INTEL® PERCEPTUAL COMPUTING SDK

Getting Started Guide

LEGAL DISCLAIMER

THIS DOCUMENT CONTAINS INFORMATION ON PRODUCTS IN THE DESIGN PHASE OF DEVELOPMENT.

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 BY VISITING INTEL'S WEB SITE HTTP://WWW.INTEL.COM.

ANY SOFTWARE SOURCE CODE REPRINTED IN THIS DOCUMENT IS FURNISHED UNDER A SOFTWARE LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF THAT LICENSE ANY SOFTWARE SOURCE CODE REPRINTED IN THIS DOCUMENT IS FURNISHED UNDER A SOFTWARE LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF THAT LICENSE

INTEL, THE INTEL LOGO, INTEL CORE, INTEL MEDIA SOFTWARE DEVELOPMENT KIT (INTEL MEDIA SDK) ARE TRADEMARKS OR REGISTERED TRADEMARKS OF INTEL CORPORATION OR ITS SUBSIDIARIES IN THE UNITED STATES AND OTHER COUNTRIES.

MPEG IS AN INTERNATIONAL STANDARD FOR VIDEO COMPRESSION/DECOMPRESSION PROMOTED BY ISO. IMPLEMENTATIONS OF MPEG CODECS, OR MPEG ENABLED PLATFORMS MAY REQUIRE LICENSES FROM VARIOUS ENTITIES, INCLUDING INTEL CORPORATION.

*OTHER NAMES AND BRANDS MAY BE CLAIMED AS THE PROPERTY OF OTHERS.

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 optimization 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 product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

Table of Contents

Intel® Perceptual Computing SDK

The Intel® Perceptual Computing SDK is a library of pattern detection and recognition algorithm implementations exposed through standardized interfaces. The library aims at lowering barriers of using these algorithms and shifting the application developers' focus from coding the algorithm details to innovating on the usage of these algorithms for next generation human computer experience.

This document gives a general overview of the features.

Document Conventions

The SDK API uses the Verdana typeface for normal prose. With the exception of section headings and the table of contents, all code-related items appear in the `Courier New` typeface (`pxcStatus`). Hyperlinks appear in underlined boldface, such as **`pxcStatus`**.

Acronyms and Abbreviations

API	Application Programming Interface

Getting Started

Hardware and Software Requirements

The SDK requires the following hardware and software requirements:

- Hardware requirements:
  - IA-32 or 2nd Gen Intel® Core™ processor with Intel® 64 architecture
  - 500 MB free hard disk space
  - Creative* Interactive Gesture camera
- Software requirements:
  - Microsoft* Windows* 7 or Windows* 8 operating system with SP1 or later
  - Microsoft* Visual Studio* C++ 2008 with service pack 1 or newer

Camera Setup

It is recommended to install the camera on top of the computer or laptop lid, and plug the USB connection into one of the USB ports, as illustrated in Figure 1.

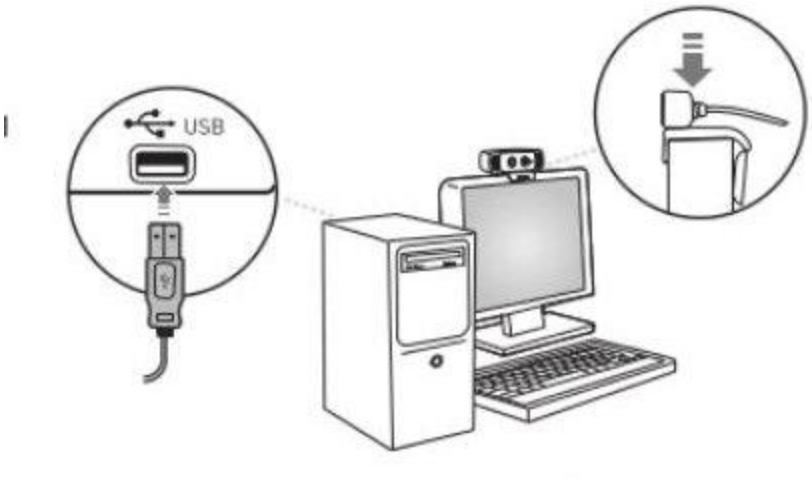


Figure 1: Camera Setup Position

When perfoming gestures, it is expected that the user leans back on the chair in a relaxed position. The user's hands moves around a virtual plane roughly 12 inches away from the camera. This virtual plane serves multiple purposes: (a) it activates hand tracking when the user's hand is within 12 inches from the camera; (b) the swipe gestures use the plane to distinguish between a left swipe and a right swipe.

It is also recommended that the user's head is always 8 inches away from the user's hands. The hand tracking software cannot reliably distinguish a hand from a head if they are too close to each other.

To avoid fertigue, it is critical that the user finds a relaxed position to satisfy the above two requirements.

Software Installation

Run the SDK installer **`intel_pc_sdk_ia32_5057.msi`** (32-bit), or **`intel_pc_sdk_intel64_5057.msi`** (64-bit). The last build number may vary depending on the releases.

You will see a welcome screen as illustructed in Figure 2. Follow the instructions to complete the installation process. By default, the SDK installs to the **`C:/Program Files/Intel/PCSDK`** directory. Note that if the SDK installer detects any existing SDK versions, the SDK installer will prompt you for an upgrade. It is however recommended to always do a clean uninstall and then install any new SDK versions.

Figure 2: Installer Welcome Screen

After installation, the SDK installer will prompt you for a system reboot. This step is critical to propogate all environmental variables.

Camera Testing

Run the **capture_viewer** application from the startup menu: **Startup→Intel® Perceptual Computing SDK 2013→Tools→Capture Viewer** as follows:

- Launch the **capture_viewer** application
- Select a color stream, a depth stream and an audio stream from the **DepthSense 325 Audio/Video Capture** module.
- Select **Control→Display** to render the streams

You should see three windows showing the color pictures, the depth map, and a visualization of audio channel(s) as illustrated in Figure 3.

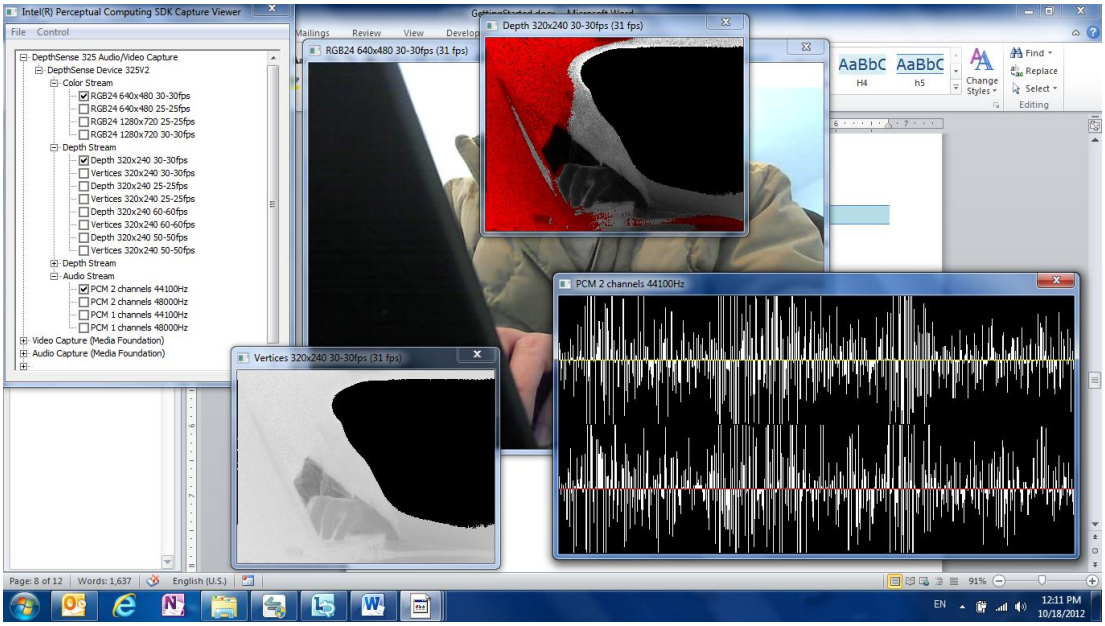


Figure 3: Camera Testing for Color, Depth and Audio

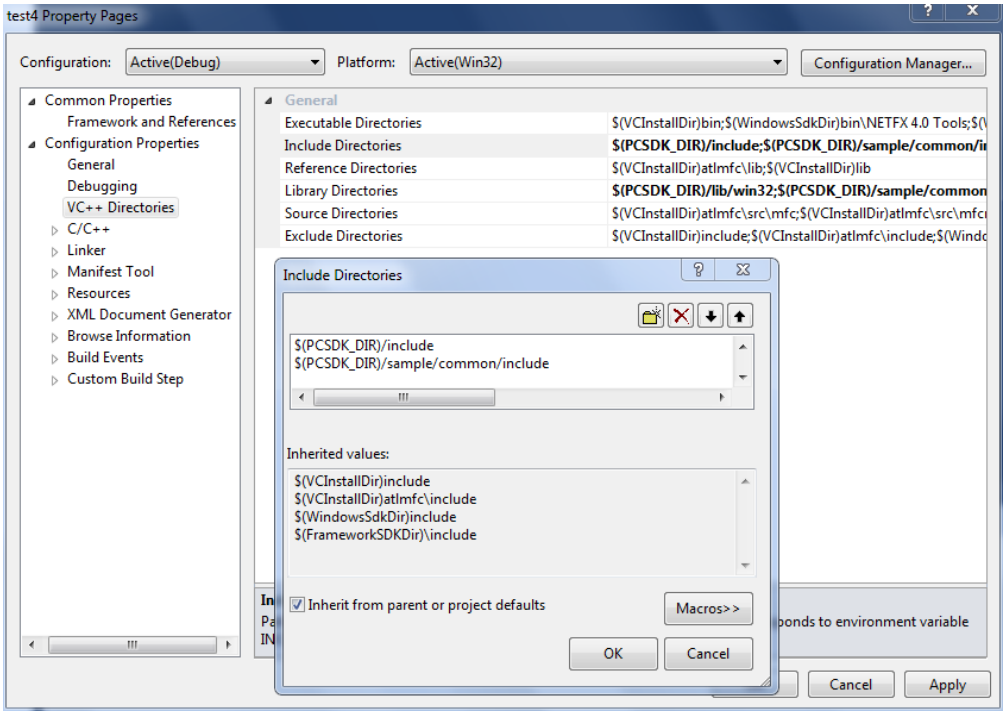
Microsoft* Visual Studio* 2008/2010 Setup

The SDK installer sets an environment variable **PCSDK_DIR** to point to the SDK installation directory. The developer can use this environment variable to set the include and library paths in the Microsoft Visual Studio environmental settings.

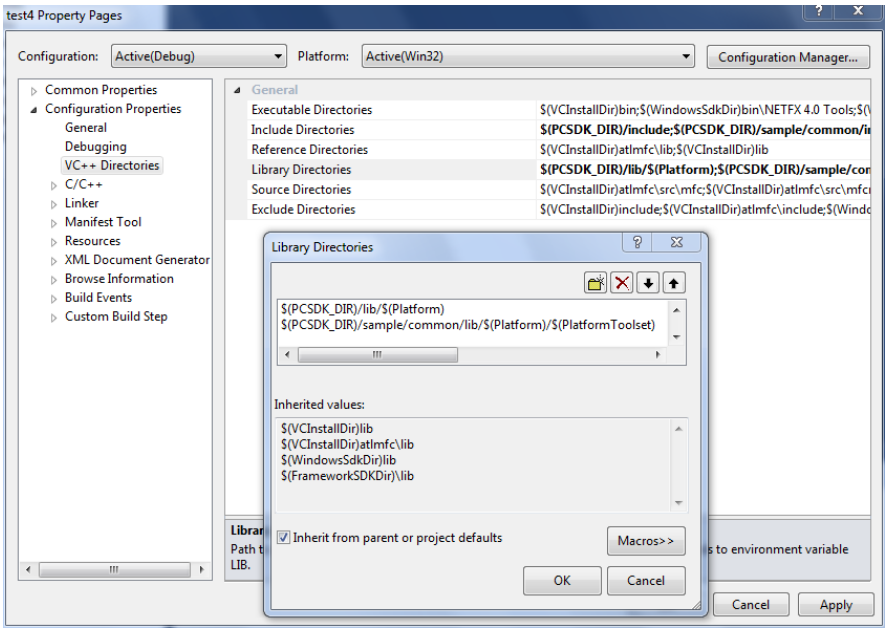
The steps for setting up Microsoft Visual Studio 2008/2010 are similar except that for Microsoft Visual Studio 2008, the settings are global in **Tools→Options→Project and Solutions →**

VC++ Directories, while in Microsoft Visual Studio 2010, the settings are part of an existing or newly created project properties:

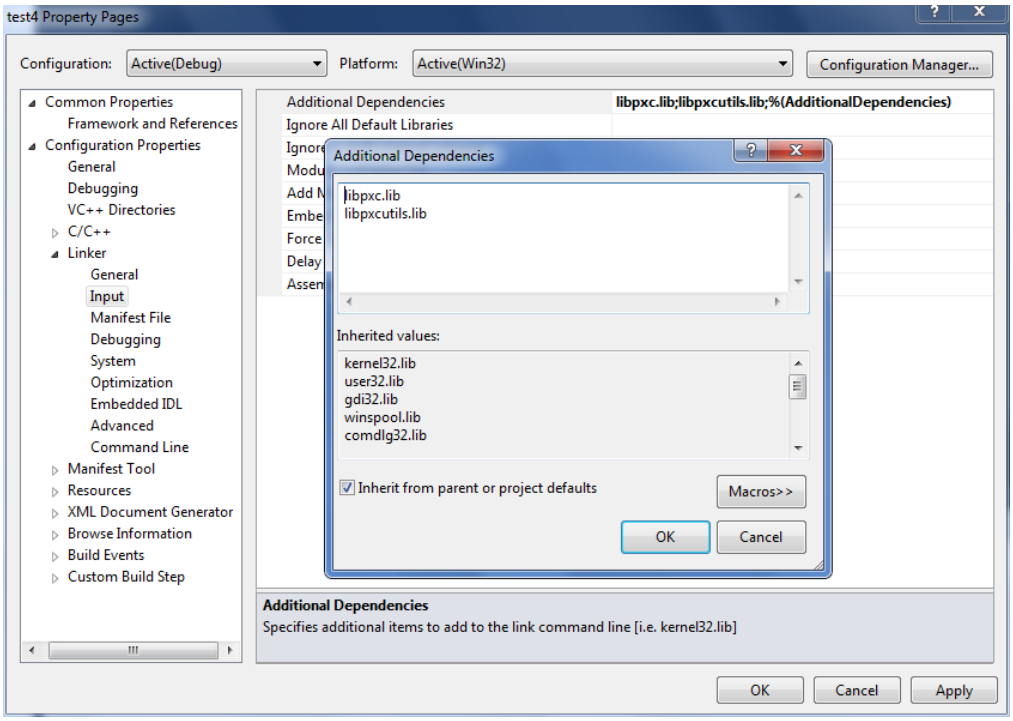
- For Microsoft Visual Studio 2008,
  - Open the project settings: **Tools→Options→Project and Solutions → VC++ Directories**.
- For Microsoft Visual Studio 2010,
  - Load an existing C++ application by using **File→Open→Project/Solution** or create a new C++ application using **File→New→Project**.
  - Open the project settings: **Project→Properties→VC++ Directories**.
- Added **$(PCSDK_DIR)/include** to the include path. Add **$(PCSDK_DIR)/sample/common/include** if you need to use the sample utility classes.



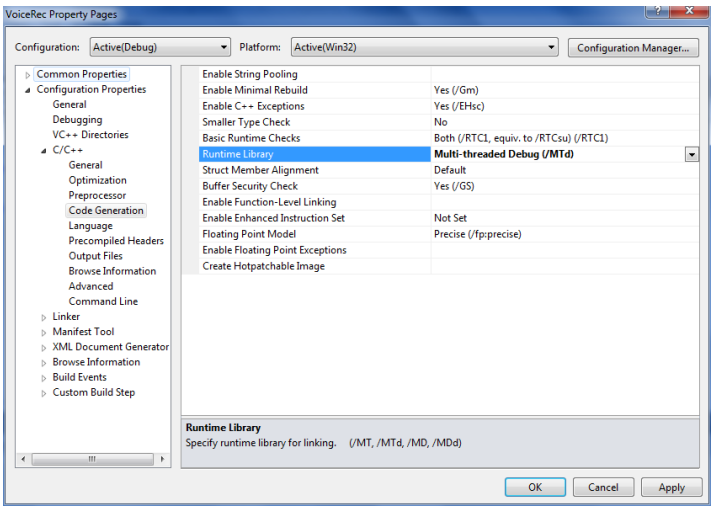
- Add **$(PCSDK_DIR)/lib/$(Platform)** the library path. Add **$(PCSDK_DIR)/sample/common/lib/$(PlatformName)/$(PlatformToolset)** if you need to use any sample utility classes.



- Add the SDK stub library **`libpxc.lib`** (for the **`Release`** build) or **`libpxc_d.lib`** (for the **`Debug`** build) to **`Additional Dependencies`** in **`Linker`**→**`Input`**. If you need to use any sample utility libraries, additionally add **`libpxcutils.lib`** (for the **`Release`** build) or **`libpxcutils_d.lib`** (for the **`Debug`** build).



- Go to the **C/C++→Code Optimization** tab and change the **Runtime Library** setting to be **Multi-Threaded Debug (/MTd)**, or **Multi-threaded (/MT)**, according to your project **Debug** or **Release** mode.



Note that if any of the compiler settings do not meet your application needs, you can recompile the **libpxc** or the **libpxcutils** libraries, whose build script is under **$(PCSDK_DIR)/src/libpxc** and **$(PCSDK_DIR)/sample/common** respectively.

A Simple SDK Application

Example 1 is a simple hello-world type of application that prints out all preinstalled SDK modules. The application will print out the following information (depending on what're preinstalled):

```
Module: DepthSense 325 Audio/Video Capture, iuid=0x32444b53
Module: Face Analysis (Intel), iuid=0x434c4946
Module: Hand/Finger Tracking & Gesture Recognition, iuid=0x49434b53
Module: Video Capture (Media Foundation), iuid=0x464d5f43
Module: Audio Capture (Media Foundation), iuid=0x41464d43
```

```
#include "stdafx.h"
#include "pxcsession.h"
#include "pxcsmartptr.h"
int _tmain(int argc, _TCHAR* argv[]) {
      PXCSmartPtr<PXCSession> session;
      PXCSession_Create(&session);
      for (int i=0;;i++) {
            PXCSession::ImplDesc desc;
            pxcStatus sts=session->QueryImpl(0,i,&desc);
            if (sts<PXC_STATUS_NO_ERROR) break;
            wprintf(L"Module: %s, iuid=0x%x\n",desc.friendlyName,desc.iuid);
      }
      return 0;
}
```

Example 1: The SDK Hello-World Application