Studio 2005
2. Ensure the ASM file is added to the solution. The File name should be visible in the solutions explorer.
3. Right-click on the file name and select Properties. This opens up the Configuration properties screen.
4. Go into custom Build step\General and edit the section as follows.

Command Line: `asxscc -mcpu 3 "$(InputPath)" -o "$(OutDir)\$(InputName).obj"`

Description: <>

Outputs: `$(OutDir)\$(InputName).obj`



**See Also:** N/A

**Product/Version:** All

- **Intel Assembler doesn't accept option -cpu**

The Intel Assembler doesn't support the Microsoft ARM Assembler option `-cpu`. The build process stops if the Intel Compiler (including Assembler) is chosen and the option `-cpu` is entered anywhere in the build options (normally in 'sources' files).

You have to search for occurrence of the option `-cpu` in the sources files of the project and comment it out. The corresponding Intel Assembler option for support of XScale CPUs is `-mcpu n`, where `n` is 1, 2, 3, 4 (please refer to the Intel Assembler Reference Manual).

**See Also:** N/A

**Product/Version:** All

**Topic: File Formats**

## Problem Report Submission

Your feedback is very important to us. To receive technical support for the tools provided in this product and technical information visit Intel® Premier Support at <https://premier.intel.com>.

For general support information please visit <http://www.intel.com/software/products/support/>.