Intel® Embedded Software Development Tool Suite 2.3 for Intel® Atom™ Processor Installation Guide and Release Notes

Installation Guide and Release Notes

Document number: 322069-008US

8 November 2011

Contents:

1 Introduction

This document provides a brief overview as well as usage information for the Intel® Embedded Software Development Tool Suite 2.3 for Intel® Atom™ processor and provides pointers to where you can find additional product information, technical support, articles and whitepapers.

It also explains how to install the Intel® Embedded Software Development Tool Suite 2.3 for Intel® Atom™ processor product. Installation is a multi-step process. Please read this document in its entirety before beginning and follow the steps in sequence.

The Intel® Embedded Software Development Tool Suite for Intel® Atom™ Processor consists of multiple components for developing, debugging, tuning and deploying system and application code targeted towards embedded designs using the Intel® Atom™ Processor.

Product Contents

The product contains the following components

1. Intel® C++ Compiler 12.1 for Linux*
2. Intel® Integrated Performance Primitives 7.0.5 for Linux*
3. Intel® VTune™ Amplifier XE 2011
   3.1. Intel® VTune™ Amplifier XE 2011 Update 5
   3.2. Sampling Collector 3.4 for Intel® VTune™ Amplifier XE 2011 (SEP)
4. Intel® Debuggers
   4.1. Intel® Application Debugger 2.3 for Intel® Atom™ Processor, Build [76.xxx.x]
   4.2. Intel® Debugger Remote Server for IA32 Linux*, Build [76.xxx.x]
   4.3. Intel® JTAG Debugger 2.3 for Intel® Atom™ Processor, Build [76.xxx.x]
   4.4. XFLASH – Intel® JTAG Debugger Flash Memory Tool 3.3.1

2   Technical Support and Documentation

Release Note Locations

The release notes for the tools components making up the Intel® Embedded Software Development Tool Suite 2.3 for Intel® Atom™ processor product can be found at the following locations after unpacking l_MID_DBG_p_2.3.tgz and running the install.sh installation script.

The paths are given relative to the installation directory <install-dir>. The default installation directory is /opt/intel unless indicated differently.

Intel® Embedded Software Development Tool Suite 2.3

- `<install-dir>/atom/Release_Install_All.pdf`

Intel® C++ Compiler for Linux*

- `<install-dir>/composerxe/Documentation/en_US/Release_NotesC.pdf`

Intel® Integrated Performance Primitives

- `<install-dir>/composerxe/Documentation/en_US/ipp/ReleaseNotes.htm`

Intel® VTune™ Amplifier XE 2011

- `<install-dir>/vtune_amplifier_xe_2011/documentation/en/release_notes_amplifier_xe_linux.pdf`

Sampling Collector for Intel® VTune™ Amplifier XE 2011

The installation directory <install-dir> for the Sampling Collector for Intel® VTune™ Amplifier XE 2011 is the directory into which the installation tar archive file sep34_axeu5_lin_ia32.tar.gz has been extracted

- `<install-dir>/docs/index.htm`

Intel® Application Debugger

- `<install-dir>/atom/idb/2.3.xxx/doc/Release_Notes_IDB.pdf`
- `<install-dir>/atom/idb/2.3.xxx/doc/Getting_Started.html`

Intel® JTAG Debugger

- `<install-dir>/atom/xdb/2.3.xxx/doc/Release_Notes_JTAG.pdf`
- `<install-dir>/atom/xdb/2.3.xxx/doc/Getting_Started.htm`

Article & Whitepaper Locations

Below is a list of articles and support locations relevant to the use of the Intel(R) Software Development Tool Suites for Intel® Atom™ processor. Please feel free to use these web locations for additional background information. Please also feel free to use the Support Forum at http://software.intel.com/en-us/forums/software-development-toolsuite-atom/ to submit issues and discuss them with other users and Intel employees.

Intel® Software Development Tool Suites for Intel® Atom™ processor - Support Forum:

http://software.intel.com/en-us/forums/software-development-toolsuite-atom/

Intel® Software Development Tool Suites for Intel® Atom™ processor - Documentation:

http://software.intel.com/en-us/articles/intel-embedded-tool-suite-documentation/

Intel® Software Development Tool Suites for Intel® Atom™ processor - Knowledge Base:

http://software.intel.com/en-us/articles/software-development-toolsuite-atom-kb/all/1/

Application Debugging in Cross-Debug Environment

http://software.intel.com/en-us/articles/cross-application-debugging/

Optimized for the Intel® Atom™ processor with Intel's Compiler

http://software.intel.com/en-us/articles/atom-optimized-compiler/

Support

To submit issues related to this product please visit the Intel Premier Support webpage and submit issues under the product **Intel(R) Embedded SW Dev Tools Atom**.

For information on how to register for and purchase support for the Intel(R) Embedded Software Development Tool Suite Intel(R) Atom(TM) processor please visit the Intel(R) Software Development Products webpage.

Additional information on the Intel® Embedded Software Development Tool Suite for Intel® Atom™ Processor is available at the following web resources:

- Product Page: http://www.intel.com/software/products/atomtools
- User Forum: http://software.intel.com/en-us/forums/software-development-toolsuite-atom/
- Knowledge Base Articles: http://software.intel.com/en-us/articles/software-development-toolsuite-atom-kb/all/1/

Please remember to register your product at https://registrationcenter.intel.com/ by providing your email address. This helps Intel recognize you as a valued customer in the support forum.

For information about how to find Technical Support, product documentation and samples, please visit http://www.intel.com/software/products/atomtools

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2®, SE3, and SSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimizations on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

3   What's New

The Intel® Embedded Software Development Tool Suite 2.3 contains the following updated features:

Intel® C++ Compiler

Please refer to <install-dir>/composerxe/Documentation/en_US/Release_NotesC.pdf for new features.

Intel® VTune™ Amplifier XE 2011

Please refer to <install-dir>/vtune_amplifier_xe_2011/Documentation/en for additional features.

- The Intel® VTune™ Amplifier XE 2011 provides an integrated performance analysis and tuning environment with graphical user interface that helps you analyze code performance on systems with Intel® Atom™ Processor. The Intel® VTune™ Amplifier XE 2011 has a new standalone graphical user interface (GUI) as well as a command-line interface (CLI).

- It also comes with predefined sampling data views and examples making even complex performance analysis more intuitive and accessible

Intel® Sampling Collector for Intel® VTune™ Amplifier XE 2011

Please refer to <install-dir>/vtune_amplifier_xe_2011/Documentation/en for additional features.

- Event-based sampling support for latest generation Intel® Atom™ Processors

- Support for Linux* kernel versions 2.6.9 through 3.1.x.

- Support for Yocto Project* 1.1, MeeGo* 1.x, as well as CE Linux* PR20-PR23 releases for Intel® Atom™ Processor CE4xxx.

Intel® Application Debugger for Intel® Atom™ Processor

Please refer to <install-dir>/atom/idb/2.3.xxx/doc/ and Release_Notes_IDB.pdf for additional features.

- Updated Eclipse* RCP based graphical user interface
- Fixes of outstanding issues
- Thread Grouping and Thread Specific Run-Control
- Enhanced C++ Class Awareness

- Updated usage model and fully documented support for QEMU based virtual device debug as well as cross-debug for real physical devices based on Intel® Atom™ Processor
- Support for Yocto Project* 1.1, MeeGo* 1.x as well as CE Linux* PR20-PR23 releases for Intel® Atom™ Processor CE4xxx.

Intel® JTAG Debugger for Intel® Atom™ Processor

Please refer to /opt/intel/atom/xdb/2.3.xxx/doc/ and Release_Notes_JTAG.pdf for additional features.

- Support for debugging of Yocto Project* based Linux* software stacks running on Intel® Atom™ Processor E6xx based platforms
- Updated Eclipse* RCP based graphical user interface
- Fixes of outstanding issues
- Flash Writer Plug-in Support for Intel® Atom™ Processor CE41xx and Intel® Atom™ Processor CE42xx

4   System Requirements

4.1.1   Host Software Requirements

The different tools components included have different sets of host Linux* OS versions that they have been validated against. Please look at the individual component's installation guide and release notes after you unpacked and ran the installer for the tool suite distribution

```
> tar -zxvf l_MID_DBG_p_2.3.xxx.tgz
```

for details. The paths are given relative to the installation directory <install-dir>. The default installation directory is /opt/intel unless indicated differently.

- `<install-dir>/composerxe/Documentation/en_US/Release_NotesC.pdf`
- `<install-dir>/composerxe/Documentation/en_US/ipp/ReleaseNotes.htm`
- `<install-dir>/`
`vtune_amplifier_xe_2011/documentation/en/release_notes_amplifier_xe_li`
`nux.pdf`
- `<install-dir>/atom/idb/2.3.xxx/doc/Release_Notes_IDB.pdf`
- `<install-dir>/atom/xdb/2.3.xxx/doc/Release_Notes_JTAG.pdf`

The supported host OS information below is only providing a general overview and represents our actively validated platforms. Please see the component release notes for more details.

- **Intel® C++ Compiler:** Asianux* 3.0, 4.0; Fedora* 12, 13, 14, 15; Red Hat Enterprise Linux* 4, 5, 6; SuSE* Linux* Enterprise Server (SLES) 10, 11; Ubuntu* 10.04, 11.04; Debian* 5.0
- **Intel® Integrated Performance Primitives:** Asianux* 3.0; Fedora* 13, 15; Red Hat Enterprise Linux* 4, 5, 6; SuSE* Linux* Enterprise Server (SLES) 10.2, 11; Ubuntu* 10.04, 11.04; Debian* 6.0
- **Intel® VTune™ Amplifier XE 2011:** Fedora* 12, 13, 14, 15; Asianux* 3, Red Hat* Enterprise Linux* 5, 6; SuSE* Linux* Enterprise Server (SLES) 10.0, 11.0; Red Flag* Linux* 5.0; Ubuntu* 9.10, 10.04, 11.04
- **Intel® Application Debugger:** Fedora* 10 (only partly validated), 14, 15; Ubuntu* 10.04, 11.04
- **Intel® JTAG Debugger:** Fedora* 10 (only partly validated), 14, 15; Ubuntu* 10.04, 11.04

Additional Package Dependencies

A successful install requires the following standard packages to be present: g++, gcc, libc6, libstdc++6, binutils

Sudo or Root Access Right Requirements

- Integration of the Intel® C++ Compiler into the Yocto Project* Application Development Toolkit requires the launch of the tool suite installation script install.sh as root or sudo user.
- Installation of the hardware drivers for the Intel® ITP-XDP3 probe to be used with the Intel® JTAG Debugger requires the launch of the tool suite installation script install.sh as root or sudo user.

Compiler Environment Setup

To setup the environment for the Intel® C++ Compiler and integrate it correctly with the build environment on your Linux host, execute the following command:

```
> source <install-dir>/composerxe/bin/compilervars.sh ia32
```

Security-enhanced Linux*

Security-enhanced Linux* settings (SELinux) are currently not supported by the Intel® VTune™ Amplifier XE 2011 and need to be either disabled or set to permissive for a successful tool suite installation. If your Linux* distribution has SELinux enabled the following error message will be issued by the installer:

```
Your system is protected with Security-enhanced Linux (SELinux).
We currently support only "Permissive" mode, which is not found on the
system.
To rectify this issue, you may either disable SELinux by
- setting the line "SELINUX=disabled" in your /etc/sysconfig/selinux file
- adding "selinux=0" kernel argument in lilo.conf or grub.conf files or make
SELinux mode adjustment by
- setting the line "SELINUX=permissive" in your /etc/sysconfig/selinux file
or ask your system administrator to make SELinux mode adjustment.

You may need to reboot your system after changing the system parameters.
More information about SELinux can be found at http://www.nsa.gov/selinux/
```

4.1.2 Target Software Requirements

The target platform should be based on one of the following environments:

- MeeGo* 1.1 based environment
- MeeGo* 1.1 compliant OS
- MeeGo* 1.x for IVI or Timesys Linux* for Intel® Atom™ Processor E600

4.1.3 Hardware Requirements

- Intel IA-32 architecture based host computer
- Ethernet TCP/IP Connection and ethernet cable
- Development platform based on the Intel® Atom™ processor Z5xx, Z6xx, N2xx, N3xx, N4xx, D4xx, D5xx, Intel® Atom™ processor CE4xxx or the Intel® Media processor CE3100

Please see the validation and support overview matrix for the supported operating systems for your quick reference below:

Host-Target Platform Support Matrix

no not validated

*) partially validated

Table 1. Intel® VTune™ Amplifier XE 2011

Host / Target	MeeGo*	CE FDK
Fedora* 10	No	CExxxx *)
Fedora* 14	Zxxx, N4xx	CExxxx
Fedora* 15	Zxxx, N4xx	CExxxx
Ubuntu* 10.04	N4xx	CExxxx
Ubuntu* 11.04	N4xx	CExxxx

Table 2. Intel(R) Application Debugger

Host / Target	MeeGo*	CE FDK
Fedora* 10	No	CExxxx *)
Fedora* 14	Zxxx *), N4xx *)	no
Fedora* 15	Zxxx, N4xx	CExxxx
Ubuntu* 10.04	No	CExxxx
Ubuntu* 11.04	N4xx	no

Table 3. Intel(R) JTAG Debugger

Host / Target	Z5xx	Z6xx	E6xx	CE3100	CE4xxx
Fedora* 10	no	No	no	McG, XDP *)	McG, XDP *)
Fedora* 14	McG, XDP	McG, XDP	McG, XDP	no	no
Fedora* 15	no	no	McG, XDP	XDP	McG, XDP
Ubuntu* 10.04	no	No	no	XDP *)	McG, XDP
Ubuntu* 11.04	McG, XDP	McG, XDP	McG, XDP	XDP *)	no

5 Installation Notes

Installing the Tool Suite

The default installation directories are:

- `/opt/intel/composerxe/`
- `/opt/intel/composerxe/ipp`
- `/opt/intel/vtune_amplifier_xe_2011/`
- `/opt/intel/atom/idb/2.3.xxx`
- `/opt/intel/atom/xdb/2.3.xxx`

for intel® C++ Compiler, Intel® Integrated Performance Primitives, Intel® VTune™ Amplifier XE 2011, Intel® Application Debugger, and Intel® JTAG Debugger respectively.

The Intel® VTune™ Amplifier XE 2011 as included in this distribution is intended for Intel® Atom™ processor targeted cross-development. It is recommended to install the Intel® VTune™ Amplifier XE 2011 on your software development host system to view and analyze the collected sampling data. It is further recommended to install the Sampling Collector for Intel® VTUne™ Amplifier XE 2011 on your Intel® Atom™ processor based target device

The debuggers are intended for Intel® Atom™ processor targeted cross-development. It is recommended to install the debuggers on your software development host system. It is further recommended to install the idbserver debug server in one of the following locations:

- Intel® Atom™ processor based target device
- Yocto Project* 1.1 virtual image running inside QEMU*
- MeeGo* 1.1 virtual image running inside QEMU*
- MeeGo* Image Creator 2 based jailroot environment

Sudo or Root Access Right Requirements

- Integration of the Intel® C++ Compiler into the Yocto Project* Application Development Toolkit requires the launch of the tool suite installation script install.sh as root or sudo user.
- Installation of the hardware drivers for the Intel® ITP-XDP3 probe to be used with the Intel® JTAG Debugger requires the launch of the tool suite installation script install.sh as root or sudo user.

For installation of the tool suite on the development host please follow the steps below:

1. Unpack the tool suite package in a directory to which you have write access.
   ```
   > tar -zxvf l_MID_DBG_p_2.3.xxx.tgz
   ```
2. If you do not have a license file, please note the product serial number. You will need it to complete the installation process. Otherwise copy the license file you may have

received via email from the Intel® Software Development Products Registration Center to `/opt/intel/licenses/`.

3. It is recommended to register your product at https://registrationcenter.intel.com. If you purchased support for this product you will need to register to take full advantage of Intel Premier Support at https://premier.intel.com.

4. Change into the directory the tar file was extracted to `../l_MID_DBG_p_2.3.xxx`

5. Run the installation script

   Execute the install script in the directory where the tar file was extracted.
   `>./install.sh`

6. If you are not logged in as root, you will be asked if you want to install as root, install as root using sudo, or install without root privileges. Installing as root (using sudo if you have that privilege) is recommended, as that will update the system RPM database. Use the `install as current user` option if you want to install to a private area. To be able to install the Intel® JTAG Debugger however it is necessary to select "install as root" or "install as root using sudo". Without root privileges the option to install the Intel® JTAG Debugger will not be offered during install.

7. The welcome message to the Intel® Embedded Software Development Tool Suite 2.3 for Intel® Atom™ Processor appears along with an outline of the installation process. Press the `Enter` key to continue.

8. The installation routine checks for the availability of all product dependencies. Please take care of these dependencies, if a warning message appears.

9. Afterwards you will be asked to read the end-user license agreement for the tool suite. Press the `Enter` key to continue with reading the license agreement. Once done type `accept` to continue with the installation.

10. When asked whether you would like to activate and install your product select one of the options provided depending on whether you have a license file available or not. If there is already a valid license file available and installed on your system, the installation routine will recommend to simply use the existing license file. If you do not have access to the internet at the time of installation, select the alternative activation option.

11. The next screen let's you review your installation options. If you would like to only install the Intel® JTAG Debugger, select [3] and change the components settings. Else, continue with the default choice [1] to start the installation.

```
Step no: 4 of 6 | Options
-----------------------------------------------------------------------------
You are now ready to begin installation. You can use all default installation
settings by simply choosing the "Start installation Now" option or you can
customize these settings by selecting any of the change options given below
first. You can view a summary of the settings by selecting
"Show pre-install summary".
-----------------------------------------------------------------------------
1. Start installation Now
```

```
2. Change install directory      [ /opt/intel ]
3. Change components to install  [ All ]
4. Show pre-install summary
h. Help
b. Back to the previous menu
q. Quit
--------------------------------------------------------------------------
Please type a selection or press "Enter" to accept default choice [1]:
```

12. During the prerequisites check the presence of the Yocto Project* Application
    Development Toolkit 1.1 (ADT) will be checked. If it is not present on your installation
    you will get a warning message.

    The Yocto Project* Application Development Toolkit is required if you want to use the
    Intel® Composer XE for building Yocto Project* targeted applications. For automatic
    Intel® Composer XE integration with the Application Development Toolkit during
    installation, please install Toolkit first and re-check the prerequisites.

    If you do not intend to develop applications targeting Yocto Project* builds, you can
    ignore this warning message.

13. The installation will be finalized and you will be informed when the installation has been
    completed successfully.

```
Step no: 6 of 6 | Complete
--------------------------------------------------------------------------
Thank you for installing and for using the
Intel(R) Embedded Software Development Tool Suite for Intel(R) Atom(TM)
processor version 2.3.
Support services start from the time you install or activate your product.
If you have not already done so, please create your support account now to
take full advantage of your product purchase.
Your support account gives you access to free product updates and upgrades
as well as interactive technical support at Intel(R) Premier Support.
To create your support account, please visit the Subscription Services web
site
https://registrationcenter.intel.com/RegCenter/registerexpress.aspx?media=WJP
--------------------------------------------------------------------------
q. Quit [default]
--------------------------------------------------------------------------
Please type a selection or press "Enter" to accept default choice [q]:
```

14. To setup the environment for the Intel® C++ Compiler and integrate it correctly with the
    build environment on your Linux host, execute the following command:

```
> source <install-dir>/composerxe/bin/compilervars.sh ia32
```

15. Below is a list of ready to go target connection configurations for the Intel® JTAG Debugger:

xdb_CE3100.sh	Intel® Media Processor CE31xx	Intel® ITP-XDP
xdb_CE4100_mcrg.sh	Intel® Atom™ Processor CE41xx	Macraigor* usb2Demon
xdb_CE4100.sh	Intel® Atom™ Processor CE41xx	Intel® ITP-XDP
xdb_CE4200_mcrg.sh	Intel® Atom™ Processor CE42xx	Macraigor* usb2Demon
xdb_CE4200.sh	Intel® Atom™ Processor CE42xx	Intel® ITP-XDP
xdb_E600_mcrg.sh	Intel® Atom™ Processor E6xx	Macraigor* usb2Demon
xdb_E600.sh	Intel® Atom™ Processor E6xx	Intel® ITP-XDP
xdb_Z500_mcrg.sh	Intel® Atom™ Processor Z5xx	Macraigor* usb2Demon
xdb_Z500.sh	Intel® Atom™ Processor Z5xx	Intel® ITP-XDP
xdb_Z600_mcrg.sh	Intel® Atom™ Processor Z6xx	Macraigor* usb2Demon
xdb_Z600.sh	Intel® Atom™ Processor Z6xx	Intel® ITP-XDP

To have the installer define a default xdb.sh configuration start script that points to one of these, please consult the advanced installation options.

Compiler Integration with Yocto Project* 1.1 Application Development Toolkit

5.1.1 Yocto Project* Application Development Toolkit (ADT) Installer

Please refer to the Yocto Project* Application Development Toolkit User's Guide at http://www.yoctoproject.org/docs/current/adt-manual/adt-manual.html for details on the Application Development Toolkit Usage.

To be able to use the Application Development Toolkit with the Intel® C++ Compiler it has to be installed using the Application Development Toolkit tarball and installer script available at http://www.yoctoproject.org/downloads/yocto-1.1/adt-installer/. More detailed information on this can be found at http://www.yoctoproject.org/docs/current/adt-manual/adt-manual.html#using-the-adt-installer. Other installation methods are not supported.

After unpacking the installer tarball you will find a file `adt_installer.conf`. This file needs to be modified to support Intel® architecture before the Application Development Toolkit can be used with the Intel® C++ Compiler.

5.1.2 Modifying the Application Development Toolkit Installer Configuration

In the file adt_installer.conf the following modifications are necessary:

1. Comment out lines 44, 46, and 48

   #YOCTOADT_ROOTFS_arm="minimal sato-sdk"

   #Specify which root filesystem file to use to extract as target sysroot. Please ensure the #entry is in the list of downloaded root filesystem files that specified above in #YOCTOADT_ROOTFS_$arch

   #YOCTOADT_TARGET_SYSROOT_IMAGE_arm="minimal"

   #The location where the target sysroot will be setup

   #YOCTOADT_TARGET_SYSROOT_LOC_arm="$HOME/test-yocto/arm"

2. Uncomment lines 51, 52, and 53

   YOCTOADT_ROOTFS_x86="minimal-dev"

   YOCTOADT_TARGET_SYSROOT_IMAGE_x86="minimal-dev"

   YOCTOADT_TARGET_SYSROOT_LOC_x86="$HOME/test-yocto/x86"

The important changes are that the three entries

YOCTOADT_ROOTFS_x86

YOCTOADT_TARGET_SYSROOT_IMAGE_x86

 YOCTOADT_TARGET_SYSROOT_LOC_x86

need to be defined.

These entries are used to configure the Application Development Toolkit environment and need to be present for the Intel® Embedded Software Development Tool Suite for Intel® Atom™ processor to integrate into this environment

5.1.3   Run the Application Development Toolkit Installer

Now the Application Development Toolkit can be installed. Simply execute

```
$ adt_installer &
```

from the unpacked installer tarfile and follow the instructions.

5.1.4   Sudo or Root Access Right Requirement

Integration of the Intel® C++ Compiler into the Yocto Project* Application Development Toolkit requires the launch of the tool suite installation script install.sh as root or sudo user.

5.1.5   Automatic Integration during Tool Suite Installation

The tool suite installation script `install.sh` will automatically check against the presence of the Yocto Project* 1.1 Application Development Toolkit framework. It will look for the Poky Linux* install in `/opt/poky/1.1/` and in addition it will look for a build target configuration at a directory specified by the `environment-setup-i586-poky-linux` file in `/opt/poky/1.1/`.

The installation then creates a backup copy of the existing `environment-setup-i586-poky-linux` file and replaces it with a file customized for the use of the Intel® C++ Compiler. Thus the integration can be reversed simply by copying the original `environment-setup-i586-poky-linux` back into its place.

Additionally the installation script copies a file yocto.env that defines the target platform settings for the Intel® C++ Compiler into the `/opt/intel/composer_xe_2011_sp1.6.233/bin/ia32/` directory.

If no Yocto Project* 1.1 Application Development Toolkit is found the following warning will be issued during installation:

```
The Yocto Project* Application Development Toolkit has not been detected on
your system. This toolkit is required if you want to use the Intel® Composer
```

XE for building Yocto Project* targeted applications. For automatic Intel®
Composer XE integration with the Application Development Toolkit during
installation, please install Toolkit first and re-check the prerequisites.
For manual integration after installation, please consult the product Release
Notes.

To use the Intel® C++ Compiler for a Yocto Project* target simply call the `icc` or `icpc` Intel®
C++ Compiler drivers with the option `–platform=yocto` from your application build
environment that you already set up for usage with the Yocto Project* 1.1 Application
Development Toolkit.

5.1.5.1 Setup of Compiler Build Environment

To setup the environment for the Intel® C++ Compiler and integrate it correctly with the build
environment on your Linux host, execute the following command:

```
> source <install-dir>/composerxe/bin/compilervars.sh ia32
```

To setup the environment for the Yocto Project* Application Development Toolkit execute the
following command:

```
> source /opt/poky/1.1/environment-setup-i586-poky-linux
```

5.1.6 Manual Compiler Integration with the Application Development Toolkit

1. Replace the existing file `/opt/poky/1.1/environment-setup-i586-poky-linux`
   with one the resembles the following entries:

```
export PATH=/opt/poky/1.1/sysroots/i686-pokysdk-
linux/usr/bin:/opt/poky/1.1/sysroots/i686-pokysdk-
linux/usr/bin/i586-poky-linux:$PATH
export PKG_CONFIG_SYSROOT_DIR=<CURRENT_USER>/test-yocto/x86
export PKG_CONFIG_PATH=<CURRENT_USER>/test-
yocto/x86/usr/lib/pkgconfig
export CONFIG_SITE=/opt/poky/1.1/site-config-i586-poky-linux
export CC=icc
export CXX=icpc
export GDB=i586-poky-linux-gdb
export TARGET_PREFIX=i586-poky-linux-
export CONFIGURE_FLAGS="--target=i586-poky-linux --host=i586-
poky-linux --build=i686-linux --with-libtool-
sysroot=<CURRENT_USER>/test-yocto/x86"
export CFLAGS=" -march=i586 -platform=yocto"
```

```
export CXXFLAGS=" -march=i586 -platform=yocto"
export LDFLAGS="  --sysroot=<CURRENT_USER>/test-yocto/x86"
export CPPFLAGS=" -platform=yocto"
export OECORE_NATIVE_SYSROOT="/opt/poky/1.1/sysroots/i686-
pokysdk-linux"
export OECORE_TARGET_SYSROOT="<CURRENT_USER>/test-yocto/x86"
export OECORE_ACLOCAL_OPTS="-I /opt/poky/1.1/sysroots/i686-
pokysdk-linux/usr/share/aclocal"
export OECORE_DISTRO_VERSION="1.1"
export OECORE_SDK_VERSION="1.1"
export YOCTO_TOOLCHAIN=/opt/poky/1.1/sysroots/i686-pokysdk-
linux/usr/bin
export YOCTO_SYSROOT=<CURRENT_USER>/test-yocto/x86
```

The –sysroot compile flags from the original `environment-setup-i586-poky-linux` can
be removed because their function is covered by the `yocto.env` file.

2. Create a `yocto.env` file in `/opt/intel.composerxe/bin/` with the following
    contents:

```
*platform:
  yocto

*yocto_sdk_toolchain:
  %$(YOCTO_TOOLCHAIN)

*sysroot:
  %$(YOCTO_SYSROOT)

*target_root:
  %(sysroot)

*gcc_install:
  %(sysroot)/usr/lib/gcc/i586-poky-linux/4.6.1

*intel_include:
  %(intel_root)/../compiler/include

*intel_lib:
  %(intel_root)/../compiler/lib/ia32

*exec_path:
  %(yocto_sdk_toolchain)/i586-poky-linux

*exec_prefix:
  i586-poky-linux-

*gxx_include:
  %(sysroot)/usr/include/c++

*link_lib_path:

%(intel_lib)%(path_separator)%(gcc_install)%(path_separator)%(sysroot)/
```

```
lib%(path_separator)%(sysroot)/usr/lib%(path_separator)%(sysroot)/usr/l
ib/i586-poky-linux/4.6.1

*link_start_files:

%{static?%{p?%(sysroot)/usr/lib/gcrt1.o;%(sysroot)/usr/lib/crt1.o};%{!s
hared?%(sysroot)/usr/lib/crt1.o}} %(sysroot)/usr/lib/crti.o
%{static?%(sysroot)/usr/lib/i586-poky-
linux/4.6.1/crtbeginT.o;%{shared?%(sysroot)/usr/lib/i586-poky-
linux/4.6.1/crtbeginS.o;%(sysroot)/usr/lib/i586-poky-
linux/4.6.1/crtbegin.o}}

*link_end_files:
  %{!static?%{shared?%(sysroot)/usr/lib/i586-poky-
linux/4.6.1/crtendS.o;%(sysroot)/usr/lib/i586-poky-
linux/4.6.1/crtend.o};%(sysroot)/usr/lib/i586-poky-
linux/4.6.1/crtend.o} %(sysroot)/usr/lib/crtn.o

*link_default_libs:
%{!static?%{i-dynamic|shared?-Bdynamic;-Bstatic}} -lsvml -limf \
  %{!static?-Bdynamic} -lm \
  %{!static?%{i-dynamic|shared?-Bdynamic;-Bstatic}} -lipgo -ldecimal \
  %{i_cxxlink? \
    %{cxxlib-gcc? \
      %{!static?%{i-static|static-libcxa?-Bstatic;-Bdynamic}} -
lcxaguard}} \
  %{openmp-stubs?%{!static?%{i-static?-Bstatic;-Bdynamic}} -lompstub} \
  %{!static?%{i-dynamic|shared?-Bdynamic;-Bstatic}} %{pic-libirc?-
lirc_pic;-lirc} \
  %{!static?-Bdynamic} -lc \
  %{cxxlib-gcc? \
    %{!cxxlib-nostd?%{!static?-Bdynamic} -lstdc++;%{!static?-Bdynamic}
-lsupc++} \
    %{static|static-libgcc? \
      %{!static?-Bstatic} -lgcc -lgcc_eh; \
        %{!shared?%{!static?%{static-libgcc?-Bstatic;-Bdynamic}} -lgcc
-lgcc_s}} \
    %{!static?-Bdynamic} -ldl -lc}
```

To setup the environment for the Intel® C++ Compiler and integrate it correctly with the build environment on your Linux host, execute the following command:

```
> source <install-dir>/composerxe/bin/compilervars.sh ia32
```

To setup the environment for the Yocto Project* Application Development Toolkit execute the following command:

```
> source /opt/poky/1.1/environment-setup-i586-poky-linux
```

Installing the Debugger Remote Server

To install the Intel® Debugger Remote Server for IA32 Linux* go its installation directory in the Intel® Application Debugger 2.3 for Intel® Atom™ Processor installation: `/opt/intel/atom/idb/2.3.xxx/bin/ia32`.

Copy the file `idbserver` onto your cross-debug target device or target virtual machine via sftp.

Please see more details on this in the Intel® Application Debugger 2.3 for Intel® Atom™ Processor release notes `Release_Notes_IDB.pdf`

Installing Intel® XDP3 JTAG Probe

If the install.sh installation script is executed using root access, su or sudo rights, the required drivers will be installed automatically. Root, su or sudo rights are required for the installation

Installing Macraigor Systems* usb2Demon* Support

To enable support for the Macraigor Systems* usb2Demon* device for debugging Intel® Atom™ processor based platforms with the Intel® JTAG Debugger it is necessary to install the Linux* drivers for the Macraigor Systems* usb2Demon* device. The driver can be found at http://www.macraigor.com/full_gnu.htm. Please scroll down all the way and download and install the Linux32 OCDRemote package recommended at http://www.macraigor.com/intel/ . Follow the installation instructions posted at the same location closely.

1. For Fedora* host installations download mcgr-hwsupport-x.x-x.i386.rpm from http://www.macraigor.com/Intel/. For Ubuntu* host installations you can download mcgr-hwsupport-x.x-x.i386.deb from the same location. You will find the latest recommended version of the OCDRemote* driver set referenced there.

*   Support for the Intel® Atom™ processor Z5xx will require at least version 7.9-0.
*   Support for the Intel® Atom™ processor Z6xx will require at least version 8.3-0.

- Support for the Intel® Atom™ processor E6xx will at least require version 8.7-0.
- Support for the Intel® Atom™ processor CE4xxx will at least require version 9.3-1

We recommend the use of at least version 9.4-0 regardless of your hardware target.

Ensure you have root or sudo access rights on your Linux* installation.

2. Install mcgr-hwsupport-x.x-x.i386.rpm using the `rpm -i` command or mcgr-hwsupport-x.x-x.i386.deb using the `dpkg -install` command .
3. You should now be able to launch the Intel(R) JTAG Debugger using the respective xdb_processorname_mcrg.sh startup script. These scripts are being put into the /opt/intel/atom/xdb/2.3.xxx/bin directory by the tool suite installer as described in the general installation section above.

For further details on how to configure the OCDRemote* driver set from Macraigor* Systems, please refer to the full installation instructions at
http://www.macraigor.com/eclipse/eclipse_faq.htm.

Installing the Sampling Collector for Intel® VTune™ Amplifier XE 2011 on the target platform

1. Copy the Sampling Collector for Intel® VTune™ Amplifier XE 2011 located at

   `~/l_MID_DBG_p_2.3.xxx/rpm/sep34_axeu5_lin_ia32.tar.gz`

   onto the Intel® Atom™ Processor based target device running Linux* or MeeGo*

2. On the target device unpack the sampling collector using the following command:

   `> tar –xvzf sep34_axeu5_lin_ia32.tar.gz`

5.1.7   Installing the Sampling Driver

Installing the sampling driver may require using a *development* and *target* system:

- The development system is the system you use to develop your application. This system has full binutils and a compiler.
- The target system is the system for which the application is targeted.

These are the steps for installing the Sampling driver:

1. Go to the /sepdk source directory:
   cd *<install-dir>*/sepdk/src
2. Build the sampling driver.
   - For a target system that has full binutils and a compiler, run the following command.
     ./build-driver &endash;ni &endash;-install-dir=../prebuilt
   - For a target system that does not have full binutils and a compiler, you need to compile the driver on your development system, with the settings for your target system:
     1. Customize the cc-sep3-driver script, included in the installation, for your kernel, compiler and other system settings.
     2. Run the customized script to build the driver on your development system.
     3. Copy the newly-built driver to the target system.
3. If the driver was built successfully, go to the directory where it was installed:
   cd *<install-dir>*/sepdk/prebuilt
4. To load the sampling driver, use the following command:

   ./insmod-sep3 -r -g [*user_group*]
   where [*user_group*] is the name of the group that can perform sampling collection on the system. To find which group you are a member of, run the command:
   groups

5. To have the system automatically load the sampling driver at boot time, run the following command:

   ./boot-script -i -g [user_group]
6. To check whether sampling driver successfully loaded, run the command:

   ./insmod-sep3 -q
7. To unload the driver, use the command:

   ./rmmod-sep3

5.1.8 Customizing the Driver Script

This topic explains the process for customizing the cc-sep3-driver script for your kernel, compiler and other system settings:

1. Open the cc-sep3-driver script in a text editor.
2. Edit the lines to set the various variables according to your development and target system:
   - Development system settings:
     - Base directory
     - Tool directory where the compiler and linker reside
     - Name of the compiler that compiles the driver
   - Target system settings:
     - Platform architecture
     - kernel arity
     - kernel version
     - kernel directory

   Save the customized script.

   Continue Installing the Sampling Driver.

5.1.9 Setting Up the SEP Runtime Environment

Set up the SEP runtime environment by completing the following steps:

1. Run the following command-line:

   cd *<install-dir>*/bin
   sh
   source setup_sep_runtime_env.sh
2. Start using SEP:

   cd $HOME
   sep -version
   sep -el

```
sep -start -d 20 -info 4 -verbose -out foo
sfdump5 foo.tb5 -processes
```

Installing the Sampling Collector for Intel® VTune™ Amplifier XE 2011 on Yocto Project* based target platform

Installation of the Sampling Collector for Intel® VTune™ Amplifier XE 2011 on a Yocto Project* based target platform follows the same steps as described in the previous chapter titled Installing the Sampling Collector for Intel® VTune™ Amplifier XE 2011.

Building the driver kernel module for the sampling collector requires kernel sources and kernel header files for the Yocto Project*.

5.1.10 Setting up Yocto Project* Build Environment on Build Host

Please refer to the Yocto Project* homepage at http://www.yoctoproject.org and the Yocto Project* Developer Manual at http://www.yoctoproject.org/docs/current/dev-manual/dev-manual.html for the latest information and for details not covered in these release notes.

To build the driver module for the Sampling Collector for Intel® VTune™ Amplifier XE 2011 requires the presence of a fully configured Yocto Project* kernel build environment.

The Yocto Project* Quick Start Guide at http://www.yoctoproject.org/docs/current/yocto-project-qs/yocto-project-qs.html outlines everything needed to create an OS kernel build environment using bitbake.

The steps below describe some of the key steps of this setup for Yocto Project* 1.1:

1. Download the Yocto Project* 1.1 source package:

   http://downloads.yoctoproject.org/releases/yocto/yocto-1.1/poky-edison-6.0.tar.bz2

   and unpack it under a non-root user account

2. On a regular user command shell, run the following:

   source <install_dir>/poky-edison-6.0/oe-init-build-env edison-6.0-build

   This will create an "edison-6.0-build" folder in the current directory

3. Go to the newly created folder and edit the local.conf file

   cd edison-6.0-build

   vi conf/local.conf

   Ensure the following settings in local.conf:

   POKY_EXTRA_INSTALL = "dropbear" // this installs SSH server/client

   BB_NUMBER_THREADS = "N" // where "N" is the number of physical cores

```
PARALLEL_MAKE = "-j N"        // where "N" is the number of physical cores

INHERIT += "rm_work"          // set this if you have less than 50GB free space available
```

4. Get latest GIT configurations for Intel® Architecture based platforms.

```
cd $HOME/yocto

git clone git://git.yoctoproject.org/meta-intel/
```

This will create a "meta-intel" folder in the current directory

5. Configure GIT to use Edison build

```
cd meta-intel

git checkout edison
```

6. Configure bitbake to use Intel® Architecture specific configurations.

```
cd edison-6.0-build

vi conf/bblayers.conf
```

Add "meta-intel/meta-XYZ" (XYZ=machinetype) to BBLAYERS variable

5.1.11 Installing the Sampling Driver

Installing the sampling driver may require using a *development* and *target* system:

- The development system is the system you use to develop your application. This system has full binutils and a compiler.
- The target system is the system for which the application is targeted.

These are the steps for installing the Sampling driver:

8. Go to the /sepdk source directory:

   cd *<install-dir>*/sepdk/src
9. Build the sampling driver.
   - Compile the driver on your development system, with the settings for your target system:
     1. [Customize the cc-sep3-driver script](), included in the installation, for your kernel, compiler and other system settings.
     2. Run the customized script to build the driver on your development system.
     3. Copy the newly-built driver to the target system.
10. If the driver was built successfully, go to the directory where it was installed:

    cd *<install-dir>*/sepdk/prebuilt
11. To load the sampling driver, use the following command:

    ./insmod-sep3 -r -g [*user_group*]

    where [*user_group*] is the name of the group that can perform sampling collection on the system. To find which group you are a member of, run the command:

    groups
12. To have the system automatically load the sampling driver at boot time, run the following command:

    ./boot-script -i -g [user_group]
13. To check whether sampling driver successfully loaded, run the command:

    ./insmod-sep3 -q
14. To unload the driver, use the command:

    ./rmmod-sep3

5.1.12 Customizing the Driver Script for Yocto Project* Targets

This topic explains the process for customizing the cc-sep3-driver script for your kernel, compiler and other system settings:

5. Open the cc-sep3-driver script in a text editor.
6. Edit the lines to set the various variables according to your development and target system. Below are example settings for Yocto Project* 1.1:

- Development system settings:
  - Base directory

    BASE=$HOME/yocto/edison-6.0-build/tmp

  - Tool directory where the compiler and linker reside

    TOOLS=$BASE/work/core2-poky-linux/gcc-4.6.1+svnr175454-
    r10/package/usr/bin

  - Name of the compiler that compiles the driver

    CC=gcc

- Target system settings:
  - Platform architecture

    PLATFORM=x32

  - kernel arity

    ARITY=smp

  - kernel version

    KERNEL_VERSION=3.0.4-yocto-standard+

  - kernel directory

    KERNEL_DIR=$BASE/sysroots/fri2-noemgd/kernel

Save the customized script.

Running "./cc-sep3-driver" in the sepdk/src directory built the SEP drivers using the kernel toolchain.

Continue Installing the Sampling Driver.

5.1.13 Setting Up the SEP Runtime Environment

Set up the SEP runtime environment by completing the following steps:

3. Run the following command-line:

   ```
   cd <install-dir>/bin
   sh
   source setup_sep_runtime_env.sh
   ```

4. Start using SEP:

   ```
   cd $HOME
   sep -version
   sep -el
   sep -start -d 20 -info 4 -verbose -out foo
   sfdump5 foo.tb5 -processes
   ```

Removing the Product

- To uninstall the complete Intel® Embedded Software Development Tool Suite for Intel® Atom™ Processor, change to the `/opt/intel/atom` directory and run the `uninstall_atom.sh` uninstall script

- To uninstall the Intel® Composer XE 2011 containing the Intel® C++ Compiler and the Intel® Integrated Performance Primitives go to the `/opt/intel/composerxe/pkg_bin` directory and run the `uninstall.sh` uninstall script

- To uninstall the Intel® VTune™ Amplifier XE 2011 go to the `/opt/intel/vtune_amplifier_xe_2011` directory and run the `uninstall.sh` uninstall script.

- To uninstall the Intel® Application Debugger go to the `/opt/intel/atom/idb/2.3.009/bin` directory and run the `uninstall_dbg.sh` script.

- To uninstall the Intel® JTAG Debugger go to the `/opt/intel/atom/xdb/2.3.009/bin` directory and run the `uninstall_sbg.sh` script.

6   Tool Suite Usage

The Intel® Embedded Software Development Tool Suite supports a wide variety of Intel® Atom™ processor based devices. This means that there is not one single use case or usage model that will fit the needs of every developer. Development for Intel® Atom™ processor based devices tends to have one aspect in common however. In most cases you will want to take advantage of the performance and the high screen resolution of your regular development environment and deploy your build to the real hardware for validation and analysis only. In short, most commonly the usage model will be one of cross development.

The Intel® Embedded Software Development Tool Suite for Intel® Atom™ processor addresses this in multiple ways.

- The Intel® C++ Compiler and the Intel® Integrated Performance Primitives will by default be installed on the development host system. To provide a protected build environment without host system library pollution it is however also supported to install these components into Image Creator, the MeeGo* SDK jailroot system or to install them into a Yocto Project* or MeeGo* developer image running inside a QEMU* virtual machine or even on a real target.
- The Intel® Application Debugger will with the standard installation package be installed on your development host. Via TCP/IP you can connect to and debug a process either running on actual Intel® Atom™ processor based target hardware or a QEMU* virtual machine or some other virtualization method (including chroot based simulation) that assures you that the application to be debugged runs in a software environment identical to the later application deployment. Please also have a look at the Getting Started Guide Getting_Started.html that is part of the tool suite distribution for further details on the Intel® Application Debugger usage.
- The Intel® JTAG Debugger will with the standard installation package be installed on your development host. Via a JTAG device you can connect to your target platforms eXtended Debug Port (XDP) and thus commence remote cross-debugging of your entire platform and system level software.
- The Intel® VTune™ Amplifier XE 2011 analyzes event based and time based sampling data on the development host. The data has been gathered on an Intel® Atom™ processor based target platform using the low overhead SEP Sampling Collector giving you accurate sampling data. The analysis can be done within the standard KDE* or Gnome* based standard Linux* GUI you are used to for your development efforts.

7   Issues and Limitations

Known Issues and Limitations

For known issues of individual Intel® Embedded Software Development Tool Suite for Intel® Atom™ Processor components please refer to the individual component release notes. Their location in the installed product can be found in chapter 2:

Technical Support and Documentation

Sampling Collector for Intel® VTune™ Amplifier XE 2011 Usage with Yocto Project*

7.1.1   Building Sampling Collector driver on host Linux* system:
Additional kernel utilities are required for building drivers and kernel modules need to be present in the kernel source tree. The following utilities need to be manually added to the standard Yocto Project* 1.1 kernel build tree: viz, recordmcount, fixdep, and modpost

7.1.2   Specification of sampling duration using –d option not supported for Yocto Project* target
When a sampling duration is specified using the –d <time> option, the shell running SEP terminates.

8   Attributions

This product includes software developed at:

The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- software copyright (c) 1999, IBM Corporation., http://www.ibm.com.
- software copyright (c) 1999, Sun Microsystems., http://www.sun.com.
- the W3C consortium (http://www.w3c.org) ,
- the SAX project (http://www.saxproject.org)
- voluntary contributions made by Paul Eng on behalf of the
Apache Software Foundation that were originally developed at iClick, Inc.,
software copyright (c) 1999.

This product includes updcrc macro,
Satchell Evaluations and Chuck Forsberg.
Copyright (C) 1986 Stephen Satchell.

This product includes software developed by the MX4J project
(http://mx4j.sourceforge.net).

This product includes ICU 1.8.1 and later.
Copyright (c) 1995-2006 International Business Machines Corporation and others.

Portions copyright (c) 1997-2007 Cypress Semiconductor Corporation.
All rights reserved.

This product includes XORP.
Copyright (c) 2001-2004 International Computer Science Institute

This product includes software licensed from Macraigor Systems, LLC.
Copyright (c) 2004-2009, Macraigor Systems LLC. All rights reserved.

This product includes software from the book
"Linux Device Drivers" by Alessandro Rubini and Jonathan Corbet,
published by O'Reilly & Associates.

This product includes hashtab.c.
Bob Jenkins, 1996.

9 Disclaimer and Legal Information

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL 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.
UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.
Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.
The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.
Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.
Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. Go to: http://www.intel.com/products/processor_number/

MPEG-1, MPEG-2, MPEG-4, H.261, H.263, H.264, MP3, DV, VC-1, MJPEG, AC3, AAC, G.711, G.722, G.722.1, G.722.2, AMRWB, Extended AMRWB (AMRWB+), G.167, G.168, G.169, G.723.1, G.726, G.728, G.729, G.729.1, GSM AMR, GSM FR are international standards promoted by ISO, IEC, ITU, ETSI, 3GPP and other organizations. Implementations of these standards, or the standard enabled platforms may require licenses from various entities, including Intel Corporation.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino Inside, Cilk, Core Inside, i960, Intel, the Intel logo, Intel AppUp, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, Intel Inside logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel Sponsors of Tomorrow., the Intel Sponsors of Tomorrow. logo, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, InTru, the InTru logo, InTru soundmark, Itanium, Itanium Inside, MCS, MMX,

Moblin, Pentium, Pentium Inside, skoool, the skoool logo, Sound Mark, The Journey Inside, vPro Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

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

Microsoft, Windows, Visual Studio, Visual C++, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.