Intel® System Studio 2013 for Linux* OS Installation Guide and Release Notes

Installation Guide and Release Notes

Document number: 322069-014US

24 June 2013

Contents

1 Introduction

This document provides a brief overview as well as usage information for the Intel® System Studio 2013 for Linux* OS and provides pointers to where you can find additional product information, technical support, articles and whitepapers.

It also explains how to install the Intel® System Studio product. Installation is a multi-step process. Please read this document in its entirety before beginning and follow the steps in sequence.

The Intel® System Studio consists of multiple components for developing, debugging, tuning and deploying system and application code targeted towards embedded designs.

The tool suite covers several different use cases targeting development for embedded intelligent system platforms ranging from Intel® Atom™ Processor based low-power embedded platforms to 2nd and 3rd generation Intel® Core™ microarchitecture based designs for high throughput data processing and signal processing. Accordingly you may want to install different components of the tool suite in different ways. This document covers the aspects of the Intel® System Studio that can be commonly applicable. The file embedded-usage.pdf focuses on low-power Intel® Atom™ Processor targeted development environment setup and sp-usage.pdf focuses on Intel® Core™ based development environment setup targeting primarily signal processing use cases.

Due to the nature of this comprehensive integrated software development tools solution, different Intel® System Studio components may be covered by different licenses. Please see the licenses included in the distribution as well as the Disclaimer and Legal Information section of these release notes for details.

Change History

This section highlights important from the previous product version and changes in product updates. For information on what is new in each component, please read the individual component release notes.

Update 1 (Intel® System Studio 2013 Update 1 for Linux*)

- Intel® C++ Compiler Support for cross-development targeting Wind River* Linux* 4

Intel® Software Manager

The installation now provides an Intel® Software Manager to provide a simplified delivery mechanism for product updates and provide current license status and news on all installed Intel software products.

You can also volunteer to provide Intel anonymous usage information about these products to help guide future product design. This option, the Intel® Software Improvement Program, is not enabled by default – you can opt-in during installation or at a later time, and may opt-out at any time. For more information please see http://intel.ly/SoftwareImprovementProgram.

Product Contents

The product contains the following components

1. Intel® C++ Compiler 13.0 for applications running on Embedded OS Linux*
2. Intel® Integrated Performance Primitives 7.1 Update 1 for Linux*
3. Intel® Math Kernel Library 11.0 Update 1 (Intel® 64 only)
4. Intel® VTune™ Amplifier 2013 for Systems
    4.1. Intel® VTune™ Amplifier Sampling Enabling Product (SEP) 3.10
5. Intel® Inspector 2013 for Systems
6. The GNU* Project Debugger – GDB v7.5
7. Intel® JTAG Debugger 3.0
8. SVEN 1.0

2 Technical Support and Documentation

Release Notes Location

The release notes and getting started guide for the tools components making up the Intel® System Studio product can be found at the following locations after unpacking l_cembd_p_2013.1.xxx.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® System Studio Product Guide

- `<install-dir>/system_studio_2013.1.xxx/documentation/embedded_compsupdoc_l.pdf`

Intel® System Studio Getting Started Guide

- `<install-dir>/system_studio_2013.1.xxx/documentation/embedded_gsdoc_l.htm`

Intel® System Studio Release Notes

- `<install-dir>/system_studio_2013.1.xxx/documentation/all-release-install.pdf`

Embedded Usage

- `<install-dir>/system_studio_2013.1.xxx/documentation/embedded-usage.pdf`

Signal Processing Usage

- `<install-dir>/system_studio_2013.1.xxx/documentation/sp-usage.pdf`

Intel® C++ Compiler

- `<install-dir>/system_studio_2013.1.xxx/documentation/compiler-release.pdf`

Intel® Integrated Performance Primitives

- `<install-dir>/system_studio_2013.1.xxx/ documentation/en_US/ipp/ReleaseNotes.htm`

Intel® Math Kernel Libraries

- `<install-dir>/system_studio_2013.1.xxx/ documentation/en_US/mkl/ReleaseNotes.htm`

Intel® JTAG Debugger

- `<install-dir>/system_studio_2013.1.xxx/ documentation/en_US/debugger/xdb/jtag-release-install.pdf`

SVEN SDK

- `<install-dir>/system_studio_2013.1.xxx/documentation/en_US/debugger/sven/sdk/SVEN_SDK.pdf`

SVEN Trace Viewer

- `<install-dir>/system_studio_2013.1.xxx/documentation/en_US/debugger/sven/viewer/SVEN_VIEWER.pdf`

Debugging with GDB

- `<install-dir>/system_studio_2013.1.xxx/documentation/en_US/debugger/gdb/GDB.pdf`

Intel® VTune™ Amplifier

- `<install-dir>/system_studio_2013.1.xxx/vtune_amplifier_2013_for_systems/documentation/en/release_notes_amplifier_linux.pdf`

- `<install-dir>/vtune_amplifier_2013_for_systems/documentation/en/remote-target-collection.pdf`

Intel® Inspector

- `<install-dir>/system_studio_2013.1.xxx/inspector_2013_for_systems/documentation/en/Release_Notes_InspectorXE_Linux.pdf`

Intel® VTune™ Amplifier Data Collector

- `<install-dir>/system_studio_2013.1.xxx/vtune_amplifier_2013_for_systems/documentation/en/remote-target-collection.pdf`

Intel® VTune™ Amplifier Sampling Enabling Product

- The user's guide explaining the usage of the SEP command line tool for hardware event-based sampling collection on embedded devices can be found at `<install-dir>/system_studio_target/sep_3.10_axeu3_linux_<arch>/SEP_User_Guide.pdf` after unpacking the system_studio_target.tgz archive.

  The installation directory <install-dir> for the Sampling Collector for Intel® VTune™ Amplifier 2013 is the directory into which the installation tar archive file system_studio_target.tgz has been extracted.
  The architecture designation <arch> can take the values ia32 or x64.

Article & Whitepaper Locations

For a list of available articles, whitepapers and related resources please visit the Intel® System Studio product page at http://software.intel.com/en-us/intel-system-studio and look at the Support tab.

Support

If you did not register your compiler during installation, please do so at the Intel® Software Development Products Registration Center. Registration entitles you to free technical support, product updates and upgrades for the duration of the support term.

To submit issues related to this product please visit the Intel Premier Support webpage and submit issues under the product **Intel(R) System Studio for Linux* OS**.

Additionally you may submit questions and browse issues in the Intel® System Studio User Forum.

For information about how to find Technical Support, product documentation and samples, please visit http://software.intel.com/en-us/intel-system-studio.

Note: If your distributor provides technical support for this product, please contact them for support rather than Intel.

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, SSE3, and SSSE3 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® System Studio has a set of features specifically targeting cross-development needs and also adds support for new Intel® Architecture based hardware platforms.

Cross-Development Support

Support for Poky-Linux* and Application Development Toolkit (ADT) based cross-development targeting Yocto Project*

- Sysroot support
- Compiler integration with Poky-Linux* based makefiles in GNU* cross-compilation environment.
- Compiler runtime library package for target deployment included

Support for cross-development targeting Wind River* Linux* 4, 5 and CE Linux* PR28

Eclipse* IDE integration for cross-build configurations

- Intel® C++ Compiler Integration into Eclipse* 3.7, Eclipse* CDT 8.0 (Indigo)
- Support for editing cross build Intel® C++ Compiler environment files from within Eclipse* IDE

Command-line Power and Performance Analysis with call trace on resource limited targets

- Tune application and whole system software performance with minimal overhead and memory footprint
- Analyze power and performance data via command line on the target device or in the full Intel® VTune™ Amplifier GUI on the host.

Support for Development in Virtualized Environments with Real-Time Requirements

- Tool Suite components have been tested for use with real-time Linux* kernels, and in virtualization manager (hypervisor) and virtual machine based environments.
- Cross-development targeting high priority applications running in their own protective virtual machine wrapper is supported.

Operating System and Hardware Platform Support

Support for Intel® Atom™ Processor and Intel® Core™ Processor Family

- All included software development tools components except for the Intel® Math Kernel Libraries (available for Intel® Core™ only) are optimized for both low power Intel® Atom™ Processor and Intel® Core™ Processor Family based designs.

Intel® AVX Optimizations

- Intel® AVX (Intel® Advanced Vector Extensions) and Intel® AVX2 are the next step in the evolution of Intel processor technology enabling higher data throughput via an updated vector instruction set.
- Increased data parallelism on Intel® Core™ processor through increased vector register width.
- Intel® C++ Compiler, Intel® MKL, and Intel® IPP adopt and support optimizations for Intel® AVX in many library functions and in auto-vectorization optimizations.

Support for Linux* OS Intel® 64 Host and a variety of Embedded IA32 and Intel® 64 Linux* OS target systems

- Tested on Fedora* and Ubuntu* host
- Tested for CE Linux* PR28, Wind River* Linux 5, Yocto Project* 1.x and Fedora* 14 targets including custom real-time kernels.

Tool Suite Component Features

3.1.1 Optimizing Compiler

- Parallel C/C++ language extension support
- Intel® Cilk™ Plus support
- Auto-vectorization taking advantage of Intel® AVX for Intel® Core™ i7 and Intel® SSSE3 for Intel® Atom™ Processor targeted applications
- Compatibility with existing GNU* Compiler generated code base
- Cross-build support for Yocto Project* 1.3, and CE Linux* PR28
- Support for –m32 and -m64 compiler options has been added. The generated code for these options is:

Host bin/ia32 –m64	- Target 64 bits
Host bin/intel64 –m32	- Target 32 bits

Please refer to
<install-dir>/system_studio_2013.1.xxx/documentation/compiler-release.pdf
for additional new features.

3.1.2 Performance and Parallelism Tuning

- Identify performance bottlenecks, suboptimal hardware resource utilization and inefficient core-utilization on mutli-core platforms.

Please refer to
<install-dir>/system_studio_2013.1.xxx/
vtune_amplifier_2013_for_systems/documentation/en
for additional new features.

3.1.3 Power and Core Frequency Tuning on Intel® Core™ Processor Family

- Identify the exact source code location that triggers unwanted wake-up events and frequency changes on Intel® Core™ Processor Family based platforms. Currently this is not supported for Intel® Atom™ Processor based platforms.

Please refer to
<install-dir>/ system_studio_2013.1.xxx/
vtune_amplifier_2013_for_systems/documentation/en
for additional new features.

3.1.4 Intel® Sampling Collector for Intel® VTune™ Amplifier 2013

- Event-based sampling support for latest generation Intel® Atom™ Processors
- Support for Linux* kernel versions 2.6.9 through 3.3.x.
- Support for Yocto Project* 1.3, as well as CE Linux* PR28 releases for Intel® Atom™ Processor CE4xxx and CE5xxx.
- Validation against the real-time scheduler Preempt-RT kernel for Yocto Project* 1.x on Intel® Atom™ Processor E6xx and Intel® Atom™ Processor N2xxx/D2xxx

Please refer to
<install-dir>/system_studio_2013.1.xxx/
vtune_amplifier_2013_for_systems/documentation/en
for additional new features.

3.1.5 Remote Intel® VTune Amplifier Collector for Embedded Systems

- Remote event-based sampling support including
  - Performance Tuning and Hotspot Analysis
  - Calltrace Dependency Analysis
  - Power and Frequency Analysis on Intel® Core™ Processor Family

Please refer to
<install-dir>/system_studio_2013.1.xxx/
vtune_amplifier_2013_for_systems/documentation/en/remote-target-collection.pdf
for additional new features.

3.1.6 Memory Checking and Threading Analysis

- Intel® Inspector 2013 supports Yocto Project* 1.2, 1.2.1, 1.3 and Wind River* Linux* 5 targets as well as CE Linux* PR28 on the Intel® Atom™ Processor CE42xx and CE53xx.
- Identify deadlocks and data races as well as memory leaks with Intel® Inspector
- Use command-line interface on target device or full Intel® Inspector GUI.

Please refer to
<install-dir>/ system_studio_2013.1.xxx/
inspector_2013_for_systems/documentation/en
for additional new features.

3.1.7 The GNU* Project Debugger GDB

- Data Race Detection (*pdbx*) for applications threaded using POSIX* Pthreads
- Last Branch Record Instruction Trace (*btrace*) on Intel® Atom™ processors
- Remote Debug & Virtualization Support
- GDB remote debug integration into Eclipse* CDT

Please refer to

<install-dir>/system_studio_2013.1.xxx/documentation/en_US/debugger/gdb/gdb.pdf

for details on these new features.

The *btrace* feature is documented in the chapter, “Branch Tracing”, and the GDB extended data race detection feature *pdbx* is documented in the chapter, “Debugging Data Races”.

The GDB Eclipse* integration as well as additional information on embedded use cases and virtualization can be found in the debug chapter of

<install-dir>/system_studio_2013.1.xxx/documentation/embedded-usage.pdf

3.1.8 Intel® JTAG Debugger for System Software Debug

- Instruction Trace Window:
  The new instruction trace window provides collapsible source assembly instruction display. Also new “Source follows Trace” and “Disassembly follows Trace” features have been added. This allows moving around in the trace output window and having the source and disassembly windows automatically jump to those locations as well.

- Extensible Firmware Interface (EFI) Support:
  When the target is booting through an EFI BIOS (PEI or DXE phase of the boot process) which has been built with debug information, the debugger can parse the PE/COFF headers and locate an address in source
  In addition the EFI plugin is able to automatically search and find the system table pointer. Once found many of the key aspects of the EFI environment including loaded services and modules can be listed.
- Debug bootloader, OS kernel and dynamically loaded device drivers via Intel® ITP-XDP3 or Macraigor usb2Demon
- Identify and fix runtime issues, memory configuration issues, and IO device issues
- Detailed page translation table and descriptor table views
- Instruction Tracing
- Bitfield Editor views
- Integrated Flashing Utility

Please refer to

<install-dir>/system_studio_2013.1.xxx/documentation/en_US/debugger/xdb
for additional new features.

3.1.9 Software Visible Event Nexus (SVEN)

- Trace Viewer:
  The SVEN Trace Viewer for Eclipse* IDE is a set of Eclipse* 3.7 (Indigo) plug-ins that allows SVEN trace data visualization investigations within the Eclipse IDE GUI environment. The SVEN Trace Viewer is implemented as a set of new trace display related views. These views are organized in an Eclipse* display perspective called SVEN.
- Software Development Kit (SDK):
  Event tracing infrastructure for low overhead data event tracing and logging in application as well as kernel space. You can use it to adapt the SVEN event tracing infrastructure to a Linux* platform. It also provides the necessary C/C++ include files and libraries for adding SVEN event instrumentation calls to kernel mode drivers and user space applications.
- Csven console:
  The csven utility is a console mode command for storing and displaying SVEN events directly on the platform. It can be uses both interactively and with scripts.

Please refer to
<install-dir>/system_studio_2013.1.xxx/documentation/en_US/debugger/sven
For additional new features

3.1.10 Intel® Math Kernel Library

- The Intel® Math Kernel Library (Intel® MKL) provides highly optimized functions for highly compute intensive signal processing and data processing applications on the Intel® Core™ Processor Family

Please refer to
<install-dir>/system_studio_2013.1.xxx/documentation/en_US/mkl
for additional new features.

3.1.11 Intel® Integrated Performance Primitives for Signal and Media Processing

- Intel ® Integrated Performance Primitives provide a rich set of function primitives and examples for media processing and signal processing targeting both Intel® Atom™ Processors and Intel® Core™ Processor Family based embedded platforms.
- Added new discrete Fourier transform functions: `DFTInit_R, DFTInit_C, DFTGetSize_R, DFTGetSize_C`).

- Added Polyphase Resampling functions.

Please refer to
<install-dir>/system_studio_2013.1.xxx/documentation/en_US/ipp
for additional new features.

4 System Requirements

4.1.1 Host Software Requirements

One of the following Linux distributions (this is the list of distributions supported by all components; other distributions may or may not work and are not recommended - please refer to Technical Support if you have questions).

- Red Hat Enterprise* Linux* 5, 6
- Ubuntu* 10.04 LTS, 11.04, 11.10, 12.04 LTS
- Fedora* 14, 15, 16, 17, 18
- openSUSE 12.1
- SUSE LINUX Enterprise Server* 10 SP4, 11 SP2

Host-Target Platform Focused Test Matrix

Host HW	Host OS	Target HW	Target OS
Intel ®64	Ubuntu* 11.04	Intel® Atom™ Processor CE53xx	CE Linux* PR28
Intel ®64	Fedora* 14	2nd gen. Intel® Core™ architecture based Intel® Xeon® Processor	Wind River Linux* 4, 5
Intel® 64	Fedora* 14	2nd gen. Intel® Core™ architecture based Intel® Xeon® Processor	Fedora* 14
IA32	Fedora* 17	Intel® Atom™ Processor E6xx	Yocto Project * 1.3

Individual Intel® System Studio 2013 for Linux* components may support additional distributions. See the individual component's installation guide and release notes after you unpacked and ran the installer for the tool suite distribution

```
> tar -zxvf l_cembd_p_2013.1.xxx.tgz
```

for details.

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 13.0:** Fedora* 17, Red Hat Enterprise Linux* 5, 6, SUSE LINUX Enterprise Server* 10, 11 SP2, Ubuntu* 11.10, 12.04, Debian* 6.0, Intel® Cluster Ready, Pardus* 2011.2 (x64 only)
- **Intel® VTune™ Amplifier 2013:** Red Hat* Enterprise Linux 4 (starting from Update 8) , Red Hat* Enterprise Linux 5 and 6, CentOS* versions equivalent to Red Hat*

Enterprise Linux* versions listed above, SUSE* Linux* Enterprise Server (SLES) 10 and 11, Fedora* 15, 16 and 17, Ubuntu* 10.04, 11.04, 11.10 and 12.04, Debian* 5.0 and 6.0

- **Intel® Inspector 2013:** Red Hat* Enterprise Linux* 5.8 and 6.3, CentOS* versions equivalent to Red Hat* Enterprise Linux* versions listed above,Fedora* 16 and 17, SUSE* Linux Enterprise Server* 10 SP4 and 11 SP2, Debian* 5 and 6, Ubuntu* 10.04, 11.10 and 12.04
- **Intel® JTAG Debugger:** Fedora* 14, 17; Ubuntu* 11.04

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>/bin/iccvars.sh <arg>
```

where `<install-dir>` is the directory structure containing the compiler `/bin` directory, and `<arg>` is one of the following architecture arguments:

`ia32`: Compilers and libraries for IA32 architectures only

`intel64`: Compilers and libraries for Intel® 64 architectures only

4.1.2 Target Software Requirements

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

- Yocto Project* 1.2, 1.3 based environment
- CE Linux* PR28 based environment
- Fedora* 14, 17 based environment
- Wind River* Linux* 4, 5 based environment

Note:

The level of target OS support by a specific Intel® System Studio component may vary.

4.1.3 Hardware Requirements

- IA32 or Intel® 64 architecture based host computer
- Development platform based on the Intel® Atom™ processor Z5xx, N4xx, N5xx, D5xx, E6xx, N2xxx, D2xxx or Intel® Atom™ processor CE4xxx, CE53xx and the Intel® Puma6™ Media Gateway
- Alternatively development platform based on 2nd or 3rd generation Intel® Core™ microarchitecture based Intel® Xeon® processor.

5 Installation Notes

Installing the Tool Suite

The default installation directories are:

- `/opt/intel/system_studio_2013.1.xxx/`
- `/opt/intel/system_studio_2013.1.xxx/ipp`
- `/opt/intel/system_studio_2013.1.xxx/mkl`
- `/opt/intel/system_studio_2013.1.xxx/debugger/xdb`
- `/opt/intel/system_studio_2013.1.xxx/debugger/gdb`
- `/opt/intel/system_studio_2013.1.xxx/debugger/sven`
- `/opt/intel/system_studio_2013.1.xxx/ vtune_amplifier_2013_for_systems/`
- `/opt/intel/system_studio_2013.1.xxx/inspector_2013_for_systems/`

for Intel® C++ Compiler, Intel® Integrated Performance Primitives, Intel® Math Kernel Library, Intel® JTAG Debugger, Intel® VTune™ Amplifier, and Intel® Inspector respectively.

The Intel® System Studio contains components under GNU* Public License (GPL) in addition to commercially licensed components. This includes the GNU* Project Debugger – GDB, the SVEN SDK and the kernel module used by the Intel® JTAG Debugger to export Linux* dynamically kernel module memory load information to host.

The Intel® VTune™ Amplifier and Intel® Inspector are available for power (Intel® Core™ Processor Family only) and performance tuning as well as memory and thread checking on the installation host. There is the capability to remotely collect performance and power data for embedded cross-development use cases as described in `remote-target-collection.pdf`.

For additional installation of command-line only versions of Intel® VTune™ Amplifier and Intel® Inspector on the development target, please follow the sub-chapter on the command line interface (CLI) installations below.

Furthermore a target package `system_studio_target.tgz` contains Intel® C++ Compiler runtime libraries, the Intel® VTune™ Amplifier Sampling Enabling Product (SEP) , target components for the Intel® VTune™ Amplifier Data Collector and the kernel module used by the Intel® JTAG Debugger to export Linux* dynamically kernel module memory load information to host.

The Intel® JTAG Debugger at this point is intended for Intel® Atom™ processor targeted system software cross-debug.

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.

Using the online installer

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

1. Ensure that you are connected to the internet and that https protocol based component downloads are permitted by your firewall.
2. Copy `online-installer.sh` into a directory of your choice.
3. Execute the install script in the directory where the tar file was extracted.
   ```
   >./online-installer.sh
   ```
4. After launching the online installer you will see the following output:
   ```
   Checking pre-requisites...
   Checking for 'wget'...
   Checking for 'awk'...
   Checking for 'tar'...
   Checking for 'cksum'...
   Extracting data...
   Starting installer...
   This is an online-installer for Intel(R) System Studio 2013
   Temporary files will be downloaded to: /tmp/root/.
   Started downloading of installer. Please, wait for several
   minutes... 100%
   Installer have been downloaded successfully.
   Extracting installer...
   Starting the PSET client...
   ```
5. The target package system_studio_target.tgz will be located in /tmp/<username> after the host installation is complete. The actual location can be verified by looking at the online installer output above.
6. Follow the same steps as outlined in “Installing the full distribution package” steps 5 through 18 below.

5.1.1 Online Installer Failure Reasons

5.1.1.1 Symptom 1: Online install bootstrapper cannot connect to IRC.

```
~$ ./online-installer.sh
Checking pre-requisites...
Checking for 'wget'...
Checking for 'awk'...
Checking for 'tar'...
Checking for 'cksum'...
Extracting data...
Starting installer...
This is an online-installer for Intel(R) System Studio 2013
Connection to the IRC site cannot be established.
~$
```

Root Cause:

- Port 443 not opened in firewall
- Port 443 not opened in ACL for specific server (can be verified using network trace).
- The Intel® System Studio online installer currently does not fully support proxy servers.

5.1.1.2 Symptom 2: Download stops in an infinite loop

Download stops in an infinite loop.

```
:~$ ./online-installer.sh
Checking pre-requisites...
Checking for 'wget'...
Checking for 'awk'...
Checking for 'tar'...
Checking for 'cksum'...
Extracting data...
Starting installer...
This is an online-installer for Intel(R) System Studio 2013
Temporary files will be downloaded to:
/tmp/<user>/l_cembd_p_2013.1.xxx.
```

Root Cause:

- Proxy server not specified on calling shell

5.1.1.3 Symptom 3: The online-installer hangs during step 6 of the installation

The online-installer hangs during step 6 of the installation

```
Step no: 6 of 7 | Installation
--------------------------------------------------------------------------------
Each component will be installed individually. If you cancel the installation,
components that have been completely installed will remain on your system. This
installation may take several minutes, depending on your system and the options
you selected.
--------------------------------------------------------------------------------
```

Root Cause:

- The Intel® System Studio online installer hangs because the user's proxy environment is not inherited after changing user to root account during installation. Please run online installer under root account, make sure that proxy settings were set before.

Installing the full distribution package

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_cembd_p_2013.1.xxx.tgz
   ```
2. Upon registering for the program you will receive a serial number and email with a license file. You will need either of these two to complete the installation process. If you want to use the license file you can point to it during install, but you can also copy it to `/opt/intel/licenses/`for automatic pickup by the installer.
3. Change into the directory the tar file was extracted to `../l_cembd_p_2013.1.xxx`
4. Execute the install script in the directory where the tar file was extracted.
   ```
   >./install.sh
   ```
5. 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. Depending on the installation location of the Yocto Project* Application Development Toolkit the integration of the Intel® C++ Compiler with it may require root privileges as well.

6. The welcome message to the Intel® System Studio for Linux* appears along with an outline of the installation process. Press the `Enter` key to continue.
7. The installation routine checks for the availability of all product dependencies. Please address these dependencies, if a warning message appears.
8. 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.
9. 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.
10. The next screen will ask you whether you would like to participate in the Intel® software Improvement Program. This will help us identify opportunities for product improvement. If you do not want to participate simply select option [2].
11. The following screen lets you review your installation options. If you would like to only install a specific tool suite component, select [3] and change the components settings. Else, continue with the default choice [1] to start the installation.

```
Step no: 5 of 7 | 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. Next you will be informed that open source components under General GNU Public License and Eclipse Public License are about to be installed. The installation can be stopped at this point by selecting [q]
13. If you do not intend to develop applications targeting Yocto Project* builds, you can ignore this warning message.

If you do a full Intel® System Studio 2013 for Linux* installation you will see installation messages for the following components:

```
-----------------------------------------------------------------------------
Installing Amplifier Command line interface component... done
-----------------------------------------------------------------------------
Installing Amplifier Sampling driver kit component... done
-----------------------------------------------------------------------------
Installing Amplifier Power driver kit component... done
-----------------------------------------------------------------------------
Installing Amplifier Graphical user interface component... done
-----------------------------------------------------------------------------
Installing Inspector Command line interface component... done
-----------------------------------------------------------------------------
Installing Inspector Graphical user interface component... done
-----------------------------------------------------------------------------
Installing Intel(R) C++ Compiler for applications running on Embedded OS
Linux* component... done
-----------------------------------------------------------------------------
Installing Intel Math Kernel Library 11.0 Update 1 on Intel(R) 64
component... done
-----------------------------------------------------------------------------
Installing Intel Integrated Performance Primitives 7.1 Update 1 on IA32
component... done
-----------------------------------------------------------------------------
Installing Intel Integrated Performance Primitives 7.1 Update 1 on Intel(R)
64
component... done
-----------------------------------------------------------------------------
Installing Intel(R) JTAG Debugger 3.0 for Linux* component... done
-----------------------------------------------------------------------------
Installing SVEN (SDK provided under GNU General Public License v3)... done
-----------------------------------------------------------------------------
Installing GNU GDB 7.5 (Provided under GNU General Public License v2)... done
-----------------------------------------------------------------------------
```

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

```
Step no: 7 of 7 | Complete

-----------------------------------------------------------------------------
Thank you for installing and using the
Intel(R) System Studio 2013

Reminder: Intel(R) VTune(TM) Amplifier users must be members of the "vtune"
permissions group in order to use Event-based Sampling.

To register your product purchase, visit
https://registrationcenter.intel.com/RegCenter/registerexpress.aspx?media=MWX

To get started using Intel(R) VTune(TM) Amplifier 2013 for Systems:
- To set your environment variables: source
```

```
/opt/intel/system_studio_2013.1.xxx/vtune_amplifier_2013_for_systems/amplxe-
vars
.sh
    - To start the graphical user interface: amplxe-gui
    - To use the command-line interface: amplxe-cl
    - For more getting started resources:
/opt/intel/system_studio_2013.1.xxx/vtune_amplifier_2013_for_systems/
      documentation/en/welcomepage/get_started.html
    - Target system data collectors are located in folder:

/opt/intel/system_studio_2013.1.xxx/vtune_amplifier_2013_for_systems/target/
To get started using Intel(R) Inspector 2013 for Systems:
- To set your environment variables: source
/opt/intel/system_studio_2013.1.xxx/inspector_2013_for_systems/inspxe-vars.sh
    - To start the graphical user interface: inspxe-gui
    - To use the command-line interface: inspxe-cl
    - For more getting started resources:
/opt/intel/system_studio_2013.1.xxx/inspector_2013_for_systems/
      documentation/en/welcomepage/get_started.html.
To get started using Intel(R) C++ Compiler 2013 for Embedded OS Linux*:
    - Set the environment variables for a terminal window using one of the
      following (replace "intel64" with "ia32" if you are using a 32-bit
      platform).
      For csh/tcsh:
           $ source /opt/intel/system_studio_2013.1.xxx/bin/iccvars.csh
intel64
      For bash:
           $ source /opt/intel/system_studio_2013.1.xxx/bin/iccvars.sh
intel64
      To invoke the installed compilers:
           For C++: icpc
           For C: icc
      To get help, append the -help option or precede with the man command.
    - For Eclipse integration of Intel(R) C++ Compiler please run the
following
script:

/opt/intel/system_studio_2013.1.xxx/eclipse_support/eclipse_integration.sh

For more getting started resources:
      /opt/intel/system_studio_2013.1.xxx/documentation/embedded_gsdoc_l.htm

To view movies and additional training, visit
http://www.intel.com/software/products.

--------------------------------------------------------------------------------
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>/bin/iccvars.sh <arg>
```

where `<install-dir>` is the directory structure containing the compiler `/bin` directory, and `<arg>` is one of the following architecture arguments:

`ia32`: Compilers and libraries for IA32 architectures only

`intel64`: Compilers and libraries for Intel® 64 architectures only

Notes

5.1.2 Silent Install

For information on automated or "silent" install capability, please see http://intel.ly/ngVHY8.

5.1.3 Selecting the provided GDB

To use the provided GNU* Project Debugger GDB instead of the default GDB debugger provided with the default GNU* tools installation of your distribution, please source the following debugger environment setup script:

```
<install-dir>/system_studio_2013.1.xxx/debugger/gdb/bin/
debuggervars.sh
```

Development target package installation

The target package system_studio_target.tgz contains Intel® C++ Compiler runtime libraries, the Intel® VTune™ Amplifier Sampling Enabling Product (SEP) , target components for the Intel® VTune™ Amplifier Data Collector, target components for the Intel® Inspector, and the idbntf.ko used by the Intel® JTAG Debugger to export Linux* dynamically kernel module memory load information to host.

To install it follow the steps below

1. Unpack the tool suite package in a directory to which you have write access.
   `> tar -zxvf l_cembd_p_2013.1.xxx.tgz`
2. From the thus created directory `../l_cembd_b_2013.1.xxx/rpm/`, copy the file `system_studio_target.tgz` to the target device using ftp, sftp or scp.
3. On the target device unpack the file in a directory to which you have write access
   `> tar -zxvf system_studio_target.tgz`
4. Add the compiler runtime libraries that you find in `../system_studio_target/compiler/lib` to your target environment search path.
5. For the dynamic kernel module load export feature follow the instructions found at `../system_studio_target/xdb/kernel-modules/idbntf/read.me`. This is also detailed in the Intel® JTAG Debugger Installation Guide and Release Notes `jtag-release-install.pdf.`

6. For the Intel® VTune Amplifier Sampling Enabling Product (SEP) follow the instructions in `../system_studio_target/sep_3.10_linux_ia32/docs/SEP_Install_Instructions_Linux.txt.`

5.1.4 Intel® Inspector Command line interface installation

If you would like to install the Intel® Inspector command line interface only for thread checking and memory checking on a development target device, please follow the steps outlined below:

1. Unpack the tool suite package in a directory to which you have write access.
   ```
   > tar -zxvf l_cembd_p_2013.1.xxx.tgz
   ```
2. From the thus created directory `../l_cembd_b_2013.1.xxx/rpm/`, copy the file `system_studio_target.tgz`to the target device using ftp, sftp or scp.
3. On the target device unpack the file in a directory to which you have write access
   ```
   > tar -zxvf system_studio_target.tgz
   ```
4. In the thus created directory `../system_studio_target/`, you will find the Intel® Inspector installation at `../system_studio_target/inspector_2013_for_systems/`
5. Execute the environment configuration script `inspxe-genvars.sh.`
6. The fully functional command-line Intel® Inspector installation can be found in the `bin32` and `bin64` subdirectories for IA32 and Intel® 64 targets respectively.

5.1.5 Intel® VTune™ Amplifier Command line data collector installation (CLI)

If you would like to install the Intel® VTune™ Amplifier `amplxe-runss` command line data collector for power tuning and performance tuning on a development target device, please follow the steps outlined below:

7. Unpack the tool suite package in a directory to which you have write access.
   ```
   > tar -zxvf l_cembd_p_2013.1.xxx.tgz
   ```
8. From the thus created directory `../l_cembd_p_2013.1.xxx/rpm/`, copy the file `system_studio_target.tgz` to the target device using ftp, sftp or scp.
9. On the target device unpack the file in a directory to which you have write access
   ```
   > tar -zxvf system_studio_target.tgz
   ```
10. In the thus created directory `..system_studio_target/`, you will find the Intel® Amplifier target data collector installation at `../system_studio_target/vtune_amplifier_2013_for_systems/target`
11. Data collection on both IA32 and Intel® 64 targets is supported.
12. Follow the instructions in the file `remote-data-collecton.pdf` and `release_notes_amplifier_linux.pdf` for more details, on how to use this utility.

5.1.6 Intel® VTune™ Amplifier Sampling Enabling Product

If you would like to install the Intel® VTune™ Amplifier Sampling Enabling Product (SEP), please follow the steps outlined below:

1. Unpack the tool suite package in a directory to which you have write access.
   ```
   > tar -zxvf l_cembd_p_2013.1.xxx.tgz
   ```
2. From the thus created directory `../l_cembd_b_2013.1.xxx/rpm/`, copy the file `system_studio_target.tgz`to the target device using ftp, sftp or scp.
3. On the target device unpack the file in a directory to which you have write access
   ```
   > tar -zxvf system_studio_target.tgz
   ```
4. In the thus created directory `../system_studio_target/`, you will find the Intel® VTune Amplifier Sampling Enabling Product at
   `../system_studio_target/sep_3.10_axeu3_linux_x32` or
   `../system_studio_target/sep_3.10_axeu3_linux_em64t`
5. Follow the instructions in
   `../system_studio_target/sep_3.10_linux_<target>/docs/SEP_Install_Instructions_Linux.txt`
   and
   `../system_studio_target/sep_3.10_linux_<target>/docs/README`
6. See `SEP_User_Guide.pdf` in the target package for more details on usage.

5.1.7 Intel® Integrated Performance Primitives redistributable shared object installation

If you are using dynamic linking when using the Intel® Integrated Performance Primitives, you will need to copy the relevant Linux* shared objects to the target device along with the application.
A list of the redistributable shared objects can be found at
`/opt/intel/system_studio_2013.1.xxx/documentation/ippredist.txt`

5.1.8 Intel® Math Kernel Library redistributable shared object installation

If you are using dynamic linking when using the Intel® Math Kernel Libraries, you will need to copy the relevant Linux* shared objects to the target device along with the application.
A list of the redistributable shared objects can be found at
`/opt/intel/system_studio_2013.1.xxx/documentation/mklredist.txt`

Eclipse* IDE Integration

5.1.9 Installation

The Intel® C++ Compiler can be automatically integrated into a preexisting Eclipse* CDT installation. The Eclipse* CDK, Eclipse* JRE and the Eclipse* CDT integrated development environment are not shipped with this package of the Intel® System Studio. The Intel® System Studio contains an interactive script that asks for the Eclipse component location and helps to integrate already installed Intel® System Studio components into the Eclipse* IDE. Currently only integration into the 64bit (x86_64) build of Eclipse* is supported.

To have the Eclipse Integration done as part of the Intel® System Studio installation navigate to

```
/opt/intel/system_studio_2013.1.xxx/eclipse_support/
```

and execute the automated Eclipse integration script.

```
> ./eclipse_integration.sh
```

If asked point the integration script to the installation directory of your Eclipse* 3.7 install. Usually this would be `/opt/eclipse/`.

The prerequisites for successful Eclipse integration are:

1. Eclipse* 3.7 (Indigo)
2. Eclipse* CDT 8.0
3. Java Runtime Environment (JRE) version 6.0 (also called 1.6) update 11 or later.

The Eclipse* 3.7.x environment that the integration has been validated against and that is strongly recommended to be used can be downloaded from the eclipse.org website at

http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/indigo/SR2/eclipse-cpp-indigo-SR2-incubation-linux-gtk-x86_64.tar.gz (64bit host)

5.1.10 Launching Eclipse for Development with the Intel C++ Compiler

Since Eclipse requires a JRE to execute, you must ensure that an appropriate JRE is available to Eclipse prior to its invocation. You can set the `PATH` environment variable to the full path of the folder of the `java` file from the JRE installed on your system or reference the full path of the java executable from the JRE installed on your system in the `-vm` parameter of the Eclipse command, e.g.:

```
eclipse -vm /JRE folder/bin/java
```

Invoke the Eclipse executable directly from the directory where it has been installed. For example:

```
<eclipse-install-dir>/eclipse/eclipse
```

5.1.11 Installing on Fedora* Systems

If the Intel C++ Compiler for Linux is installed on an Intel® 64 architecture Fedora* system as a "local" installation, i.e. not installed as root, the installation may fail to properly execute the Eclipse graphical user interfaces of the compiler or debugger. The failure mechanism will typically be displayed as a `JVM Terminated` error. The error condition can also occur if the software is installed from the root account at the system level, but executed by less privileged user accounts.

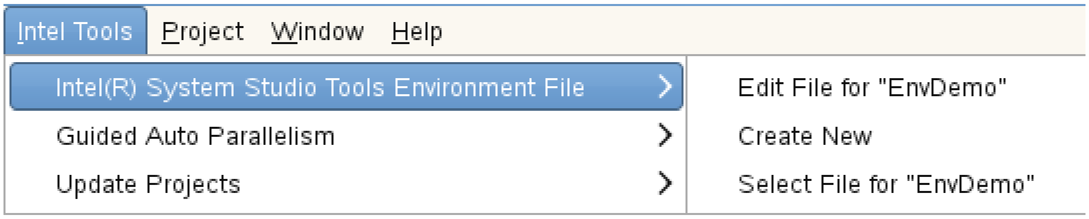
The cause for this failure is that a more granular level of security has been implemented on Fedora, but this new security capability can adversely affect access to system resources, such as dynamic libraries. This new *SELinux* security capability may require adjustment by your system administrator in order for the compiler installation to work for regular users.

5.1.12 Editing Compiler Cross-Build Environment Files

Environment File Support appears under “Intel Tools - Intel® System Studio Tools Environment File” on the menu bar.

If a project has been selected in the Project Explorer, a user will see “Edit File for <project name>, “Create New” and “Select File for <project name>.

If a project has not been selected, a user will see “Edit File” and “Create New”.



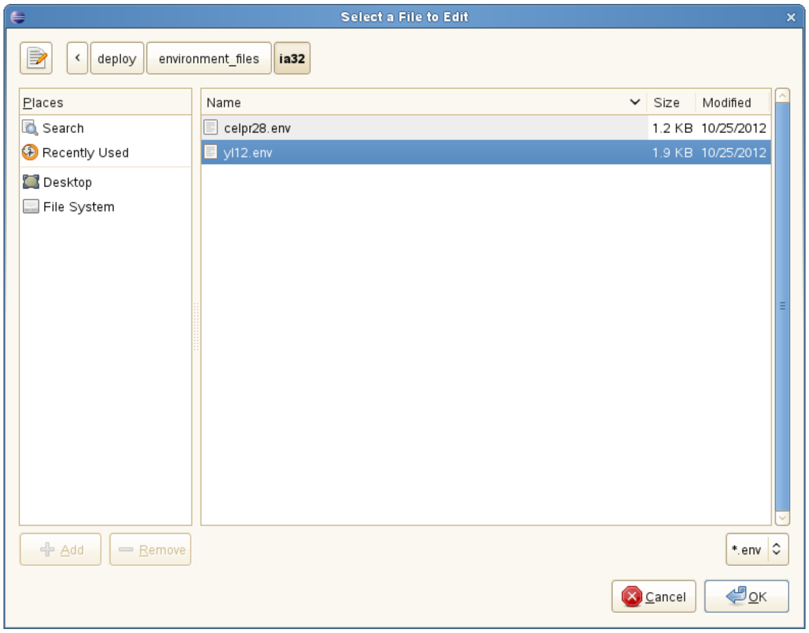
Intel Tools->Intel® System Studio Tools Environment File ->Edit File for <project name>

This option lets a user edit the environment file associated with the current project. If no project is set, an error message box will appear asking them to select a file first.

Intel Tools->Intel® System Studio Tools Environment File ->Edit File

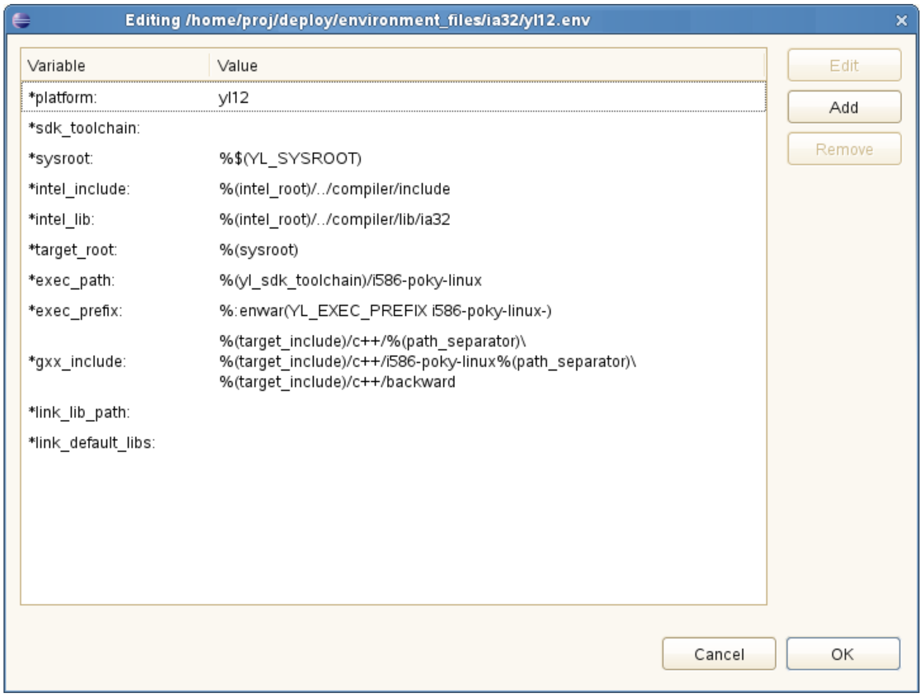
on the other hand, lets a user edit any existing environment file.

1. In this case, “Select a File to Edit” dialog appears.



Name	Size	Modified
celpr28.env	1.2 KB	10/25/2012
yl12.env	1.9 KB	10/25/2012

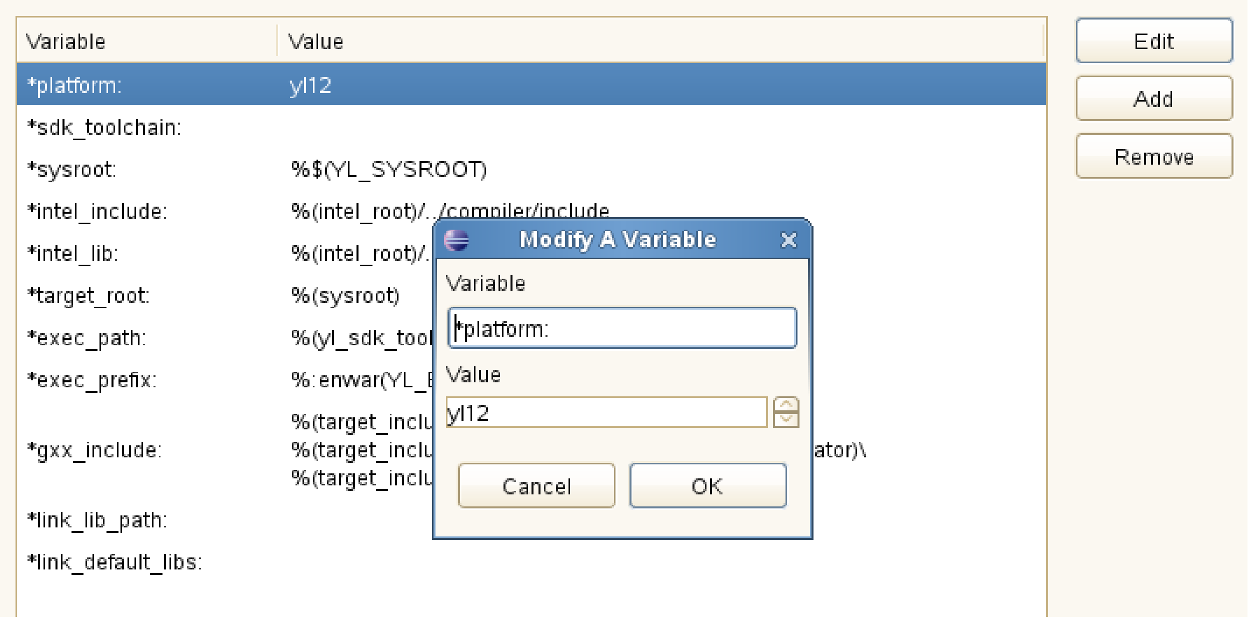
2. Then a user will see an edit dialog such as this.



Editing /home/proj/deploy/environment_files/ia32/yl12.env

Variable	Value
*platform:	yl12
*sdk_toolchain:	
*sysroot:	%$(YL_SYSROOT)
*intel_include:	%(intel_root)/../compiler/include
*intel_lib:	%(intel_root)/../compiler/lib/ia32
*target_root:	%(sysroot)
*exec_path:	%(yl_sdk_toolchain)/i586-poky-linux
*exec_prefix:	%:enwar(YL_EXEC_PREFIX i586-poky-linux-)
*gxx_include:	%(target_include)/c++/%(path_separator)\ %(target_include)/c++/i586-poky-linux%(path_separator)\ %(target_include)/c++/backward
*link_lib_path:	
*link_default_libs:	

3. To edit a variable value, click on a variable and then “Edit”. “Modify a Variable” box will appear.



Variable	Value
*platform:	yl12
*sdk_toolchain:	
*sysroot:	%$(YL_SYSROOT)
*intel_include:	%(intel_root)/../compiler/include
*intel_lib:	%(intel_root)/.
*target_root:	%(sysroot)
*exec_path:	%(yl_sdk_tool
*exec_prefix:	%:enwar(YL_E
*gxx_include:	%(target_inclu %(target_inclu ator)\ %(target_inclu
*link_lib_path:	
*link_default_libs:	

Modify A Variable

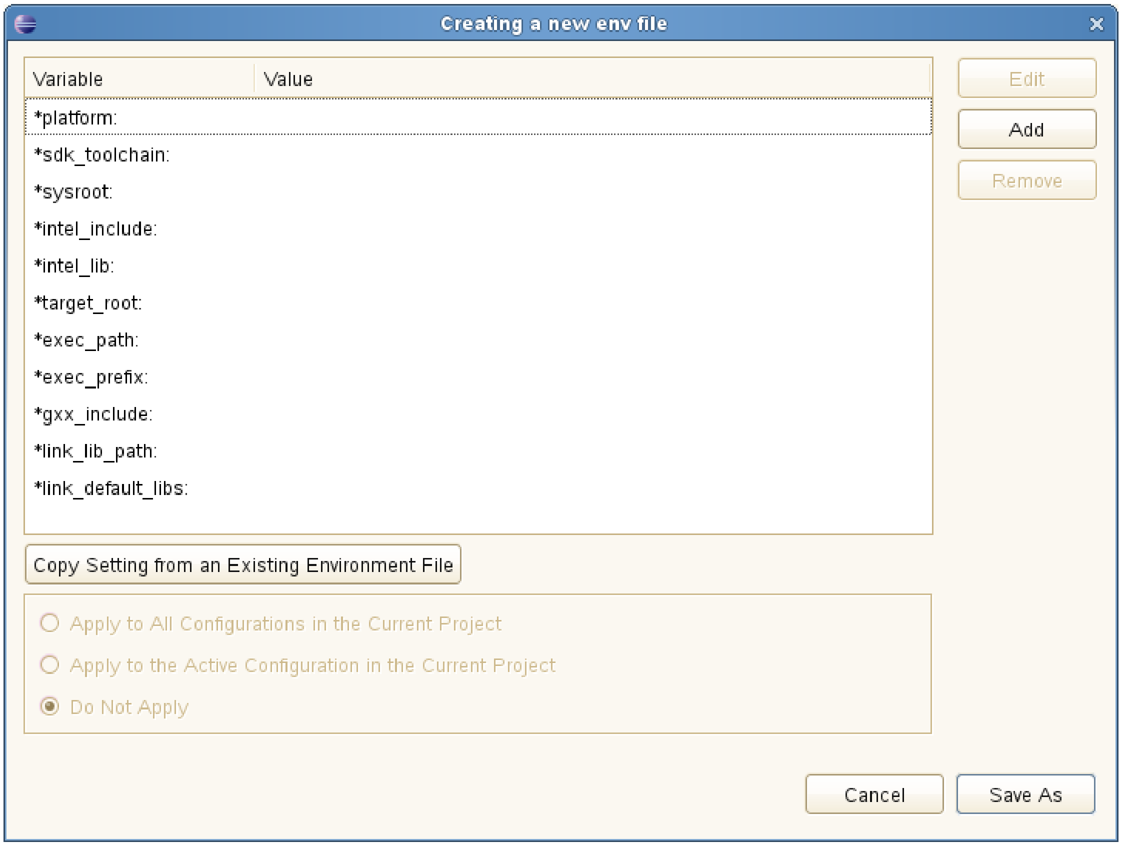
Variable: *platform:

Value: yl12

4. Select “OK” to save the change and get out of the editing box.
5. Use “Add” to add a new variable-value pair. “Add A New Variable” will appear.
6. To remove a pair, select a pair and select “Remove”. A confirmation dialog will appear. Press OK.
7. When finished with all editing (/ adding / removing), select “OK” in the “Editing..” window.

Intel Tools->Intel® System Studio Tools Environment File ->Create New

This option lets a user create a new .env file. A “Creating a new env file” appears.



Variable	Value
*platform:	
*sdk_toolchain:	
*sysroot:	
*intel_include:	
*intel_lib:	
*target_root:	
*exec_path:	
*exec_prefix:	
*gxx_include:	
*link_lib_path:	
*link_default_libs:	

○ Apply to All Configurations in the Current Project
○ Apply to the Active Configuration in the Current Project
● Do Not Apply

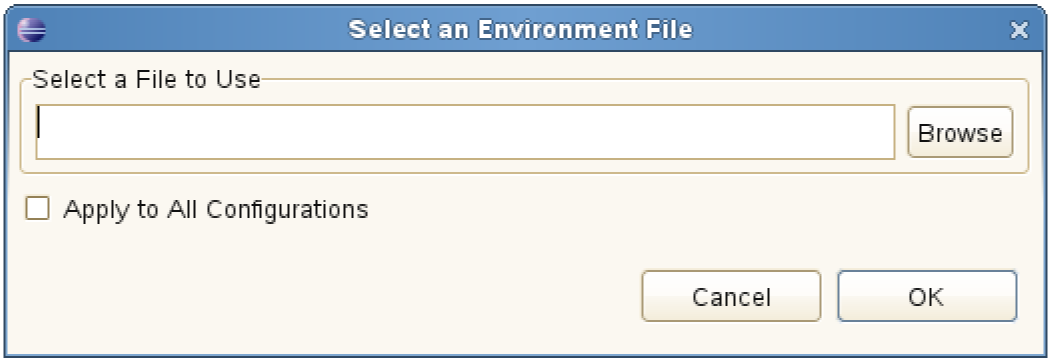
If you'd like to use an existing env file as a base, select "Copy Setting from an Existing Environment File". Then select the base file in the "Select a File to Copy the Setting from" dialog . The values specified in that file now should be propagated to the value fields.

Edit your file.

Select "Apply to All.." – "Do Not Apply" to control where this environment file should be used. If no project has been selected in Project Explorer, these options are grayed out.

Select "Save As" to select the file name and save.

1. Intel Tools->Intel® System Studio Tools Environment File ->Select File for <project name>



This option lets a user select a file that's used with the active configuration the current project. Select "Apply to All Configurations", if the file should be used in all configurations in the current project.

5.1.13 Cheat Sheets

The Intel® C++ Compiler Eclipse* Integration additionally provides Eclipse* style cheat sheets on how to set up a project for embedded use cases using the Intel® C++ Compiler
In the Eclipse* IDE see

Help > Cheat Sheets > Intel C++ Integration

5.1.14 SVEN Trace Viewer integration into Eclipse*

Please refer to the SVEN Trace Viewer Installation chapter in

<install-dir>/system_studio_2013.1.xxx/documentation/en_US/debugger/sven/viewer/
SVEN_VIEWER.pdf

5.1.15 Integrating the provided GDB into Eclipse* for remote debug

Remote debugging with GDB using the Eclipse* IDE requires installation of the C/C++ Development Toolkit (CDT) (http://www.eclipse.org/downloads/packages/eclipse-ide-cc-linux-developers-includes-incubating-components/indigosr2) as well as Remote System Explorer (RSE) plugins (http://download.eclipse.org/tm/downloads/). In addition RSE has to be configured from within Eclipse* to establish connection with the target hardware.

1. Copy the `gdbserver` provided by the product installation
   `(<install-dir>/system_studio_2013.1.xxx/debugger/gdb/<arch>/<python>/bin/)`
   to the target system and add it to the execution PATH environment variable on the target.

2. Configure Eclipse* to point to the correct GDB installation:

a. Inside the Eclipse* IDE click on Window>Preferences from the pulldown menu.

b. Once the preferences dialogue appears select C++>Debug>GDB from the treeview on the left.

c. The GDB executable can be chosen by editing the “GDB debugger” text box. Point to `<install-dir>/system_studio_2013.1.xxx/debugger/gdb/<arch>/<python>/bin/`,

   where `<arch>` is ia32 or intel64 and `<python>` is py24, py26, or py27, depending on architecture and Python* installation

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.x86_64.rpm or mcgr-hwsupport-x.x-x.i386.rpm from http://www.macraigor.com/full_gnu.htm. For Ubuntu* host installations you can download mcgr-hwsupport-x.x-x.x86_64.deb or 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.

The Intel® JTAG Debugger has been validated for use with the Macraigor Systems* usb2Demon* device and OCDRemote* 9.90. We recommend using at least version 9.90 of the Macraigor* Systems* driver for most platforms except for the Intel® Atom™ Processor CE53xx, where we currently recommend v9.72.

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

2. Install mcgr-hwsupport-x.x-x.x86_64.rpm or mcgr-hwsupport-x.x-x.i386.rpm using the `rpm -i` command,

   or install mcgr-hwsupport-x.x-x.x86_64.deb 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 start_xdb_MCRG_processorname.sh startup script. These scripts are being put into the /opt/intel/system_studio_2013.1.xxx 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.

Removing the Product

To uninstall the Intel® System Studio, change to the `/opt/intel/system_studio_2013.1.xxx` directory and run the `uninstall.sh` uninstall script.

You will be given the following choices:

```
0. All components
1. Intel(R) VTune(TM) Amplifier 2013 for Systems
2. Intel(R) Inspector 2013 for Systems
3. Intel(R) C++ Compiler 13.0 for Linux*
4. Intel(R) JTAG Debugger 3.0 for Linux*
5. SVEN (SDK provided under GNU General Public License v2)
6. GNU* GDB 7.5 (Provided under GNU General Public License
v3)

x. Exit [default]
```

6 Tool Suite Usage

The Intel® System Studio supports a wide variety of Intel® Atom™ processor and Intel® Core™ processor family 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 embedded devices and intelligent systems 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® System Studio addresses this in multiple ways.

- The Intel® System Studio package will by default be installed on the development host system.
- If your development needs include the use of chroot to provide a protected build environment without host system library pollution, the Intel® Composer (the Intel® C++ Compiler, Intel® Math Kernel Libraries, Intel® Integrated Performance Primitives) and other components may in addition also be installed into a jailroot or chroot type environment by executing the master install.sh installation script inside aforementioned chroot environment.
- The Intel® Inspector command-line version can be installed on the embedded target device for thread checking and memory checking there.
- The Intel® VTune™ Amplifier provides options ranging from using amplxe-runss for Python* and ssh assisted remote data collection and local embedded device collection using a sampling driver for event based sampling only to installing a full command-line version of the Intel® VTune™ Amplifier on the target device. This provides a range of options and features for your footprint and resource constraints to best fit the needs for efficient power and performance data collection.
- 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.

Compiler Usage for Cross-Compilation

6.1.1 Build Applications for Yocto Project* Target

The Intel® System Studio installer will automatically integrate the Intel® C++ Compiler with an existing Application Development Toolkit (ADT) installation for Yocto Project* 1.2.x and 1.3

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://downloads.yoctoproject.org/releases/yocto/yocto-1.2/adt_installer, http://downloads.yoctoproject.org/releases/yocto/yocto-1.2.1/adt_installer or http://downloads.yoctoproject.org/releases/yocto/yocto-1.3/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

The Yocto Project* distributions and board support packages can be found at http://www.yoctoproject.org/download

Yocto Project* documentation can be found at http://www.yoctoproject.org/documentation

6.1.1.1 Environment Variables

The required environment variables that need to be set using the `source` command are:

```
YL_SYSROOT=/opt/poky/1.2/sysroots/i586-poky-linux
YL_TOOLCHAIN=/opt/poky/1.2/sysroots/i686-pokysdk-linux/usr/bin
```

or

```
YL_SYSROOT=/opt/poky/1.2.1/sysroots/i586-poky-linux
YL_TOOLCHAIN=/opt/poky/1.2.1/sysroots/i686-pokysdk-linux/usr/bin
```

or

```
YL_SYSROOT=/opt/poky/1.3/sysroots/i586-poky-linux
YL_TOOLCHAIN=/opt/poky/1.3/sysroots/i686-pokysdk-linux/usr/bin
```

6.1.1.2 Compiler Platform Option for Cross-Compilation

The compiler option specifying the cross-compilation is

```
icc  -platform=yl12 my_source_file.c
icpc -platform=yl12 my_source_file.c
```

or

```
icc  -platform=yl13 my_source_file.c
icpc -platform=yl13 my_source_file.c
```

The corresponding environment file is located in `<install-dir>/bin/ia32`.

6.1.2 Build Applications for Wind River* Linux* OS Target

To find out more about Wind River* Linux* 5 and on how to install the required Wind River* Linux* toolchain go to http://www.windriver.com/products/linux/

It is based on Yocto Project* and follows many of the same development paradigms http://www.windriver.com/announces/yocto-project/

6.1.2.1 Environment Variables

The required environment variables that need to be set using the `source` command are:

```
WRL_TOOLCHAIN=<install_dir>/wrl50/wrlinux-5/layers/wr-toolchain/4.6-60

WRL_SYSROOT=<install_dir>/wrl50/<target>/export/sysroot/intel-xeon-core_glibc_std/bitbake_build/tmp/sysroots/intel-xeon-core
```

where <install_dir> is the installation path for the Wind River* Linux* software development toolkit, e.g. `/export/SDK` and target is `<ia32>` or `<intel64>`.

6.1.2.2 Compiler Platform Option for Cross-Compilation

The compiler option specifying the cross-compilation is

```
icc  -platform=wrl50 my_source_file.c
icpc -platform=wrl50 my_source_file.c
```

The corresponding environment file is located in `<install-dir>/bin/<target>`

where `<install-dir>` is `/opt/intel/system_studio_2013.1.xxx` and target is `<ia32>` or `<intel64>`.

6.1.3 Build Applications for CE Linux* Target

To get access to the sources and the build environment for your Intel® Atom™ Processor CE42xx or CE53xx based device, please contact the manufacturer of your media streaming device, blu-ray player or IPTV.
If you are a device manufacturer, please contact your Intel Application Engineer to get access.

Below we describe how the Intel® C++ Compiler can interact with a CE Linux* build environment that you installed on your host development machine

6.1.3.1 Environment Variables

The required environment variables that need to be set using the `source` command are:

```
CEL_TOOLCHAIN=<install_dir>/celpr28/staging_dir
CEL_SYSROOT=<install_dir>/celpr28/i686-linux-elf
```

where `<install_dir>` is the installation path for the CE Linux* PR28 product release support package

6.1.3.2 Compiler Platform Option for Cross-Compilation

The compiler option specifying the cross-compilation is

```
icc  -platform=celpr28  my_source_file.c
icpc -platform=celpr28  my_source_file.c
```

The corresponding environment file is located in `<install-dir>/bin/ia32`

where `<install-dir>` is `/opt/intel/system_studio_2013.1.xxx`

6.1.4 Compiler run-time libraries

If you are linking your application built with the Intel® C++ Compiler dynamically, you will need to make the Intel® C++ Compiler runtime libraries available on the target platform. You can do this by following the steps below:

1. Unpack the tool suite package in a directory to which you have write access.
   `> tar -zxvf l_cembd_p_2013.1.xxx.tgz`
2. From the thus created directory `../l_cembd_b_2013.1.xxx/rpm/,` copy the file `system_studio_target.tgz`to the target device using ftp, sftp or scp.
3. On the target device unpack the file in a directory to which you have write access
   `> tar -zxvf system_studio_target.tgz`
4. In the thus created directory `../system_studio_target/` the libraries can be found at `../system_studio_target/compiler/lib`.

Selecting the provided GDB

To use the provided GNU* Project Debugger GDB instead of the default GDB debugger provided with the default GNU* tools installation of your distribution, please source the following debugger environment setup script:

```
<install-dir>/system_studio_2013.1.xxx/debugger/gdb/bin/
debuggervars.sh
```

7 Issues and Limitations

Known Issues and Limitations

For known issues of individual Intel® System Studio 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

7.1.1 Offline installation with non-root access

When installing as a current non-privileged user, activation option “Use a different computer with internet access” does not work. To activate the product without internet connection enabled, use a license file or use root or sudo privileges instead

7.1.2 Running online-installer behind proxy server fails

Running online-installer behind proxy server produces error: "Connection to the IRC site cannot be established". Please see the Installation Notes for more details

7.1.3 The online-installer has to be run with sudo or root access

If the online-installer is run as a regular user the installation will hang in step 6 of the installation. Please see the Installation Notes for more details.

```
Step no: 6 of 7 | Installation
--------------------------------------------------------------------------------
Each component will be installed individually. If you cancel the installation,
components that have been completely installed will remain on your system. This
installation may take several minutes, depending on your system and the options
you selected.
--------------------------------------------------------------------------------
```

7.1.4 Yocto Project* Application Development Toolkit 1.2 (ADT) has only limited support for static linking

When using the Application Development Toolkit 1.2 and specifying the build option – static in a makefile to enforce a static linking model for libraries, the build may fail with unresolved library dependencies.

7.1.5 Compiler integration into Eclipse* requires 64bit build of Eclipse* IDE

The Intel® System Studio contains an interactive script that asks for the Eclipse component location and helps to integrate already installed Intel® System Studio components into the Eclipse* IDE. Currently only integration into the 64bit (x86_64) build of Eclipse* is supported.

The Eclipse* 3.7.x environment that the integration has been validated against and that is strongly recommended to be used can be downloaded from the eclipse.org website at

http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/indigo/SR2/eclipse-cpp-indigo-SR2-incubation-linux-gtk-x86_64.tar.gz (64bit host)

Intel® VTune™ Amplifier Power and Frequency Analysis

7.1.6 /boot/config-'uname –r' file must be present on platform

In order to enable CPU power data collection please make sure your environment does have a file named `/boot/config-`uname -r`` located in your `/boot/config` directory

If there is no such file you should run the following command:

```
$ cat /proc/config.gz | gunzip - > /boot/config-`uname -r`
```

7.1.7 Power and Frequency Analysis only supported on Intel® Core™ Processor Family

Power and frequency analysis currently requires at least a 2nd generation Intel® Core™ Processor Family based platform.

Sampling Collector (SEP) for Intel® VTune™ Amplifier Usage with Yocto Project*

7.1.8 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* 1.2 kernel build tree: viz, recordmcount, fixdep, and modpost.

Software Visible Event Nexus (SVEN)

7.1.9 Eclipse* IDE Trace Viewer add-on is not signed

The SVEN Trace Viewer add-on for Eclipse* CDT 3.7 (Indigo) is not digitally signed in this first release. You will receive the following warning upon installation:

"Warning: You are installing software that contains unsigned content. The authenticity or validity of this software cannot be established. Do you want to continue with the installation?"

Please approve this message to complete the integration of the SVEN Trace Viewer into Eclipse*.

7.1.10 SVEN SDK is not targeting Intel® Atom™ Processor CE series

The SVEN SDK included in this distribution is not intended for use with the Intel® Atom™ Processor CE41xx, CE42xx, CE53xx, or the Intel® Puma6™ Media Gateway.

To take advantage of all the capabilities provided by SVEN on those platforms, please contact your Intel Field representative.

7.1.11 Data Race Detection (*pdbx*) requires Python* 2.6 or Python* 2.7

The data race detection feature (pdbx) is designed to work on systems with Python* 2.6 or 2.7 installed. Some error messages might appear when running on system with Python* 2.4.

Python* 2.7 can be downloaded from http://www.python.org/download/releases/2.7.3/.

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

The Intel® C++ Compiler, Intel® Integrated Performance Primitives, Intel® Math Kernel Library, Intel® VTune™ Amplifier, and Intel® Inspector for Systems are provided under Intel's End User License Agreement (EULA).

The GNU* Project Debugger, GDB is provided under the General GNU Public License GPL V3.

The SVEN SDK is provided under the General GNU Public License GPL V2.

The SVEN Trace Viewer is provided under the Eclipse Public License EPL V1.

The Intel® JTAG Debugger kernel module idbntf.ko is provided under the General GNU Public License GPL.

Please consult the licenses included in the distribution for details.

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.