Intel® INDE Visual Coding Framework Windows Samples Guide

Version Beta-1 2016

LEGAL DISCLAIMER

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.

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.

Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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 SSE3 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

Contents

1 Overview

The Intel® INDE Visual Coding Framework (VCF) Software Development Kit (SDK) is a software development library that enables programmatic access to the VCF cross-platform runtime.

This document provides an overview of the available samples for Windows* OS/platforms. Please refer to corresponding documents for sample guides for other OS/platforms.

For general information about VCF, please refer to the following web page: https://software.intel.com/en-us/visual-coding-framework

All VCF sample code is hosted on GitHub repository: https://github.com/INDExOS/visual-coding-framework

The VCF API currently supports language bindings for C and C++.

The sample code requires input such as media content and VCF GraphML input files. These files are located in the **"content"** folder.

2 Sample code Inventory

Sample Name	Description
simple_transcode	This sample showcases basic usage of the C++ VCF API by loading and running a video only transcoding graph (File->Decode->Encode->FileOut) The sample utilizes the following files from the content folder: - `tos1920x800_1000.mp4` - `vcf_transcode.graphml`
simple_transcode_events	This sample extends "simple_transcode" sample to showcase how to subscribe for VCF events. The sample utilizes the following files from the content folder: - `tos1920x800_1000.mp4` - `vcf_transcode.graphml`
simple_transcode - C	This sample is similar to the "simple_transcode" sample except that uses C language bindings. The sample utilizes the following files from the content folder: - `tos1920x800_1000.mp4` - `vcf_transcode.graphml`
simple_decode_render	This sample showcases VCF video and audio decode and rendering.

	Note that video rendering requires creation of a render window, with associated message loop. The sample utilizes the following files from the content folder: - `tos1920x800_1000.mp4` - `vcf_decode_render.graphml`
simple_camera_record_ and_render	This sample showcases VCF camera capture and encoding of video stream to file, including display of recorded stream. Note that video rendering requires creation of a render window, with associated message loop. Note that it is assumed that system has a YUY2 camera built-in our attached. The sample utilizes the following files from the content folder: - `vcf_camera_endode_render.graphml`

3 Requirements

Software

- Microsoft Visual Studio* 2013 or better
- Microsoft Windows* 7 or better
- Recent Intel Graphics driver. https://downloadcenter.intel.com

Hardware

- 3rd generation Intel® Core™ processor and Intel® micro architecture (code name Ivy Bridge) or better.
- Intel® Atom™ with Intel® Processor Graphics

Environment

- All samples depend on the "INTELVCFSDKROOT" environment variable to be set. This variable is set automatically when installing VCF. Users may also set this variable manually via Control Panel/System/Advanced System Settings/Environment Variables.
  The variable must point to the install location of the VCF SDK. E.g.
  `"C:\Intel\INDE\Visual_Coding_Framework_<version>\sdk\"`
- A reboot of the system is recommended in order for the Environment Variables to take effect.

4 Build Instructions

To build any of the samples projects follow the below steps:

1. Open the sample solution file (for instance: “simple_transcode_vs2013.sln”)
    a. Note that the VCF API library is built using Visual Studio 2013, so linking with older versions of Visual Studio will fail. If there is strong need for supporting older versions of Visual Studio please raise the issue on the VCF Support Forum.
    b. For Visual Studio 2015, users can safely upgrade from the 2013 project.
2. Select your preferred build configuration (E.g. x64/Debug)
3. Build the solution
4. The resulting application executable is created in the “`_build/<platform>/<config>/`” folder

Note that all sample projects invoke a post build step, which copies all the re-distributable VCF components (such as the core VCF runtime DLL) from the SDK location to the target build folder. The required input content for the sample is also copied to the build folder.

5 Execution Instructions

Since all the re-distributable VCF components are copied to the build folder during the build process there are no additional steps required to when executing the generated sample binary.

Below is an example of the result of executing the binary generated from the “simple_transcode_events” sample:

Command:

```
vcf_simple_transcode_events.exe
```

Result:

```
Status : Graph execution started...
[adts @ 00000046d0ce1520] Encoder did not produce proper pts,
making some up.
StatusProgress : 342
StatusProgress : 865
StatusProgress : 1371
StatusProgress : 1877
StatusProgress : 2368
```

```
Status : Graph execution completed successfully
BenchmarkData:status:0,execTimeS:5.93518,initTimeS:0.273315,numPa
ckets:2809,execFps:473.279,latencyAvgMs:1.9715,latencyMaxMs:10.51
99,latencyMinMs:0.0078004,
StatusProgress : 2809
VCF Graph execution completed (result code = 0)Note that this
workload generates an encoded file named “output.mp4”
```

Note: The “adts” warning, on the second line, can be safely ignored. Future revisions of VCF may be able to suppress these irrelevant warnings.