SDK Dispatcher ReadMe

Overview

This document describes the design of the SDK Dispatcher.

The dispatcher is a layer that lies between application and SDK implementations. Upon initialization, the dispatcher locates the appropriate platform-specific SDK implementation. If there is none, it will select the software SDK implementation. The dispatcher will redirect subsequent function calls to the same functions in the selected SDK implementation.

We recommend that applications always use the SDK functions through a dispatcher since platform-specific SDK implementations or software SDK implementations may have different names and reside in separate locations. The dispatching process represents a unique way to locate these SDK implementations.

To maintain forward and backward compatibility, we do not recommend modifying the dispatcher in any way, except in the following cases:

- Debugging in the case that the dispatcher fails to locate an SDK implementation
- Compiling the dispatcher library with a compiler that the SDK does not support
- Rebuilding the dispatcher static library to be a dynamically linked library (shared object) for modular design
- Tailoring redirected function entries to those applications actually used for a smaller footprint

SDK Implementation Registration

Both software and platform-specific SDK implementations installed on target system in the predefined folders.

Software SDK Implementations

Software SDK implementations are installed on the system in the:

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

Copyright © 2012-2018, Intel Corporation
Platform-Specific SDK Implementations

Platform-specific SDK implementations are installed on the system in the:

<table>
<thead>
<tr>
<th>Path</th>
<th>Platform type</th>
</tr>
</thead>
<tbody>
<tr>
<td>/opt/intel/mediasdk/lib64</td>
<td>for Intel® 64 architecture</td>
</tr>
</tbody>
</table>

The dispatcher locates SDK implementations by their reserved names:

<table>
<thead>
<tr>
<th>Library</th>
<th>Platform-Specific SDK Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>libmfxhw64-p.so.&lt;mj&gt;.&lt;mn&gt;</td>
<td>Reserved hardware library for Intel® 64 architecture</td>
</tr>
</tbody>
</table>

where p stands for Production build of the library, <mj> - major version of the library’s API, <mn> - minor version of the library’s API.

Example 1 illustrates an example of registered SDK platform-specific implementations in the file structure:

```
$ ls /opt/intel/mediasdk/lib64
libmfxhw64-p.so.1.27
```

Example 1: SDK Platform-Specific Implementations in the file structure

The Dispatching Procedure

The following describes the dispatching procedure:

1. Enumerate graphics adapters on the system and get their device and vendor IDs. This is being done by searching for the adapter devices in the /sys/bus/pci/devices folder.

   ```
   $ ls /sys/bus/pci/devices
   ```

2. Search for the SDK Platform-Specific or Software implementation in the folder /opt/intel/mediasdk/lib<arch> depending on MFXInit parameters as follows:

<table>
<thead>
<tr>
<th>MFXInit Initialization Parameter</th>
<th>Device &amp; Vendor Identifiers</th>
</tr>
</thead>
</table>

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

Copyright © 2012-2018, Intel Corporation
2. Searching for the SDK platform-specific implementation in the folder
   /opt/intel/mediasdk/lib<arch> defined on the previous step is done as follows:
   o Load SDK implementation with the requested major API version and maximum
     minor API version which is higher or equal to requested one
   3. Search the default SDK implementations in the default system paths
      or path addressed by LD_LIBRARY_PATH environment variable by first searching for platform-specific one
      (libmfxhw64.so) and then software (libmfxsw64.so).

Additional Recommendations

To ensure SDK API compatibility, unless applications explicitly specify a version of an SDK
implementation, we recommend that a dispatcher always match the SDK API version, during
session initialization (MFXInit), with the latest released SDK API version.

Build Instructions

To build dispatcher go to the opensource/mfx_dispatch directory and execute following command sequence:

```
$ mkdir build
$ cd build
$ cmake ../
$ make
```

Binaries will appear in the folder __lib; for example:

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

Copyright © 2012-2018, Intel Corporation
There is slight difference between using Dispatcher library from executable module or from shared object. To mitigate symbol conflict between itself and SDK shared object on Linux*, application should:

1. link against `libdispatch_shared.a` instead of `libmfx.a`
2. define `MFX_DISPATCHER_EXPOSED_PREFIX` before any SDK includes

```
$ ls -l $MFX_HOME/opensource/mfx_dispatch/build/__lib
libdispatch_trace.a
libdispatch_shared.a
libmfx.a
```
Legal Information

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or 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, the Intel logo, Intel Core, Intel 64, Intel HD Graphics, 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.

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

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

Copyright © 2012-2018, Intel Corporation